

# SPM12 Manual

---

The FIL Methods Group  
(and honorary members)

John Ashburner  
Gareth Barnes  
Chun-Chuan Chen  
Jean Daunizeau  
Guillaume Flandin  
Karl Friston  
Stefan Kiebel  
James Kilner  
Vladimir Litvak  
Rosalyn Moran  
Will Penny  
Adeel Razi  
Klaas Stephan  
Peter Zeidman  
  
Darren Gitelman  
Rik Henson  
Chloe Hutton  
Volkmar Glauche  
J r mie Mattout  
Christophe Phillips

Functional Imaging Laboratory  
Wellcome Trust Centre for Neuroimaging  
Institute of Neurology, UCL  
12 Queen Square, London WC1N 3BG, UK  
January 12, 2016

<http://www.fil.ion.ucl.ac.uk/spm/>



# Contents

<b>I</b>	<b>Temporal processing</b>	<b>17</b>
<b>1</b>	<b>Slice Timing</b>	<b>19</b>
1.1	Data . . . . .	20
1.1.1	Session . . . . .	20
1.2	Number of Slices . . . . .	20
1.3	TR . . . . .	20
1.4	TA . . . . .	20
1.5	Slice order . . . . .	20
1.6	Reference Slice . . . . .	21
1.7	Filename Prefix . . . . .	21
<b>II</b>	<b>Spatial processing</b>	<b>23</b>
<b>2</b>	<b>Realign</b>	<b>25</b>
2.1	Realign: Estimate . . . . .	25
2.1.1	Data . . . . .	25
2.1.2	Estimation Options . . . . .	26
2.2	Realign: Reslice . . . . .	27
2.2.1	Images . . . . .	27
2.2.2	Reslice Options . . . . .	27
2.3	Realign: Estimate & Reslice . . . . .	28
2.3.1	Data . . . . .	28
2.3.2	Estimation Options . . . . .	28
2.3.3	Reslice Options . . . . .	29
<b>3</b>	<b>Realign &amp; Unwarp</b>	<b>31</b>
3.1	Data . . . . .	34
3.1.1	Session . . . . .	34
3.2	Estimation Options . . . . .	34
3.2.1	Quality . . . . .	34
3.2.2	Separation . . . . .	34
3.2.3	Smoothing (FWHM) . . . . .	34
3.2.4	Num Passes . . . . .	34
3.2.5	Interpolation . . . . .	35
3.2.6	Wrapping . . . . .	35
3.2.7	Weighting . . . . .	35
3.3	Unwarp Estimation Options . . . . .	35
3.3.1	Basis Functions . . . . .	35
3.3.2	Regularisation . . . . .	35
3.3.3	Reg. Factor . . . . .	35
3.3.4	Jacobian deformations . . . . .	35
3.3.5	First-order effects . . . . .	36
3.3.6	Second-order effects . . . . .	36
3.3.7	Smoothing for unwarp (FWHM) . . . . .	36

3.3.8	Re-estimate movement params	36
3.3.9	Number of Iterations	36
3.3.10	Taylor expansion point	36
3.4	Unwarp Reslicing Options	36
3.4.1	Resliced images (unwarp)?	36
3.4.2	Interpolation	37
3.4.3	Wrapping	37
3.4.4	Masking	37
3.4.5	Filename Prefix	37
<b>4</b>	<b>Coregister</b>	<b>39</b>
4.1	Coregister: Estimate	39
4.1.1	Reference Image	39
4.1.2	Source Image	40
4.1.3	Other Images	40
4.1.4	Estimation Options	40
4.2	Coregister: Reslice	40
4.2.1	Image Defining Space	40
4.2.2	Images to Reslice	40
4.2.3	Reslice Options	40
4.3	Coregister: Estimate & Reslice	41
4.3.1	Reference Image	41
4.3.2	Source Image	41
4.3.3	Other Images	41
4.3.4	Estimation Options	41
4.3.5	Reslice Options	42
<b>5</b>	<b>Segment</b>	<b>43</b>
5.1	Data	44
5.1.1	Channel	44
5.2	Tissues	45
5.2.1	Tissue	45
5.3	Warping & MRF	46
5.3.1	MRF Parameter	46
5.3.2	Clean Up	46
5.3.3	Warping Regularisation	47
5.3.4	Affine Regularisation	47
5.3.5	Smoothness	47
5.3.6	Sampling distance	48
5.3.7	Deformation Fields	48
<b>6</b>	<b>Normalise</b>	<b>49</b>
6.1	Normalise: Estimate	49
6.1.1	Data	50
6.1.2	Estimation Options	50
6.2	Normalise: Write	51
6.2.1	Data	51
6.2.2	Writing Options	52
6.3	Normalise: Estimate & Write	52
6.3.1	Data	52
6.3.2	Estimation Options	53
6.3.3	Writing Options	54



<b>7</b>	<b>Smooth</b>	<b>57</b>
7.1	Images to Smooth	57
7.2	FWHM	57
7.3	Data Type	57
7.4	Implicit masking	57
7.5	Filename Prefix	57
<b>III</b>	<b>fMRI Statistics</b>	<b>59</b>
<b>8</b>	<b>fMRI model specification</b>	<b>61</b>
8.1	Timing parameters	61
8.1.1	Units for design	62
8.1.2	Interscan interval	62
8.1.3	Microtime resolution	62
8.1.4	Microtime onset	62
8.2	Data & Design	63
8.2.1	Subject/Session	64
8.3	Factorial design	65
8.3.1	Factor	66
8.4	Basis Functions	66
8.4.1	Canonical HRF	66
8.4.2	Other basis sets	66
8.5	Model Interactions (Volterra)	67
8.6	Directory	67
8.7	Global normalisation	67
8.8	Explicit mask	67
8.9	Serial correlations	68
8.10	Reviewing your design	68
<b>9</b>	<b>fMRI model estimation</b>	<b>71</b>
9.1	Select SPM.mat	71
9.2	Method	71
9.2.1	Classical	71
9.2.2	Bayesian 1st-level	72
9.2.3	Bayesian 2nd-level	75
9.3	Output files	76
9.3.1	Classical 1st-level	76
9.3.2	Bayesian 1st-level	76
9.4	Model comparison	76
<b>10</b>	<b>Factorial design specification</b>	<b>79</b>
10.1	Directory	80
10.2	Design	80
10.2.1	One-sample t-test	80
10.2.2	Two-sample t-test	80
10.2.3	Paired t-test	81
10.2.4	Multiple regression	81
10.2.5	One-way ANOVA	82
10.2.6	One-way ANOVA - within subject	83
10.2.7	Full factorial	84
10.2.8	Flexible factorial	86
10.3	Covariates	88
10.3.1	Covariate	88
10.4	Multiple covariates	88
10.4.1	Covariates	89
10.5	Masking	89

10.5.1	Threshold masking	89
10.5.2	Implicit Mask	90
10.5.3	Explicit Mask	90
10.6	Global calculation	90
10.6.1	Omit	90
10.6.2	User	90
10.6.3	Mean	91
10.7	Global normalisation	91
10.7.1	Overall grand mean scaling	91
10.7.2	Normalisation	91

## IV EEG/MEG 93

### 11 SPM for MEG/EEG overview 95

11.1	Welcome to SPM for M/EEG	95
11.2	Changes from SPM8 to SPM12	96

### 12 EEG/MEG preprocessing – Reference 97

12.1	Conversion of data	97
12.2	Converting arbitrary data	99
12.3	The M/EEG SPM format	99
12.4	Preparing the data after conversion and specifying batch inputs	100
12.4.1	Prepare (batch)	104
12.5	Integration of SPM and Fieldtrip	104
12.6	Loading data into workspace	104
12.7	The <code>meeg</code> object	104
12.7.1	Constructor <code>meeg</code>	105
12.7.2	Array-like interface	105
12.7.3	<code>display</code>	105
12.7.4	Number methods	105
12.7.5	Reading and manipulation of information	105
12.7.6	Reading of information	108
12.7.7	Manipulations of the data on disk	109
12.7.8	Struct-like interface	110
12.8	SPM functions	110
12.8.1	Epoching the data: <code>spm_eeg_epochs</code>	110
12.8.2	Filtering the data: <code>spm_eeg_filter</code>	111
12.8.3	Baseline correction: <code>spm_eeg_bc</code>	111
12.8.4	Artefact detection and rejection: <code>spm_eeg_artefact</code>	111
12.8.5	Downsampling: <code>spm_eeg_downsample</code>	112
12.8.6	Rereferencing: <code>spm_eeg_montage</code>	112
12.8.7	Grand mean: <code>spm_eeg_grandmean</code>	112
12.8.8	Merge: <code>spm_eeg_merge</code>	112
12.8.9	Multimodal fusion: <code>spm_eeg_fuse</code>	113
12.8.10	Cropping: <code>spm_eeg_crop</code>	113
12.8.11	Combine planar: <code>spm_eeg_combineplanar</code>	113
12.8.12	Data reduction: <code>spm_eeg_reduce</code>	113
12.8.13	Time-frequency decomposition: <code>spm_eeg_tf</code>	113
12.8.14	Rescaling and baseline correction of time-frequency: <code>spm_eeg_tf_rescale</code>	114
12.8.15	Averaging over time or frequency: <code>spm_eeg_avgtime</code> , <code>spm_eeg_avgfreq</code>	114
12.8.16	Averaging: <code>spm_eeg_average</code>	114
12.8.17	Contrast over epochs: <code>spm_eeg_contrast</code>	114
12.8.18	Copy: <code>spm_eeg_copy</code>	115
12.8.19	Remove bad trials: <code>spm_eeg_remove_bad_trials</code>	115
12.9	Displaying data with SPM M/EEG REVIEW	115
12.9.1	Data visualization	116

12.9.2 Source reconstructions visualization . . . . .	116
12.9.3 Script generation . . . . .	117
<b>13 Analysis in sensor space . . . . .</b>	<b>119</b>
13.0.4 Output . . . . .	119
13.0.5 Smoothing . . . . .	120
<b>14 3D source reconstruction: Imaging approach . . . . .</b>	<b>121</b>
14.1 Introduction . . . . .	121
14.2 Getting started . . . . .	122
14.3 Source space modeling . . . . .	122
14.4 Coregistration . . . . .	123
14.5 Forward computation ( <i>forward</i> ) . . . . .	124
14.6 Inverse reconstruction . . . . .	125
14.7 Summarizing the results of inverse reconstruction as an image . . . . .	126
14.8 Rendering interface . . . . .	127
14.9 Group inversion . . . . .	127
14.10Batching source reconstruction . . . . .	127
14.11Appendix: Data structure . . . . .	127
<b>15 Localization of Equivalent Current Dipoles . . . . .</b>	<b>129</b>
15.1 Introduction . . . . .	129
15.2 Procedure in SPM12 . . . . .	130
15.2.1 Head and forward model . . . . .	130
15.2.2 VB-ECD reconstruction . . . . .	130
15.2.3 Result display . . . . .	131
<b>16 Dynamic Causal Modelling for M/EEG . . . . .</b>	<b>133</b>
16.1 Introduction . . . . .	133
16.2 Overview . . . . .	134
16.3 Calling DCM for ERP/ERF . . . . .	134
16.4 load, save, select model type . . . . .	135
16.5 Data and design . . . . .	135
16.6 Electromagnetic model . . . . .	136
16.7 Neuronal model . . . . .	136
16.8 Estimation . . . . .	137
16.9 Results . . . . .	137
16.10Cross-spectral densities . . . . .	138
16.10.1 Model specification . . . . .	138
16.10.2The Lead-Field . . . . .	138
16.10.3Connections . . . . .	138
16.10.4Cross Spectral Densities . . . . .	138
16.10.5Output and Results . . . . .	139
16.11Induced responses . . . . .	139
16.11.1 Data . . . . .	139
16.11.2Electromagnetic model . . . . .	139
16.11.3Neuronal model . . . . .	139
16.11.4Wavelet transform . . . . .	139
16.11.5Results . . . . .	139
16.12Phase-coupled responses . . . . .	140
16.12.1 Data . . . . .	140
16.12.2Electromagnetic model . . . . .	140
16.12.3Neuronal model . . . . .	140
16.12.4Hilbert transform . . . . .	140
16.12.5Results . . . . .	140

<b>V Utilities</b>	<b>143</b>
<b>17 Display Image</b>	<b>145</b>
17.1 Image to Display . . . . .	146
<b>18 Check Registration</b>	<b>149</b>
18.1 Images to Display . . . . .	149
<b>19 Rendering</b>	<b>151</b>
19.1 Surface Extraction . . . . .	151
19.1.1 Input Images . . . . .	151
19.1.2 Surfaces . . . . .	151
19.2 Surface Rendering . . . . .	152
19.2.1 Objects . . . . .	152
19.2.2 Lights . . . . .	153
<b>20 Image Calculator</b>	<b>155</b>
20.1 Input Images . . . . .	155
20.2 Output Filename . . . . .	155
20.3 Output Directory . . . . .	155
20.4 Expression . . . . .	156
20.5 Additional Variables . . . . .	156
20.5.1 Variable . . . . .	156
20.6 Options . . . . .	156
20.6.1 Data Matrix . . . . .	156
20.6.2 Masking . . . . .	156
20.6.3 Interpolation . . . . .	156
20.6.4 Data Type . . . . .	157
<b>21 Import</b>	<b>159</b>
21.1 DICOM Import . . . . .	159
21.1.1 DICOM files . . . . .	159
21.1.2 Directory structure for converted files . . . . .	159
21.1.3 Output directory . . . . .	160
21.1.4 Protocol name filter . . . . .	160
21.1.5 Conversion options . . . . .	160
21.2 MINC Import . . . . .	160
21.2.1 MINC files . . . . .	160
21.2.2 Options . . . . .	160
21.3 ECAT Import . . . . .	161
21.3.1 ECAT files . . . . .	161
21.3.2 Options . . . . .	161
21.4 PAR/REC Import . . . . .	161
21.4.1 PAR files . . . . .	161
21.4.2 Options . . . . .	161
<b>22 De-face Images</b>	<b>163</b>
22.1 Images to de-face . . . . .	163
<b>23 Deformations</b>	<b>165</b>
23.1 Composition . . . . .	165
23.1.1 Dartel flow . . . . .	165
23.1.2 Deformation Field . . . . .	166
23.1.3 Identity (Reference Image) . . . . .	166
23.1.4 Identity (Bounding Box and Voxel Size) . . . . .	166
23.1.5 Imported _sn.mat . . . . .	167
23.1.6 Inverse . . . . .	167
23.2 Output . . . . .	167

23.2.1	Save Deformation . . . . .	167
23.2.2	Pullback . . . . .	168
23.2.3	Pushforward . . . . .	168
23.2.4	Surface . . . . .	170
23.2.5	Save Jacobian Determinants . . . . .	170
<b>VI Tools</b>		<b>171</b>
<b>24</b>	<b>FieldMap Toolbox</b>	<b>173</b>
24.1	Introduction . . . . .	173
24.2	Presubtracted Phase and Magnitude Data . . . . .	173
24.2.1	Data . . . . .	173
24.3	Real and Imaginary Data . . . . .	175
24.3.1	Data . . . . .	176
24.4	Phase and Magnitude Data . . . . .	176
24.4.1	Data . . . . .	176
24.5	Precalculated FieldMap (in Hz) . . . . .	176
24.5.1	Data . . . . .	176
24.6	Apply VDM . . . . .	177
24.6.1	Data . . . . .	177
24.6.2	Reslice Options . . . . .	177
24.7	Creating Field Maps Using the FieldMap GUI . . . . .	178
24.7.1	Create field map in Hz . . . . .	178
24.7.2	Create voxel displacement map (VDM) and unwarp EPI . . . . .	181
24.8	Using the FieldMap in Batch scripts . . . . .	182
24.9	Using the VDM file with Unwarp . . . . .	183
24.10	Appendices . . . . .	183
24.10.1	Processing Hz field maps . . . . .	183
24.10.2	Converting Hz field map to VDM . . . . .	184
24.10.3	Matching field map data to EPI data . . . . .	184
<b>25</b>	<b>Dartel Tools</b>	<b>185</b>
25.1	Initial Import . . . . .	186
25.1.1	Parameter Files . . . . .	186
25.1.2	Output Directory . . . . .	187
25.1.3	Bounding box . . . . .	187
25.1.4	Voxel size . . . . .	187
25.1.5	Image option . . . . .	187
25.1.6	Grey Matter . . . . .	187
25.1.7	White Matter . . . . .	187
25.1.8	CSF . . . . .	187
25.2	Run Dartel (create Templates) . . . . .	187
25.2.1	Images . . . . .	187
25.2.2	Settings . . . . .	187
25.3	Run Dartel (existing Templates) . . . . .	189
25.3.1	Images . . . . .	189
25.3.2	Settings . . . . .	189
25.4	Normalise to MNI Space . . . . .	190
25.4.1	Dartel Template . . . . .	190
25.4.2	Select according to . . . . .	190
25.4.3	Voxel sizes . . . . .	191
25.4.4	Bounding box . . . . .	191
25.4.5	Preserve . . . . .	191
25.4.6	Gaussian FWHM . . . . .	191
25.5	Create Warped . . . . .	191
25.5.1	Flow fields . . . . .	192

25.5.2	Images	192
25.5.3	Modulation	192
25.5.4	Time Steps	192
25.5.5	Interpolation	192
25.6	Jacobian determinants	192
25.6.1	Flow fields	192
25.6.2	Time Steps	192
25.7	Create Inverse Warped	193
25.7.1	Flow fields	193
25.7.2	Images	193
25.7.3	Time Steps	193
25.7.4	Interpolation	193
25.8	Population to ICBM Registration	193
25.8.1	Dartel Template	193
25.9	Kernel Utilities	193
25.9.1	Kernel from Images	194
25.9.2	Kernel from Flows	194
<b>26</b>	<b>Shoot Tools</b>	<b>195</b>
26.1	Run Shooting (create Templates)	195
26.1.1	Images	196
26.2	Run Shoot (existing Templates)	196
26.2.1	Images	196
26.2.2	Templates	196
26.3	Kernel Utilities	196
26.3.1	Kernel from velocities	196
26.3.2	Generate Scalar Momenta	197
26.3.3	Kernel from Images	197
<b>27</b>	<b>Longitudinal Registration</b>	<b>199</b>
27.1	Pairwise Longitudinal Registration	199
27.1.1	Time 1 Volumes	200
27.1.2	Time 2 Volumes	200
27.1.3	Time Difference	200
27.1.4	Noise Estimate	200
27.1.5	Warping Regularisation	200
27.1.6	Bias Regularisation	200
27.1.7	Save Mid-point average	201
27.1.8	Save Jacobian Rate	201
27.1.9	Save Divergence Rate	201
27.1.10	Deformation Fields	201
27.2	Serial Longitudinal Registration	201
27.2.1	Volumes	201
27.2.2	Times	202
27.2.3	Noise Estimate	202
27.2.4	Warping Regularisation	202
27.2.5	Bias Regularisation	202
27.2.6	Save Mid-point average	203
27.2.7	Save Jacobians	203
27.2.8	Save Divergence	203
27.2.9	Deformation Fields	203

<b>28 Old Normalise</b>	<b>205</b>
28.1 Old Normalise: Estimate	206
28.1.1 Data	206
28.1.2 Estimation Options	206
28.2 Old Normalise: Write	207
28.2.1 Data	207
28.2.2 Writing Options	208
28.3 Old Normalise: Estimate & Write	208
28.3.1 Data	208
28.3.2 Estimation Options	209
28.3.3 Writing Options	210
<b>29 Old Segment</b>	<b>211</b>
29.1 Data	212
29.2 Output Files	212
29.2.1 Grey Matter	215
29.2.2 White Matter	215
29.2.3 Cerebro-Spinal Fluid	215
29.2.4 Bias Corrected	215
29.2.5 Clean up any partitions	215
29.3 Custom	215
29.3.1 Tissue probability maps	215
29.3.2 Gaussians per class	216
29.3.3 Affine Regularisation	216
29.3.4 Warping Regularisation	216
29.3.5 Warp Frequency Cutoff	217
29.3.6 Bias regularisation	217
29.3.7 Bias FWHM	217
29.3.8 Sampling distance	217
29.3.9 Masking image	217
<b>VII Data sets and examples</b>	<b>219</b>
<b>30 Auditory fMRI data</b>	<b>221</b>
30.1 Preamble (dummy scans)	221
30.2 Spatial pre-processing	223
30.2.1 Realignment	223
30.2.2 Coregistration	223
30.2.3 Segmentation	223
30.2.4 Normalise	227
30.2.5 Smoothing	227
30.3 Model specification, review and estimation	231
30.3.1 Estimate	231
30.4 Inference	231
30.4.1 Contrast manager	235
30.4.2 Masking	235
30.4.3 Thresholds	235
30.4.4 Files	236
30.4.5 Maximum Intensity Projections	236
30.4.6 Design matrix	238
30.4.7 Statistical tables	238
30.4.8 Plotting responses at a voxel	239
30.4.9 Overlays	240

<b>31 Face fMRI data</b>	<b>243</b>
31.1 Spatial pre-processing	243
31.1.1 Display	243
31.1.2 Realignment	243
31.1.3 Slice timing correction	248
31.1.4 Coregistration	248
31.1.5 Segmentation	248
31.1.6 Normalise	251
31.1.7 Smoothing	251
31.2 Modelling categorical responses	253
31.2.1 Estimate	255
31.2.2 Inference for categorical design	255
31.2.3 Statistical tables	255
31.2.4 F-contrasts	257
31.2.5 F-contrasts for testing effects of movement	261
31.3 Modelling parametric responses	261
31.3.1 Estimate	263
31.3.2 Plotting parametric responses	263
31.4 Bayesian analysis	266
31.4.1 Specification	266
31.4.2 Estimation	267
31.4.3 Inference	268
<b>32 Face group fMRI data</b>	<b>271</b>
32.1 Introduction	271
32.2 Data	271
32.3 Canonical HRF	272
32.4 Informed basis set	274
32.4.1 Nonsphericity	276
32.4.2 Informed Results	276
32.4.3 T- and F-contrasts	280
32.5 FIR basis set	282
32.5.1 Nonsphericity again	283
32.5.2 FIR Results	286
<b>33 Mixed Effects Analysis</b>	<b>291</b>
33.1 Introduction	291
<b>34 Verbal Fluency PET data</b>	<b>293</b>
34.1 Introduction	293
34.2 Single subject	293
34.3 Multiple subjects	294
34.3.1 Subject and Condition design	296
34.3.2 Subject and Time design	296
34.3.3 Subject by Condition design	298
34.3.4 Contrast manager	300
34.3.5 Masking and thresholds	302
34.3.6 MIPs and results tables	303
34.3.7 Small volume correction	305
34.3.8 Extracting data from regions	305
34.3.9 Inclusive Masking	307
34.3.10 Conjunctions	307



<b>35 Dynamic Causal Modeling for fMRI</b>	<b>311</b>
35.1 Theoretical background	311
35.2 Bayesian model selection	314
35.3 Practical example	315
35.3.1 Defining the GLM	316
35.3.2 Extracting time series	317
35.3.3 Specifying and estimating the DCM	318
35.3.4 Comparing models	321
<b>36 Psychophysiological Interactions (PPI)</b>	<b>325</b>
36.1 Theoretical background	325
36.2 Psycho-Physiologic Interaction Analysis: Summary of Steps	327
36.3 Practical example	327
36.3.1 GLM analysis - Design setup and estimation	328
36.3.2 GLM analysis - Results	332
36.4 GLM analysis - Extracting VOIs	334
36.5 PPI analysis - Create PPI variable	334
36.5.1 PPI GLM analysis - Design setup and estimation	336
36.5.2 PPI analysis - Results	337
36.5.3 PPI analysis - Plotting	338
<b>37 Bayesian Model Inference</b>	<b>341</b>
37.1 Background	341
37.2 Data	341
37.3 Analysis	342
37.3.1 Single Family	342
37.3.2 Bayesian Model Averaging	345
37.3.3 Family level inference	345
37.3.4 Summary Statistics and Group Analyses	345
37.4 BMS.mat file	345
37.4.1 Model level results	349
37.4.2 Family level results	349
37.4.3 Bayesian model averaging (BMA)	349
37.5 model_space.mat file	350
<b>38 Dynamic Causal Modelling for resting state fMRI</b>	<b>351</b>
38.1 Theoretical background	351
38.2 Practical example	353
38.2.1 Defining the GLM	353
38.2.2 Extracting time series	355
38.2.3 Specifying and estimating the DCM	355
<b>39 MEG source localisation</b>	<b>361</b>
39.1 Overview	361
39.2 Simulation	361
39.3 Imaging solutions for evoked or induced responses	363
39.3.1 IID (minimum norm)	363
39.3.2 Smooth priors (COH)	364
39.3.3 The Multiple sparse priors algorithm	364
39.3.4 Making summary images	366
39.3.5 Other MSP options	366
39.4 Dipole fitting to the average	367
39.4.1 Load/preview the data	367
39.4.2 Inversion	368

<b>40 EEG Mismatch negativity data</b>	<b>371</b>
40.1 Preprocessing	371
40.1.1 Simple conversion and reviewing	371
40.1.2 Preparing batch inputs	372
40.1.3 Preprocessing step by step	373
40.1.4 Automatisation of preprocessing	376
40.2 Sensor space analysis	378
40.2.1 Batching statistics	379
40.3 Source reconstruction	379
40.3.1 Mesh	379
40.3.2 Coregister	381
40.3.3 Forward model	381
40.3.4 Invert	381
40.3.5 Batching source reconstruction	382
40.4 Dynamic Causal Modeling	386
<b>41 Advanced topics in M/EEG artefact removal</b>	<b>391</b>
41.1 Artefact marking	391
41.2 Reviewing marked artefacts	392
41.3 Trial rejection based on marked artefacts	392
41.4 Explicit artefact exclusion in robust averaging	393
41.5 Topography-based artefact correction	393
41.6 Fieldtrip visual artefact rejection	398
<b>42 Multimodal, Multisubject data fusion</b>	<b>401</b>
42.1 Overview	401
42.2 Getting Started	402
42.3 Preprocessing M/EEG data	402
42.3.1 Convert (and epoch)	402
42.3.2 Prepare	404
42.3.3 Downsample	404
42.3.4 Baseline Correction	405
42.3.5 Deleting intermediate steps (optional)	405
42.3.6 Merging (concatenating runs)	406
42.3.7 Prepare (a montage for re-referencing the EEG)	406
42.3.8 Montage	407
42.4 Evoked analysis	407
42.4.1 Crop	407
42.4.2 Artefact detection	408
42.4.3 Combine Planar Gradiometers	408
42.4.4 Trial averaging	408
42.4.5 Contrasting conditions	408
42.4.6 Time-Sensor images	410
42.5 Scalp-Time Statistics across trials within one subject	413
42.5.1 Model Specification	413
42.5.2 Model Estimation	413
42.5.3 Setting up contrasts	413
42.6 Time-Frequency Analysis (Evoked and Induced power)	414
42.6.1 Wavelet estimation	416
42.6.2 Crop	416
42.6.3 Average	417
42.6.4 Baseline rescaling	417
42.6.5 Contrasting conditions	417
42.6.6 Creating 2D time-frequency images	417
42.6.7 Model Specification, Estimation and Contrasts	420
42.7 fMRI Preprocessing and Statistics	422
42.7.1 Realignment of EPI (fMRI) data	422

42.7.2	Normalisation/Segmentation of T1 images . . . . .	422
42.7.3	Coregistration of mean EPI (fMRI) to T1 (sMRI) . . . . .	422
42.7.4	Application of Normalisation parameters to EPI data . . . . .	422
42.7.5	Smoothing . . . . .	423
42.7.6	Creating a 1st-level (fMRI) GLM . . . . .	423
42.7.7	Model Estimation . . . . .	423
42.7.8	Setting up contrasts . . . . .	423
42.7.9	Group Statistics on fMRI data . . . . .	425
42.8	Source Reconstruction . . . . .	425
42.8.1	Create Head Model . . . . .	427
42.8.2	Model Inversion . . . . .	428
42.8.3	Time-frequency contrasts . . . . .	428
42.8.4	Group Statistics on Source Reconstructions . . . . .	431
42.9	Group Source Reconstruction . . . . .	433
42.9.1	Group Statistics on Source Reconstructions . . . . .	433
42.10	Group MEEG Source Reconstruction with fMRI priors . . . . .	433
42.10.1	Group Statistics on Source Reconstructions . . . . .	435
42.11	References . . . . .	435
42.12	Acknowledgements . . . . .	437
<b>43</b>	<b>DCM for Induced Responses</b>	<b>439</b>
43.1	Data . . . . .	439
43.2	Getting Started . . . . .	439
43.3	Setting up DCM . . . . .	439
43.3.1	load, save, select model type . . . . .	440
43.3.2	Data and design . . . . .	440
43.3.3	Electromagnetic model . . . . .	442
43.4	Neuronal model . . . . .	444
43.5	Estimation . . . . .	445
43.6	Results . . . . .	446
43.6.1	Frequency modes . . . . .	446
43.6.2	Time modes . . . . .	446
43.6.3	Time-Frequency . . . . .	446
43.6.4	Coupling (A-Hz) . . . . .	446
43.6.5	Coupling (B-Hz) . . . . .	446
43.6.6	Coupling (A-modes) . . . . .	446
43.6.7	Coupling (B-Hz) . . . . .	446
43.6.8	Input (C-Hz) . . . . .	447
43.6.9	Input (u-ms) . . . . .	447
43.6.10	Dipoles . . . . .	447
43.6.11	Save as img . . . . .	447
43.7	Model comparison . . . . .	447
<b>44</b>	<b>DCM for Phase Coupling</b>	<b>449</b>
44.1	Data . . . . .	449
44.2	Getting Started . . . . .	449
44.3	Data and design . . . . .	449
44.4	Electromagnetic model . . . . .	450
44.5	Neuronal model . . . . .	450
44.6	Results . . . . .	451
44.7	Extensions . . . . .	451

<b>45 DCM for Cross Spectral Densities: Anaesthesia Depth in Rodent Data</b>	<b>455</b>
45.1 Overview	455
45.2 Main Results	456
45.3 Using the Graphical User Interface to Obtain those Results	456
45.3.1 The data	456
45.3.2 Dynamic Causal Modelling of Cross Spectral Densities	456
45.3.3 Comparing models using Bayesian Model Selection	460
<b>46 Using Dartel</b>	<b>463</b>
46.1 Using Dartel for VBM	463
46.1.1 Using Spatial→Segment	463
46.1.2 Using Dartel Tools→Run Dartel (create Template)	464
46.1.3 Using Dartel Tools→Normalise to MNI Space	464
46.2 Spatially normalising functional data to MNI space	466
46.2.1 An alternative approach for using Dartel to spatially normalise to MNI Space	469
46.3 Warping Images to Existing Templates	471
46.4 Warping one individual to match another	471
<b>VIII Batch Interface</b>	<b>475</b>
<b>47 Batch tutorial</b>	<b>477</b>
47.1 Single subject	477
47.1.1 Study specific input data	478
47.1.2 Necessary processing steps	478
47.1.3 Add modules to the batch	478
47.1.4 Configure subject-independent data	478
47.1.5 Data flow	480
47.1.6 Entering subject-specific data	483
47.2 Advanced features	483
47.2.1 Multiple sessions	483
47.2.2 Processing multiple subjects in GUI	485
47.2.3 Command line interface	485
47.2.4 Modifying a saved job	487
<b>48 Developer's guide</b>	<b>489</b>
48.1 SPM and Matlabbatch code organisation	489
48.1.1 Code organisation	489
48.1.2 Interfaces between SPM and Matlabbatch	489
48.2 Configuration Code Details	489
48.2.1 Virtual Outputs	490
48.2.2 SPM Startup	490
48.2.3 Defaults Settings	490
48.3 Utilities	491
48.3.1 Batch Utilities	491
48.3.2 MATLAB Code Generation	491
48.3.3 Configuration Management	491
<b>IX Bibliography</b>	<b>493</b>

## Part I

# Temporal processing



# Chapter 1

## Slice Timing

### Contents

---

<b>1.1 Data</b> .....	<b>20</b>
1.1.1 Session .....	20
<b>1.2 Number of Slices</b> .....	<b>20</b>
<b>1.3 TR</b> .....	<b>20</b>
<b>1.4 TA</b> .....	<b>20</b>
<b>1.5 Slice order</b> .....	<b>20</b>
<b>1.6 Reference Slice</b> .....	<b>21</b>
<b>1.7 Filename Prefix</b> .....	<b>21</b>

---

Correct differences in image acquisition time between slices. Slice-time corrected files are prepended with an 'a'.

Note: The sliceorder arg that specifies slice acquisition order is a vector of N numbers, where N is the number of slices per volume. Each number refers to the position of a slice within the image file. The order of numbers within the vector is the temporal order in which those slices were acquired. To check the order of slices within an image file, use the SPM Display option and move the cross-hairs to a voxel co-ordinate of z=1. This corresponds to a point in the first slice of the volume.

The function corrects differences in slice acquisition times. This routine is intended to correct for the staggered order of slice acquisition that is used during echo-planar scanning. The correction is necessary to make the data on each slice correspond to the same point in time. Without correction, the data on one slice will represent a point in time as far removed as 1/2 the TR from an adjacent slice (in the case of an interleaved sequence).

This routine "shifts" a signal in time to provide an output vector that represents the same (continuous) signal sampled starting either later or earlier. This is accomplished by a simple shift of the phase of the sines that make up the signal. Recall that a Fourier transform allows for a representation of any signal as the linear combination of sinusoids of different frequencies and phases. Effectively, we will add a constant to the phase of every frequency, shifting the data in time.

Shifter - This is the filter by which the signal will be convolved to introduce the phase shift. It is constructed explicitly in the Fourier domain. In the time domain, it may be described as an impulse (delta function) that has been shifted in time the amount described by TimeShift. The correction works by lagging (shifting forward) the time-series data on each slice using sinc-interpolation. This results in each time series having the values that would have been obtained had the slice been acquired at the same time as the reference slice. To make this clear, consider a neural event (and ensuing hemodynamic response) that occurs simultaneously on two adjacent slices. Values from slice "A" are acquired starting at time zero, simultaneous to the neural event,

while values from slice "B" are acquired one second later. Without correction, the "B" values will describe a hemodynamic response that will appear to have began one second EARLIER on the "B" slice than on slice "A". To correct for this, the "B" values need to be shifted towards the Right, i.e., towards the last value.

This correction assumes that the data are band-limited (i.e. there is no meaningful information present in the data at a frequency higher than that of the Nyquist). This assumption is supported by the study of Josephs et al (1997, NeuroImage) that obtained event-related data at an effective TR of 166 msec. No physiological signal change was present at frequencies higher than our typical Nyquist (0.25 Hz).

When using the slice timing correction it is very important that you input the correct slice order, and if there is any uncertainty then users are encouraged to work with their physicist to determine the actual slice acquisition order.

One can also consider augmenting the model by including the temporal derivative in the informed basis set instead of slice timing, which can account for +/- 1 second of changes in timing.

Written by Darren Gitelman at Northwestern U., 1998. Based (in large part) on ACQCORRECT.PRO from Geoff Aguirre and Eric Zarahn at U. Penn.

## 1.1 Data

Subjects or sessions. The same parameters specified below will be applied to all sessions.

### 1.1.1 Session

Select images to slice-time correct.

## 1.2 Number of Slices

Enter the number of slices.

## 1.3 TR

Enter the TR (in seconds).

## 1.4 TA

Enter the TA (in seconds). It is usually calculated as  $TR - (TR/nslices)$ . You can simply enter this equation with the variables replaced by appropriate numbers.

If the next two items are entered in milliseconds, this entry will not be used and can be set to 0.

## 1.5 Slice order

Enter the slice order. Bottom slice = 1. Sequence types and examples of code to enter are given below:

```
ascending (first slice=bottom): [1:1:nslices]
descending (first slice=top): [nslices:-1:1]
interleaved (middle-top):
for k = 1:nslices
round((nslices-k)/2 + (rem((nslices-k),2) * (nslices - 1)/2)) + 1,
end
interleaved (bottom -> up): [1:2:nslices 2:2:nslices]
interleaved (top -> down): [nslices:-2:1, nslices-1:-2:1]
```

Alternatively you can enter the slice timing in ms for each slice individually.



If doing so, the next item (Reference Slice) will contain a reference time (in ms) instead of the slice index of the reference slice.

For Siemens scanners, this can be acquired in MATLAB from the dicom header as follows (use any volume after the first one):

```
hdr = spm_dicom_headers('dicom.ima');  
slice_times = hdr1.Private_0019_1029
```

Note that slice ordering is assumed to be from foot to head. If it is not, enter instead: TR - INTRASCAN\_TIME - SLICE\_TIMING\_VECTOR

## 1.6 Reference Slice

Enter the reference slice.

If slice times are provided instead of slice indices in the previous item, this value should represent a reference time (in ms) instead of the slice index of the reference slice.

## 1.7 Filename Prefix

Specify the string to be prepended to the filenames of the slice-time corrected image file(s). Default prefix is 'a'.



## Part II

# Spatial processing



# Chapter 2

## Realign

### Contents

---

<b>2.1 Realign: Estimate</b> . . . . .	<b>25</b>
2.1.1 Data . . . . .	25
2.1.2 Estimation Options . . . . .	26
<b>2.2 Realign: Reslice</b> . . . . .	<b>27</b>
2.2.1 Images . . . . .	27
2.2.2 Reslice Options . . . . .	27
<b>2.3 Realign: Estimate &amp; Reslice</b> . . . . .	<b>28</b>
2.3.1 Data . . . . .	28
2.3.2 Estimation Options . . . . .	28
2.3.3 Reslice Options . . . . .	29

---

Within-subject registration of image time series.

### 2.1 Realign: Estimate

This routine realigns a time-series of images acquired from the same subject using a least squares approach and a 6 parameter (rigid body) spatial transformation [31]. The first image in the list specified by the user is used as a reference to which all subsequent scans are realigned. The reference scan does not have to be the first chronologically and it may be wise to choose a "representative scan" in this role.

The aim is primarily to remove movement artefact in fMRI and PET time-series (or more generally longitudinal studies). The headers are modified for each of the input images, such that they reflect the relative orientations of the data. The details of the transformation are displayed in the results window as plots of translation and rotation. A set of realignment parameters are saved for each session, named `rp_*.txt`. These can be modelled as confounds within the general linear model [31].

#### 2.1.1 Data

Add new sessions for this subject. In the coregistration step, the sessions are first realigned to each other, by aligning the first scan from each session to the first scan of the first session. Then the images within each session are aligned to the first image of the session. The parameter estimation is performed this way because it is assumed (rightly or not) that there may be systematic differences in the images between sessions.

## Session

Select scans for this session. In the coregistration step, the sessions are first realigned to each other, by aligning the first scan from each session to the first scan of the first session. Then the images within each session are aligned to the first image of the session. The parameter estimation is performed this way because it is assumed (rightly or not) that there may be systematic differences in the images between sessions.

### 2.1.2 Estimation Options

Various registration options. If in doubt, simply keep the default values.

#### Quality

Quality versus speed trade-off. Highest quality (1) gives most precise results, whereas lower qualities gives faster realignment. The idea is that some voxels contribute little to the estimation of the realignment parameters. This parameter is involved in selecting the number of voxels that are used.

#### Separation

The separation (in mm) between the points sampled in the reference image. Smaller sampling distances gives more accurate results, but will be slower.

#### Smoothing (FWHM)

The FWHM of the Gaussian smoothing kernel (mm) applied to the images before estimating the realignment parameters.

- \* PET images typically use a 7 mm kernel.

- \* MRI images typically use a 5 mm kernel.

#### Num Passes

Register to first: Images are registered to the first image in the series. Register to mean: A two pass procedure is used in order to register the images to the mean of the images after the first realignment.

PET images are typically registered to the mean. This is because PET data are more noisy than fMRI and there are fewer of them, so time is less of an issue.

MRI images are typically registered to the first image. The more accurate way would be to use a two pass procedure, but this probably wouldn't improve the results so much and would take twice as long to run.

#### Interpolation

The method by which the images are sampled when estimating the optimum transformation. Higher degree interpolation methods provide the better interpolation, but they are slower because they use more neighbouring voxels [93, 94, 95].

#### Wrapping

This indicates which directions in the volumes the values should wrap around in. For example, in MRI scans, the images wrap around in the phase encode direction, so (e.g.) the subject's nose may poke into the back of the subject's head. These are typically:

No wrapping - for PET or images that have already been spatially transformed. Also the recommended option if you are not really sure.

- Wrap in Y - for (un-resliced) MRI where phase encoding is in the Y direction (voxel space).

### Weighting

The option of providing a weighting image to weight each voxel of the reference image differently when estimating the realignment parameters. The weights are proportional to the inverses of the standard deviations. This would be used, for example, when there is a lot of extra-brain motion - e.g., during speech, or when there are serious artifacts in a particular region of the images.

## 2.2 Realign: Reslice

This function reslices a series of registered images such that they match the first image selected voxel-for-voxel. The resliced images are named the same as the originals, except that they are prefixed by 'r'.

### 2.2.1 Images

Select scans to reslice to match the first.

### 2.2.2 Reslice Options

Various reslicing options. If in doubt, simply keep the default values.

#### Resliced images

All Images (1..n) : This reslices all the images - including the first image selected - which will remain in its original position.

Images 2..n : Reslices images 2..n only. Useful for if you wish to reslice (for example) a PET image to fit a structural MRI, without creating a second identical MRI volume.

All Images + Mean Image : In addition to reslicing the images, it also creates a mean of the resliced image.

Mean Image Only : Creates the mean resliced image only.

#### Interpolation

The method by which the images are sampled when being written in a different space. Nearest Neighbour is fastest, but not recommended for image realignment. Trilinear Interpolation is probably OK for PET, but not so suitable for fMRI because higher degree interpolation generally gives better results [93, 94, 95]. Although higher degree methods provide better interpolation, but they are slower because they use more neighbouring voxels. Fourier Interpolation [24, 20] is another option, but note that it is only implemented for purely rigid body transformations. Voxel sizes must all be identical and isotropic.

#### Wrapping

This indicates which directions in the volumes the values should wrap around in. For example, in MRI scans, the images wrap around in the phase encode direction, so (e.g.) the subject's nose may poke into the back of the subject's head. These are typically:

No wrapping - for PET or images that have already been spatially transformed.

Wrap in Y - for (un-resliced) MRI where phase encoding is in the Y direction (voxel space).

#### Masking

Because of subject motion, different images are likely to have different patterns of zeros from where it was not possible to sample data. With masking enabled, the program searches through the whole time series looking for voxels which need to be sampled from outside the original images. Where this occurs, that voxel is set to zero for the whole set of images (unless the image format can represent NaN, in which case NaNs are used where possible).

**Filename Prefix**

Specify the string to be prepended to the filenames of the resliced image file(s). Default prefix is 'r'.

**2.3 Realign: Estimate & Reslice**

This routine realigns a time-series of images acquired from the same subject using a least squares approach and a 6 parameter (rigid body) spatial transformation [31]. The first image in the list specified by the user is used as a reference to which all subsequent scans are realigned. The reference scan does not have to be the first chronologically and it may be wise to chose a "representative scan" in this role.

The aim is primarily to remove movement artefact in fMRI and PET time-series (or more generally longitudinal studies) [4]. The headers are modified for each of the input images, such that they reflect the relative orientations of the data. The details of the transformation are displayed in the results window as plots of translation and rotation. A set of realignment parameters are saved for each session, named `rp_*.txt`. After realignment, the images are resliced such that they match the first image selected voxel-for-voxel. The resliced images are named the same as the originals, except that they are prefixed by 'r'.

**2.3.1 Data**

Add new sessions for this subject. In the coregistration step, the sessions are first realigned to each other, by aligning the first scan from each session to the first scan of the first session. Then the images within each session are aligned to the first image of the session. The parameter estimation is performed this way because it is assumed (rightly or not) that there may be systematic differences in the images between sessions.

**Session**

Select scans for this session. In the coregistration step, the sessions are first realigned to each other, by aligning the first scan from each session to the first scan of the first session. Then the images within each session are aligned to the first image of the session. The parameter estimation is performed this way because it is assumed (rightly or not) that there may be systematic differences in the images between sessions.

**2.3.2 Estimation Options**

Various registration options. If in doubt, simply keep the default values.

**Quality**

Quality versus speed trade-off. Highest quality (1) gives most precise results, whereas lower qualities gives faster realignment. The idea is that some voxels contribute little to the estimation of the realignment parameters. This parameter is involved in selecting the number of voxels that are used.

**Separation**

The separation (in mm) between the points sampled in the reference image. Smaller sampling distances gives more accurate results, but will be slower.

**Smoothing (FWHM)**

The FWHM of the Gaussian smoothing kernel (mm) applied to the images before estimating the realignment parameters.

\* PET images typically use a 7 mm kernel.

\* MRI images typically use a 5 mm kernel.



### Num Passes

Register to first: Images are registered to the first image in the series. Register to mean: A two pass procedure is used in order to register the images to the mean of the images after the first realignment.

PET images are typically registered to the mean. This is because PET data are more noisy than fMRI and there are fewer of them, so time is less of an issue.

MRI images are typically registered to the first image. The more accurate way would be to use a two pass procedure, but this probably wouldn't improve the results so much and would take twice as long to run.

### Interpolation

The method by which the images are sampled when estimating the optimum transformation. Higher degree interpolation methods provide the better interpolation, but they are slower because they use more neighbouring voxels [93, 94, 95].

### Wrapping

This indicates which directions in the volumes the values should wrap around in. For example, in MRI scans, the images wrap around in the phase encode direction, so (e.g.) the subject's nose may poke into the back of the subject's head. These are typically:

No wrapping - for PET or images that have already been spatially transformed. Also the recommended option if you are not really sure.

Wrap in Y - for (un-resliced) MRI where phase encoding is in the Y direction (voxel space).

### Weighting

The option of providing a weighting image to weight each voxel of the reference image differently when estimating the realignment parameters. The weights are proportional to the inverses of the standard deviations. This would be used, for example, when there is a lot of extra-brain motion - e.g., during speech, or when there are serious artifacts in a particular region of the images.

## 2.3.3 Reslice Options

Various reslicing options. If in doubt, simply keep the default values.

### Resliced images

All Images (1..n) : This reslices all the images - including the first image selected - which will remain in its original position.

Images 2..n : Reslices images 2..n only. Useful for if you wish to reslice (for example) a PET image to fit a structural MRI, without creating a second identical MRI volume.

All Images + Mean Image : In addition to reslicing the images, it also creates a mean of the resliced image.

Mean Image Only : Creates the mean resliced image only.

### Interpolation

The method by which the images are sampled when being written in a different space. Nearest Neighbour is fastest, but not recommended for image realignment. Trilinear Interpolation is probably OK for PET, but not so suitable for fMRI because higher degree interpolation generally gives better results [93, 94, 95]. Although higher degree methods provide better interpolation, but they are slower because they use more neighbouring voxels. Fourier Interpolation [24, 20] is another option, but note that it is only implemented for purely rigid body transformations. Voxel sizes must all be identical and isotropic.

**Wrapping**

This indicates which directions in the volumes the values should wrap around in. For example, in MRI scans, the images wrap around in the phase encode direction, so (e.g.) the subject's nose may poke into the back of the subject's head. These are typically:

No wrapping - for PET or images that have already been spatially transformed.

Wrap in Y - for (un-resliced) MRI where phase encoding is in the Y direction (voxel space).

**Masking**

Because of subject motion, different images are likely to have different patterns of zeros from where it was not possible to sample data. With masking enabled, the program searches through the whole time series looking for voxels which need to be sampled from outside the original images. Where this occurs, that voxel is set to zero for the whole set of images (unless the image format can represent NaN, in which case NaNs are used where possible).

**Filename Prefix**

Specify the string to be prepended to the filenames of the resliced image file(s). Default prefix is 'r'.

# Chapter 3

## Realign & Unwarp

### Contents

---

<b>3.1 Data</b> .....	<b>34</b>
3.1.1 Session .....	34
<b>3.2 Estimation Options</b> .....	<b>34</b>
3.2.1 Quality .....	34
3.2.2 Separation .....	34
3.2.3 Smoothing (FWHM) .....	34
3.2.4 Num Passes .....	34
3.2.5 Interpolation .....	35
3.2.6 Wrapping .....	35
3.2.7 Weighting .....	35
<b>3.3 Unwarp Estimation Options</b> .....	<b>35</b>
3.3.1 Basis Functions .....	35
3.3.2 Regularisation .....	35
3.3.3 Reg. Factor .....	35
3.3.4 Jacobian deformations .....	35
3.3.5 First-order effects .....	36
3.3.6 Second-order effects .....	36
3.3.7 Smoothing for unwarp (FWHM) .....	36
3.3.8 Re-estimate movement params .....	36
3.3.9 Number of Iterations .....	36
3.3.10 Taylor expansion point .....	36
<b>3.4 Unwarp Reslicing Options</b> .....	<b>36</b>
3.4.1 Resliced images (unwarp)? .....	36
3.4.2 Interpolation .....	37
3.4.3 Wrapping .....	37
3.4.4 Masking .....	37
3.4.5 Filename Prefix .....	37

---

Within-subject registration and unwarping of time series.

The realignment part of this routine realigns a time-series of images acquired from the same subject using a least squares approach and a 6 parameter (rigid body) spatial transformation. The first image in the list specified by the user is used as a reference to which all subsequent scans are realigned. The reference scan does not have to be the first chronologically and it may be wise to choose a "representative scan" in this role.

The aim is primarily to remove movement artefact in fMRI and PET time-series (or more generally longitudinal studies). This affects the header of each of the input images which contains details about the voxel-to-world mapping. The details of the transformation are displayed in the results window as plots of translation and rotation. A set of realignment parameters are saved for each session, named `rp_*.txt`.

In the coregistration step, the sessions are first realigned to each other, by aligning the first scan from each session to the first scan of the first session. Then the images within each session are aligned to the first image of the session. The parameter estimation is performed this way because it is assumed (rightly or not) that there may be systematic differences in the images between sessions.

See `spm_uw_estimate.m` for a detailed description of the implementation.

Even after realignment there is considerable variance in fMRI time series that covary with, and is most probably caused by, subject movements [2]. It is also the case that this variance is typically large compared to experimentally induced variance. Anyone interested can include the estimated movement parameters as covariates in the design matrix, and take a look at an F-contrast encompassing those columns. It is quite dramatic. The result is loss of sensitivity, and if movements are correlated to task specificity. I.e. we may mistake movement induced variance for true activations. The problem is well known, and several solutions have been suggested. A quite pragmatic (and conservative) solution is to include the estimated movement parameters (and possibly squared) as covariates in the design matrix. Since we typically have loads of degrees of freedom in fMRI we can usually afford this. The problems occur when movements are correlated with the task, since the strategy above will discard "good" and "bad" variance alike (i.e. remove also "true" activations).

The "covariate" strategy described above was predicated on a model where variance was assumed to be caused by "spin history" effects, but will work pretty much equally good/bad regardless of what the true underlying cause is. Others have assumed that the residual variance is caused mainly by errors introduced by the interpolation kernel in the resampling step of the realignment. One has tried to solve this through higher order resampling (huge Sinc kernels, or k-space resampling). Unwarp is based on a different hypothesis regarding the residual variance. EPI images are not particularly faithful reproductions of the object, and in particular there are severe geometric distortions in regions where there is an air-tissue interface (e.g. orbitofrontal cortex and the anterior medial temporal lobes). In these areas in particular the observed image is a severely warped version of reality, much like a funny mirror at a fair ground. When one moves in front of such a mirror ones image will distort in different ways and ones head may change from very elongated to seriously flattened. If we were to take digital snapshots of the reflection at these different positions it is rather obvious that realignment will not suffice to bring them into a common space.

The situation is similar with EPI images, and an image collected for a given subject position will not be identical to that collected at another. We call this effect susceptibility-by-movement interaction. Unwarp is predicated on the assumption that the susceptibility-by-movement interaction is responsible for a sizable part of residual movement related variance.

Assume that we know how the deformations change when the subject changes position (i.e. we know the derivatives of the deformations with respect to subject position). That means that for a given time series and a given set of subject movements we should be able to predict the "shape changes" in the object and the ensuing variance in the time series. It also means that, in principle, we should be able to formulate the inverse problem, i.e. given the observed variance (after realignment) and known (estimated) movements we should be able to estimate how deformations change with subject movement. We have made an attempt at formulating such an inverse model, and at solving for the "derivative fields". A deformation field can be thought of as little vectors at each position in space showing how that particular location has been deflected. A "derivative field" is then the rate of change of those vectors with respect to subject movement. Given these "derivative fields" we should be able to remove the variance caused by the susceptibility-by-movement interaction. Since the underlying model is so restricted we would also expect experimentally induced variance to be preserved. Our experiments have also shown this to be true.

In theory it should be possible to estimate also the "static" deformation field, yielding an unwrapped (to some true geometry) version of the time series. In practise that doesn't really seem

to work. Hence, the method deals only with residual movement related variance induced by the susceptibility-by-movement interaction. This means that the time-series will be undistorted to some "average distortion" state rather than to the true geometry. If one wants additionally to address the issue of anatomical fidelity one should combine Unwarp with a measured fieldmap.

The description above can be thought of in terms of a Taylor expansion of the field as a function of subject movement. Unwarp alone will estimate the first (and optionally second, see below) order terms of this expansion. It cannot estimate the zeroth order term (the distortions common to all scans in the time series) since that doesn't introduce (almost) any variance in the time series. The measured fieldmap takes the role of the zeroth order term. Refer to the FieldMap toolbox and the documents FieldMap.man and FieldMap\_principles.man for a description of how to obtain fieldmaps in the format expected by Unwarp.

If we think of the field as a function of subject movement it should in principle be a function of six variables since rigid body movement has six degrees of freedom. However, the physics of the problem tells us that the field should not depend on translations nor on rotation in a plane perpendicular to the magnetic flux. Hence it should in principle be sufficient to model the field as a function of out-of-plane rotations (i.e. pitch and roll). One can object to this in terms of the effects of shimming (object no longer immersed in a homogenous field) that introduces a dependence on all movement parameters. In addition SPM/Unwarp cannot really tell if the transversal slices it is being passed are really perpendicular to the flux or not. In practice it turns out though that it is never (at least we haven't seen any case) necessary to include more than Pitch and Roll. This is probably because the individual movement parameters are typically highly correlated anyway, which in turn is probably because most heads that we scan are attached to a neck around which rotations occur. On the subject of Taylor expansion we should mention that there is the option to use a second-order expansion (through the defaults) interface. This implies estimating also the rate-of-change w.r.t. to some movement parameter of the rate-of-change of the field w.r.t. some movement parameter (colloquially known as a second derivative). It can be quite interesting to watch (and it is amazing that it is possible) but rarely helpful/necessary.

In the defaults there is also an option to include Jacobian intensity modulation when estimating the fields. "Jacobian intensity modulation" refers to the dilution/concentration of intensity that ensue as a consequence of the distortions. Think of a semi-transparent coloured rubber sheet that you hold against a white background. If you stretch a part of the sheet (induce distortions) you will see the colour fading in that particular area. In theory it is a brilliant idea to include also these effects when estimating the field (see e.g. Andersson et al, NeuroImage 20:870-888). In practice for this specific problem it is NOT a good idea.

It should be noted that this is a method intended to correct data afflicted by a particular problem. If there is little movement in your data to begin with this method will do you little good. If on the other hand there is appreciable movement in your data ( $>1\text{deg}$ ) it will remove some of that unwanted variance. If, in addition, movements are task related it will do so without removing all your "true" activations. The method attempts to minimise total (across the image volume) variance in the data set. It should be realised that while (for small movements) a rather limited portion of the total variance is removed, the susceptibility-by-movement interaction effects are quite localised to "problem" areas. Hence, for a subset of voxels in e.g. frontal-medial and orbitofrontal cortices and parts of the temporal lobes the reduction can be quite dramatic ( $>90$ ). The advantages of using Unwarp will also depend strongly on the specifics of the scanner and sequence by which your data has been acquired. When using the latest generation scanners distortions are typically quite small, and distortion-by-movement interactions consequently even smaller. A small check list in terms of distortions is

- a) Fast gradients->short read-out time->small distortions
- b) Low field (i.e.  $<3\text{T}$ )->small field changes->small distortions
- c) Low res (64x64)->short read-out time->small distortions
- d) SENSE/SMASH->short read-out time->small distortions

If you can tick off all points above chances are you have minimal distortions to begin with and Unwarp might not be of use to you.

## 3.1 Data

Data sessions to unwarp.

### 3.1.1 Session

Only add similar session data to a realign+unwarp branch, i.e., choose Data or Data+phase map for all sessions, but don't use them interchangeably.

In the coregistration step, the sessions are first realigned to each other, by aligning the first scan from each session to the first scan of the first session. Then the images within each session are aligned to the first image of the session. The parameter estimation is performed this way because it is assumed (rightly or not) that there may be systematic differences in the images between sessions.

### Images

Select scans for this session.

In the coregistration step, the sessions are first realigned to each other, by aligning the first scan from each session to the first scan of the first session. Then the images within each session are aligned to the first image of the session. The parameter estimation is performed this way because it is assumed (rightly or not) that there may be systematic differences in the images between sessions.

### Phase map (vdm\* file)

Select pre-calculated phase map, or leave empty for no phase correction. The vdm\* file is assumed to be already in alignment with the first scan of the first session.

## 3.2 Estimation Options

Various registration options that could be modified to improve the results. Whenever possible, the authors of SPM try to choose reasonable settings, but sometimes they can be improved.

### 3.2.1 Quality

Quality versus speed trade-off. Highest quality (1) gives most precise results, whereas lower qualities gives faster realignment. The idea is that some voxels contribute little to the estimation of the realignment parameters. This parameter is involved in selecting the number of voxels that are used.

### 3.2.2 Separation

The separation (in mm) between the points sampled in the reference image. Smaller sampling distances gives more accurate results, but will be slower.

### 3.2.3 Smoothing (FWHM)

The FWHM of the Gaussian smoothing kernel (mm) applied to the images before estimating the realignment parameters.

\* PET images typically use a 7 mm kernel.

\* MRI images typically use a 5 mm kernel.

### 3.2.4 Num Passes

Register to first: Images are registered to the first image in the series. Register to mean: A two pass procedure is used in order to register the images to the mean of the images after the first realignment.

\* PET images are typically registered to the mean.

\* MRI images are typically registered to the first image.

### 3.2.5 Interpolation

The method by which the images are sampled when estimating the optimum transformation. Higher degree interpolation methods provide the better interpolation, but they are slower because they use more neighbouring voxels [93, 94, 95].

### 3.2.6 Wrapping

These are typically:

- \* No wrapping - for images that have already been spatially transformed.
- \* Wrap in Y - for (un-resliced) MRI where phase encoding is in the Y direction (voxel space).

### 3.2.7 Weighting

The option of providing a weighting image to weight each voxel of the reference image differently when estimating the realignment parameters. The weights are proportional to the inverses of the standard deviations. For example, when there is a lot of extra-brain motion - e.g., during speech, or when there are serious artifacts in a particular region of the images.

## 3.3 Unwarp Estimation Options

Various registration & unwarping estimation options.

### 3.3.1 Basis Functions

Number of basis functions to use for each dimension. If the third dimension is left out, the order for that dimension is calculated to yield a roughly equal spatial cut-off in all directions. Default: [12 12 \*]

### 3.3.2 Regularisation

Unwarp looks for the solution that maximises the likelihood (minimises the variance) while simultaneously maximising the smoothness of the estimated field (c.f. Lagrange multipliers). This parameter determines how to balance the compromise between these (i.e. the value of the multiplier). Test it on your own data (if you can be bothered) or go with the defaults.

Regularisation of derivative fields is based on the regorder'th (spatial) derivative of the field. The choices are 0, 1, 2, or 3. Default: 1

### 3.3.3 Reg. Factor

Regularisation factor. Default: Medium.

### 3.3.4 Jacobian deformations

In the defaults there is also an option to include Jacobian intensity modulation when estimating the fields. "Jacobian intensity modulation" refers to the dilution/concentration of intensity that ensue as a consequence of the distortions. Think of a semi-transparent coloured rubber sheet that you hold against a white background. If you stretch a part of the sheet (induce distortions) you will see the colour fading in that particular area. In theory it is a brilliant idea to include also these effects when estimating the field (see e.g. Andersson et al, NeuroImage 20:870-888). In practice for this specific problem it is NOT a good idea. Default: No

### 3.3.5 First-order effects

Theoretically (ignoring effects of shimming) one would expect the field to depend only on subject out-of-plane rotations. Hence the default choice ("Pitch and Roll", i.e., [4 5]). Go with that unless you have very good reasons to do otherwise

Vector of first order effects to model. Movements to be modelled are referred to by number. 1= x translation; 2= y translation; 3= z translation 4 = x rotation, 5 = y rotation and 6 = z rotation.

To model pitch & roll enter: [4 5]

To model all movements enter: [1:6]

Otherwise enter a customised set of movements to model

### 3.3.6 Second-order effects

List of second order terms to model second derivatives of. This is entered as a vector of movement parameters similar to first order effects, or leave blank for NONE

Movements to be modelled are referred to by number:

1= x translation; 2= y translation; 3= z translation 4 = x rotation, 5 = y rotation and 6 = z rotation.

To model the interaction of pitch & roll enter: [4 5]

To model all movements enter: [1:6]

The vector will be expanded into an n x 2 matrix of effects. For example [4 5] will be expanded to:

```
[ 4 4
  4 5
  5 5 ]
```

### 3.3.7 Smoothing for unwarp (FWHM)

FWHM (mm) of smoothing filter applied to images prior to estimation of deformation fields.

### 3.3.8 Re-estimate movement params

Re-estimation means that movement-parameters should be re-estimated at each unwarping iteration. Default: Yes.

### 3.3.9 Number of Iterations

Maximum number of iterations. Default: 5.

### 3.3.10 Taylor expansion point

Point in position space to perform Taylor-expansion around. Choices are ('First', 'Last' or 'Average'). 'Average' should (in principle) give the best variance reduction. If a field-map acquired before the time-series is supplied then expansion around the 'First' MIGHT give a slightly better average geometric fidelity.

## 3.4 Unwarp Reslicing Options

Various registration & unwarping estimation options.

### 3.4.1 Resliced images (unwarp)?

All Images (1..n)

This reslices and unwarps all the images.

All Images + Mean Image

In addition to reslicing the images, it also creates a mean of the resliced images.



### 3.4.2 Interpolation

The method by which the images are sampled when being written in a different space. Nearest Neighbour is fastest, but not recommended for image realignment. Trilinear Interpolation is probably OK for PET, but not so suitable for fMRI because higher degree interpolation generally gives better results [93, 94, 95]. Although higher degree methods provide better interpolation, but they are slower because they use more neighbouring voxels.

### 3.4.3 Wrapping

These are typically:

- \* No wrapping - for images that have already been spatially transformed.
- \* Wrap in Y - for (un-resliced) MRI where phase encoding is in the Y direction (voxel space).

### 3.4.4 Masking

Because of subject motion, different images are likely to have different patterns of zeros from where it was not possible to sample data. With masking enabled, the program searches through the whole time series looking for voxels which need to be sampled from outside the original images. Where this occurs, that voxel is set to zero for the whole set of images (unless the image format can represent NaN, in which case NaNs are used where possible).

### 3.4.5 Filename Prefix

Specify the string to be prepended to the filenames of the smoothed image file(s). Default prefix is 'u'.



# Chapter 4

## Coregister

### Contents

---

<b>4.1 Coregister: Estimate</b>	<b>39</b>
4.1.1 Reference Image	39
4.1.2 Source Image	40
4.1.3 Other Images	40
4.1.4 Estimation Options	40
<b>4.2 Coregister: Reslice</b>	<b>40</b>
4.2.1 Image Defining Space	40
4.2.2 Images to Reslice	40
4.2.3 Reslice Options	40
<b>4.3 Coregister: Estimate &amp; Reslice</b>	<b>41</b>
4.3.1 Reference Image	41
4.3.2 Source Image	41
4.3.3 Other Images	41
4.3.4 Estimation Options	41
4.3.5 Reslice Options	42

---

Within-subject registration using a rigid-body model. A rigid-body transformation (in 3D) can be parameterised by three translations and three rotations about the different axes.

You get the options of estimating the transformation, reslicing images according to some rigid-body transformations, or estimating and applying rigid-body transformations.

### 4.1 Coregister: Estimate

The registration method used here is based on work by Collignon et al [19]. The original interpolation method described in this paper has been changed in order to give a smoother cost function. The images are also smoothed slightly, as is the histogram. This is all in order to make the cost function as smooth as possible, to give faster convergence and less chance of local minima.

At the end of coregistration, the voxel-to-voxel affine transformation matrix is displayed, along with the histograms for the images in the original orientations, and the final orientations. The registered images are displayed at the bottom.

Registration parameters are stored in the headers of the "source" and the "other" images.

#### 4.1.1 Reference Image

This is the image that is assumed to remain stationary (sometimes known as the target or template image), while the source image is moved to match it.

### 4.1.2 Source Image

This is the image that is jiggled about to best match the reference.

### 4.1.3 Other Images

These are any images that need to remain in alignment with the source image.

### 4.1.4 Estimation Options

Various registration options, which are passed to the Powell optimisation algorithm [89].

#### Objective Function

Registration involves finding parameters that either maximise or minimise some objective function. For inter-modal registration, use Mutual Information [19, 97], Normalised Mutual Information [92], or Entropy Correlation Coefficient [70]. For within modality, you could also use Normalised Cross Correlation.

#### Separation

The average distance between sampled points (in mm). Can be a vector to allow a coarse registration followed by increasingly fine ones.

#### Tolerances

The accuracy for each parameter. Iterations stop when differences between successive estimates are less than the required tolerance.

#### Histogram Smoothing

Gaussian smoothing to apply to the 256x256 joint histogram. Other information theoretic coregistration methods use fewer bins, but Gaussian smoothing seems to be more elegant.

## 4.2 Coregister: Reslice

Reslice images to match voxel-for-voxel with an image defining some space. The resliced images are named the same as the originals except that they are prefixed by 'r'.

### 4.2.1 Image Defining Space

This is analogous to the reference image. Images are resliced to match this image (providing they have been coregistered first).

### 4.2.2 Images to Reslice

These images are resliced to the same dimensions, voxel sizes, orientation etc as the space defining image.

### 4.2.3 Reslice Options

Various reslicing options.

### Interpolation

The method by which the images are sampled when being written in a different space. Nearest Neighbour is fastest, but not normally recommended. It can be useful for re-orienting images while preserving the original intensities (e.g. an image consisting of labels). Trilinear Interpolation is OK for PET, or realigned and re-sliced fMRI. If subject movement (from an fMRI time series) is included in the transformations then it may be better to use a higher degree approach. Note that higher degree B-spline interpolation [93, 94, 95] is slower because it uses more neighbours.

### Wrapping

These are typically:

No wrapping - for PET or images that have already been spatially transformed.

Wrap in Y - for (un-resliced) MRI where phase encoding is in the Y direction (voxel space).

### Masking

Because of subject motion, different images are likely to have different patterns of zeros from where it was not possible to sample data. With masking enabled, the program searches through the whole time series looking for voxels which need to be sampled from outside the original images. Where this occurs, that voxel is set to zero for the whole set of images (unless the image format can represent NaN, in which case NaNs are used where possible).

### Filename Prefix

Specify the string to be prepended to the filenames of the resliced image file(s). Default prefix is 'r'.

## 4.3 Coregister: Estimate & Reslice

The registration method used here is based on work by Collignon et al [19]. The original interpolation method described in this paper has been changed in order to give a smoother cost function. The images are also smoothed slightly, as is the histogram. This is all in order to make the cost function as smooth as possible, to give faster convergence and less chance of local minima.

At the end of coregistration, the voxel-to-voxel affine transformation matrix is displayed, along with the histograms for the images in the original orientations, and the final orientations. The registered images are displayed at the bottom.

Registration parameters are stored in the headers of the "source" and the "other" images. These images are also resliced to match the source image voxel-for-voxel. The resliced images are named the same as the originals except that they are prefixed by 'r'.

### 4.3.1 Reference Image

This is the image that is assumed to remain stationary (sometimes known as the target or template image), while the source image is moved to match it.

### 4.3.2 Source Image

This is the image that is jiggled about to best match the reference.

### 4.3.3 Other Images

These are any images that need to remain in alignment with the source image.

### 4.3.4 Estimation Options

Various registration options, which are passed to the Powell optimisation algorithm [89].

**Objective Function**

Registration involves finding parameters that either maximise or minimise some objective function. For inter-modal registration, use Mutual Information [19, 97], Normalised Mutual Information [92], or Entropy Correlation Coefficient [70]. For within modality, you could also use Normalised Cross Correlation.

**Separation**

The average distance between sampled points (in mm). Can be a vector to allow a coarse registration followed by increasingly fine ones.

**Tolerances**

The accuracy for each parameter. Iterations stop when differences between successive estimates are less than the required tolerance.

**Histogram Smoothing**

Gaussian smoothing to apply to the 256x256 joint histogram. Other information theoretic coregistration methods use fewer bins, but Gaussian smoothing seems to be more elegant.

**4.3.5 Reslice Options**

Various reslicing options.

**Interpolation**

The method by which the images are sampled when being written in a different space. Nearest Neighbour is fastest, but not normally recommended. It can be useful for re-orienting images while preserving the original intensities (e.g. an image consisting of labels). Trilinear Interpolation is OK for PET, or realigned and re-sliced fMRI. If subject movement (from an fMRI time series) is included in the transformations then it may be better to use a higher degree approach. Note that higher degree B-spline interpolation [93, 94, 95] is slower because it uses more neighbours.

**Wrapping**

These are typically:

No wrapping - for PET or images that have already been spatially transformed.

Wrap in Y - for (un-resliced) MRI where phase encoding is in the Y direction (voxel space).

**Masking**

Because of subject motion, different images are likely to have different patterns of zeros from where it was not possible to sample data. With masking enabled, the program searches through the whole time series looking for voxels which need to be sampled from outside the original images. Where this occurs, that voxel is set to zero for the whole set of images (unless the image format can represent NaN, in which case NaNs are used where possible).

**Filename Prefix**

Specify the string to be prepended to the filenames of the resliced image file(s). Default prefix is 'r'.

# Chapter 5

## Segment

### Contents

---

<b>5.1 Data</b> . . . . .	<b>44</b>
5.1.1 Channel . . . . .	44
<b>5.2 Tissues</b> . . . . .	<b>45</b>
5.2.1 Tissue . . . . .	45
<b>5.3 Warping &amp; MRF</b> . . . . .	<b>46</b>
5.3.1 MRF Parameter . . . . .	46
5.3.2 Clean Up . . . . .	46
5.3.3 Warping Regularisation . . . . .	47
5.3.4 Affine Regularisation . . . . .	47
5.3.5 Smoothness . . . . .	47
5.3.6 Sampling distance . . . . .	48
5.3.7 Deformation Fields . . . . .	48

---

This procedure is an extension of the old unified segmentation algorithm (and was known as "New Segment" in SPM8). The algorithm is essentially the same as that described in the Unified Segmentation paper, except for (i) a slightly different treatment of the mixing proportions, (ii) the use of an improved registration model, (iii) the ability to use multi-spectral data, (iv) an extended set of tissue probability maps, which allows a different treatment of voxels outside the brain. Some of the options in the toolbox do not yet work, and it has not yet been seamlessly integrated into the SPM8 software. Also, the extended tissue probability maps need further refinement. The current versions were crudely generated (by JA) using data that was kindly provided by Cynthia Jongen of the Imaging Sciences Institute at Utrecht, NL.

This function segments, bias corrects and spatially normalises - all in the same model [8]. Many investigators use tools within older versions of SPM for a technique that has become known as "optimised" voxel-based morphometry (VBM). VBM performs region-wise volumetric comparisons among populations of subjects. It requires the images to be spatially normalised, segmented into different tissue classes, and smoothed, prior to performing statistical tests [98, 73, 6, 7]. The "optimised" pre-processing strategy involved spatially normalising subjects' brain images to a standard space, by matching grey matter in these images, to a grey matter reference. The historical motivation behind this approach was to reduce the confounding effects of non-brain (e.g. scalp) structural variability on the registration. Tissue classification in older versions of SPM required the images to be registered with tissue probability maps. After registration, these maps represented the prior probability of different tissue classes being found at each location in an image. Bayes rule can then be used to combine these priors with tissue type probabilities derived from voxel intensities, to provide the posterior probability.

This procedure was inherently circular, because the registration required an initial tissue classification, and the tissue classification requires an initial registration. This circularity is resolved here by combining both components into a single generative model. This model also includes parameters that account for image intensity non-uniformity. Estimating the model parameters (for a maximum a posteriori solution) involves alternating among classification, bias correction and registration steps. This approach provides better results than simple serial applications of each component.

## 5.1 Data

Specify the number of different channels (for multi-spectral classification). If you have scans of different contrasts for each of the subjects, then it is possible to combine the information from them in order to improve the segmentation accuracy. Note that only the first channel of data is used for the initial affine registration with the tissue probability maps.

### 5.1.1 Channel

Specify a channel for processing. If multiple channels are used (eg PD & T2), then the same order of subjects must be specified for each channel and they must be in register (same position, size, voxel dims etc..). The different channels can be treated differently in terms of inhomogeneity correction etc. You may wish to correct some channels and save the corrected images, whereas you may wish not to do this for other channels.

#### Volumes

Select scans from this channel for processing. If multiple channels are used (eg T1 & T2), then the same order of subjects must be specified for each channel and they must be in register (same position, size, voxel dims etc..).

#### Bias regularisation

MR images are usually corrupted by a smooth, spatially varying artifact that modulates the intensity of the image (bias). These artifacts, although not usually a problem for visual inspection, can impede automated processing of the images.

An important issue relates to the distinction between intensity variations that arise because of bias artifact due to the physics of MR scanning, and those that arise due to different tissue properties. The objective is to model the latter by different tissue classes, while modelling the former with a bias field. We know a priori that intensity variations due to MR physics tend to be spatially smooth, whereas those due to different tissue types tend to contain more high frequency information. A more accurate estimate of a bias field can be obtained by including prior knowledge about the distribution of the fields likely to be encountered by the correction algorithm. For example, if it is known that there is little or no intensity non-uniformity, then it would be wise to penalise large values for the intensity non-uniformity parameters. This regularisation can be placed within a Bayesian context, whereby the penalty incurred is the negative logarithm of a prior probability for any particular pattern of non-uniformity.

Knowing what works best should be a matter of empirical exploration. For example, if your data has very little intensity non-uniformity artifact, then the bias regularisation should be increased. This effectively tells the algorithm that there is very little bias in your data, so it does not try to model it.

#### Bias FWHM

FWHM of Gaussian smoothness of bias. If your intensity non-uniformity is very smooth, then choose a large FWHM. This will prevent the algorithm from trying to model out intensity variation due to different tissue types. The model for intensity non-uniformity is one of i.i.d. Gaussian noise that has been smoothed by some amount, before taking the exponential. Note also that



smoother bias fields need fewer parameters to describe them. This means that the algorithm is faster for smoother intensity non-uniformities.

### Save Bias Corrected

This is the option to save a bias corrected version of your images from this channel, or/and the estimated bias field. MR images are usually corrupted by a smooth, spatially varying artifact that modulates the intensity of the image (bias). These artifacts, although not usually a problem for visual inspection, can impede automated processing of the images. The bias corrected version should have more uniform intensities within the different types of tissues.

## 5.2 Tissues

The data for each subject are classified into a number of different tissue types. The tissue types are defined according to tissue probability maps, which define the prior probability of finding a tissue type at a particular location. Typically, the order of tissues is grey matter, white matter, CSF, bone, soft tissue and air/background (if using tpm/TPM.nii).

### 5.2.1 Tissue

A number of options are available for each of the tissues. You may wish to save images of some tissues, but not others. If planning to use Dartel, then make sure you generate “imported” tissue class images of grey and white matter (and possibly others). Different numbers of Gaussians may be needed to model the intensity distributions of the various tissues.

#### Tissue probability map

Select the tissue probability image for this class. These should be maps of eg grey matter, white matter or cerebro-spinal fluid probability. A nonlinear deformation field is estimated that best overlays the tissue probability maps on the individual subjects’ image.

Rather than assuming stationary prior probabilities based upon mixing proportions, additional information is used, based on other subjects’ brain images. Priors are usually generated by registering a large number of subjects together, assigning voxels to different tissue types and averaging tissue classes over subjects. Three tissue classes are used: grey matter, white matter and cerebro-spinal fluid. A fourth class is also used, which is simply one minus the sum of the first three. These maps give the prior probability of any voxel in a registered image being of any of the tissue classes - irrespective of its intensity.

The model is refined further by allowing the tissue probability maps to be deformed according to a set of estimated parameters. This allows spatial normalisation and segmentation to be combined into the same model.

#### Num. Gaussians

The number of Gaussians used to represent the intensity distribution for each tissue class can be greater than one. In other words, a tissue probability map may be shared by several clusters. The assumption of a single Gaussian distribution for each class does not hold for a number of reasons. In particular, a voxel may not be purely of one tissue type, and instead contain signal from a number of different tissues (partial volume effects). Some partial volume voxels could fall at the interface between different classes, or they may fall in the middle of structures such as the thalamus, which may be considered as being either grey or white matter. Various other image segmentation approaches use additional clusters to model such partial volume effects. These generally assume that a pure tissue class has a Gaussian intensity distribution, whereas intensity distributions for partial volume voxels are broader, falling between the intensities of the pure classes. Unlike these partial volume segmentation approaches, the model adopted here simply assumes that the intensity distribution of each class may not be Gaussian, and assigns belonging probabilities according to these non-Gaussian distributions. Typical numbers of Gaussians could

be two for grey matter, two for white matter, two for CSF, three for bone, four for other soft tissues and two for air (background).

Note that if any of the Num. Gaussians is set to non-parametric, then a non-parametric approach will be used to model the tissue intensities. This may work for some images (eg CT), but not others - and it has not been optimised for multi-channel data. Note that it is likely to be especially problematic for images with poorly behaved intensity histograms due to aliasing effects that arise from having discrete values on the images.

### Native Tissue

The native space option allows you to produce a tissue class image ( $c^*$ ) that is in alignment with the original (see Figure 29.1). It can also be used for “importing” into a form that can be used with the Dartel toolbox ( $rc^*$ ).

### Warped Tissue

You can produce spatially normalised versions of the tissue class - both with ( $mwc^*$ ) and without ( $wc^*$ ) modulation (see below). These can be used for voxel-based morphometry. All you need to do is smooth them and do the stats.

“Modulation” is to compensate for the effect of spatial normalisation. When warping a series of images to match a template, it is inevitable that volumetric differences will be introduced into the warped images. For example, if one subject’s temporal lobe has half the volume of that of the template, then its volume will be doubled during spatial normalisation. This will also result in a doubling of the voxels labelled grey matter. In order to remove this confound, the spatially normalised grey matter (or other tissue class) is adjusted by multiplying by its relative volume before and after warping. If warping results in a region doubling its volume, then the correction will halve the intensity of the tissue label. This whole procedure has the effect of preserving the total amount of grey matter signal in the normalised partitions. Actually, in this version of SPM the warped data are not scaled by the Jacobian determinants when generating the “modulated” data. Instead, the original voxels are projected into their new location in the warped images. This exactly preserves the tissue count, but has the effect of introducing aliasing artifacts - especially if the original data are at a lower resolution than the warped images. Smoothing should reduce this artifact though.

Note also that the “unmodulated” data are generated slightly differently in this version of SPM. In this version, the projected data are corrected using a kind of smoothing procedure. This is not done exactly as it should be done (to save computational time), but it does a reasonable job. It also has the effect of extrapolating the warped tissue class images beyond the range of the original data. This extrapolation is not perfect, as it is only an estimate, but it may still be a good thing to do.

## 5.3 Warping & MRF

A number of warping options are provided, but the main one that you could consider changing is the one for specifying whether deformation fields or inverse deformation fields should be generated.

### 5.3.1 MRF Parameter

When tissue class images are written out, a few iterations of a simple Markov Random Field (MRF) cleanup procedure are run. This parameter controls the strength of the MRF. Setting the value to zero will disable the cleanup.

### 5.3.2 Clean Up

This uses a crude routine for extracting the brain from segmented images. It begins by taking the white matter, and eroding it a couple of times to get rid of any odd voxels. The algorithm continues on to do conditional dilations for several iterations, where the condition is based upon gray or white matter being present. This identified region is then used to clean up the grey and

white matter partitions. Note that the fluid class will also be cleaned, such that aqueous and vitreous humour in the eyeballs, as well as other assorted fluid regions (except CSF) will be removed.

If you find pieces of brain being chopped out in your data, then you may wish to disable or tone down the cleanup procedure. Note that the procedure uses a number of assumptions about what each tissue class refers to. If a different set of tissue priors are used, then this routine should be disabled.

### 5.3.3 Warping Regularisation

Registration involves simultaneously minimising two terms. One of these is a measure of similarity between the images (mean-squared difference in the current situation), whereas the other is a measure of the roughness of the deformations. This measure of roughness involves the sum of the following terms:

- \* Absolute displacements need to be penalised by a tiny amount. The first element encodes the amount of penalty on these. Ideally, absolute displacements should not be penalised, but it is necessary for technical reasons.

- \* The ‘membrane energy’ of the deformation is penalised (2nd element), usually by a relatively small amount. This penalises the sum of squares of the derivatives of the velocity field (ie the sum of squares of the elements of the Jacobian tensors).

- \* The ‘bending energy’ is penalised (3rd element). This penalises the sum of squares of the 2nd derivatives of the velocity.

- \* Linear elasticity regularisation is also included (4th and 5th elements). The first parameter ( $\mu$ ) is similar to that for linear elasticity, except it penalises the sum of squares of the Jacobian tensors after they have been made symmetric (by averaging with the transpose). This term essentially penalises length changes, without penalising rotations.

- \* The final term also relates to linear elasticity, and is the weight that denotes how much to penalise changes to the divergence of the velocities ( $\lambda$ ). This divergence is a measure of the rate of volumetric expansion or contraction.

The amount of regularisation determines the tradeoff between the terms. More regularisation gives smoother deformations, where the smoothness measure is determined by the bending energy of the deformations.

### 5.3.4 Affine Regularisation

The procedure is a local optimisation, so it needs reasonable initial starting estimates. Images should be placed in approximate alignment using the Display function of SPM before beginning. A Mutual Information affine registration with the tissue probability maps (D’Agostino et al, 2004) is used to achieve approximate alignment. Note that this step does not include any model for intensity non-uniformity. This means that if the procedure is to be initialised with the affine registration, then the data should not be too corrupted with this artifact. If there is a lot of intensity non-uniformity, then manually position your image in order to achieve closer starting estimates, and turn off the affine registration.

Affine registration into a standard space can be made more robust by regularisation (penalising excessive stretching or shrinking). The best solutions can be obtained by knowing the approximate amount of stretching that is needed (e.g. ICBM templates are slightly bigger than typical brains, so greater zooms are likely to be needed). For example, if registering to an image in ICBM/MNI space, then choose this option. If registering to a template that is close in size, then select the appropriate option for this.

### 5.3.5 Smoothness

For PET or SPECT, set this value to about 5 mm, or more if the images have smoother noise. For MRI, you can usually use a value of 0 mm. This is used to derive a fudge factor to account for correlations between neighbouring voxels. Smoother data have more spatial correlations, rendering the assumptions of the model inaccurate.

### 5.3.6 Sampling distance

This encodes the approximate distance between sampled points when estimating the model parameters. Smaller values use more of the data, but the procedure is slower and needs more memory. Determining the “best” setting involves a compromise between speed and accuracy.

### 5.3.7 Deformation Fields

Deformation fields can be saved to disk, and used by the Deformations Utility. For spatially normalising images to MNI space, you will need the forward deformation, whereas for spatially normalising (eg) GIFTI surface files, you’ll need the inverse. It is also possible to transform data in MNI space on to the individual subject, which also requires the inverse transform. Deformations are saved as .nii files, which contain three volumes to encode the x, y and z coordinates.

# Chapter 6

## Normalise

### Contents

---

<b>6.1 Normalise: Estimate</b>	<b>49</b>
6.1.1 Data	50
6.1.2 Estimation Options	50
<b>6.2 Normalise: Write</b>	<b>51</b>
6.2.1 Data	51
6.2.2 Writing Options	52
<b>6.3 Normalise: Estimate &amp; Write</b>	<b>52</b>
6.3.1 Data	52
6.3.2 Estimation Options	53
6.3.3 Writing Options	54

---

There are two components to spatial normalisation: There is the estimation part, whereby a deformation is estimated by deforming template data to match an individual scan; And there is the actual writing of the spatially normalised images, using the previously estimated deformation.

This is a vanilla approach to spatial normalisation.

It is not generally recommended for morphometric studies, or other studies of differences among populations.

The reason is that the atlas data will differ systematically from the data under study, which is likely to lead to an inherently biased set of findings.

### 6.1 Normalise: Estimate

Spatial normalisation is now done via the segmentation routine (which was known as “New Segment” in SPM8). The algorithm is essentially the same as that described in the Unified Segmentation paper [8], except for (i) a slightly different treatment of the mixing proportions, (ii) the use of an improved registration model, (iii) the ability to use multi-spectral data, (iv) an extended set of tissue probability maps, which allows a different treatment of voxels outside the brain.

Note that on a 32 bit computer, the most memory that SPM or any other program can use at any time is 4Gbytes (or sometimes only 2Gbytes). This is because the largest number that can be represented with 32 bits is 4,294,967,295, which limits how much memory may be addressed by any one process. Out of memory errors may occasionally be experienced when trying to work with large images. 64-bit computers can usually handle such cases.

If you encounter problems with spatial normalisation, it is advisable to use the Check reg button to see how well aligned the original data are with the MNI-space templates released with SPM. If mis-alignment is greater than about 3cm and 15 degrees, you could try to manually

re-position the images prior to attempting to align them. This may be done using the Display button.

### 6.1.1 Data

List of subjects. Images of each subject should be warped differently.

#### Subject

Data for this subject. The same parameters are used within subject.

**Image to Align** The image that the template (atlas) data is warped into alignment with. The result is a set of warps, which can be applied to this image, or any other image that is in register with it.

### 6.1.2 Estimation Options

Various settings for estimating deformations.

#### Bias regularisation

MR images are usually corrupted by a smooth, spatially varying artifact that modulates the intensity of the image (bias). These artifacts, although not usually a problem for visual inspection, can impede automated processing of the images.

An important issue relates to the distinction between intensity variations that arise because of bias artifact due to the physics of MR scanning, and those that arise due to different tissue properties. The objective is to model the latter by different tissue classes, while modelling the former with a bias field. We know a priori that intensity variations due to MR physics tend to be spatially smooth, whereas those due to different tissue types tend to contain more high frequency information. A more accurate estimate of a bias field can be obtained by including prior knowledge about the distribution of the fields likely to be encountered by the correction algorithm. For example, if it is known that there is little or no intensity non-uniformity, then it would be wise to penalise large values for the intensity non-uniformity parameters. This regularisation can be placed within a Bayesian context, whereby the penalty incurred is the negative logarithm of a prior probability for any particular pattern of non-uniformity.

Knowing what works best should be a matter of empirical exploration. For example, if your data has very little intensity non-uniformity artifact, then the bias regularisation should be increased. This effectively tells the algorithm that there is very little bias in your data, so it does not try to model it.

#### Bias FWHM

FWHM of Gaussian smoothness of bias. If your intensity non-uniformity is very smooth, then choose a large FWHM. This will prevent the algorithm from trying to model out intensity variation due to different tissue types. The model for intensity non-uniformity is one of i.i.d. Gaussian noise that has been smoothed by some amount, before taking the exponential. Note also that smoother bias fields need fewer parameters to describe them. This means that the algorithm is faster for smoother intensity non-uniformities.

#### Tissue probability map

Select the tissue probability atlas. These should contain probability maps of all the various tissues found in the image data (such that probabilities are greater than or equal to zero, and they sum to one at each voxel. A nonlinear deformation field is estimated that best overlays the atlas on the individual subjects' image.

### Affine Regularisation

The procedure is a local optimisation, so it needs reasonable initial starting estimates. Images should be placed in approximate alignment using the Display function of SPM before beginning. A Mutual Information affine registration with the tissue probability maps (D’Agostino et al, 2004) is used to achieve approximate alignment. Note that this step does not include any model for intensity non-uniformity. This means that if the procedure is to be initialised with the affine registration, then the data should not be too corrupted with this artifact. If there is a lot of intensity non-uniformity, then manually position your image in order to achieve closer starting estimates, and turn off the affine registration.

Affine registration into a standard space can be made more robust by regularisation (penalising excessive stretching or shrinking). The best solutions can be obtained by knowing the approximate amount of stretching that is needed (e.g. ICBM templates are slightly bigger than typical brains, so greater zooms are likely to be needed). For example, if registering to an image in ICBM/MNI space, then choose this option. If registering to a template that is close in size, then select the appropriate option for this.

### Warping Regularisation

The objective function for registering the tissue probability maps to the image to process, involves minimising the sum of two terms. One term gives a function of how probable the data is given the warping parameters. The other is a function of how probable the parameters are, and provides a penalty for unlikely deformations. Smoother deformations are deemed to be more probable. The amount of regularisation determines the tradeoff between the terms. Pick a value around one. However, if your normalised images appear distorted, then it may be an idea to increase the amount of regularisation (by an order of magnitude). More regularisation gives smoother deformations, where the smoothness measure is determined by the bending energy of the deformations.

### Smoothness

For PET or SPECT, set this value to about 5 mm, or more if the images have smoother noise. For MRI, you can usually use a value of 0 mm. This is used to derive a fudge factor to account for correlations between neighbouring voxels. Smoother data have more spatial correlations, rendering the assumptions of the model inaccurate.

### Sampling distance

This encodes the approximate distance between sampled points when estimating the model parameters. Smaller values use more of the data, but the procedure is slower and needs more memory. Determining the “best” setting involves a compromise between speed and accuracy.

## 6.2 Normalise: Write

Allows previously estimated warps (stored in “y\_”+image name+“\_sn.mat” files) to be applied to series of images.

### 6.2.1 Data

List of subjects. Images of each subject should be warped differently.

#### Subject

Data for this subject. The same parameters are used within subject.

**Deformation Field** Deformations can be thought of as vector fields, and represented by three-volume images. In SPM, deformation fields are saved in NIfTI format, with dimensions  $x_{dim} \times y_{dim} \times z_{dim} \times 1 \times 3$ . Each voxel contains the x, y and z mm coordinates of where the deformation points.

**Images to Write** These are the images for warping according to the estimated parameters. They can be any images that are in register with the image used to generate the deformation.

## 6.2.2 Writing Options

Various options for writing normalised images.

### Bounding box

The bounding box (in mm) of the volume which is to be written (relative to the anterior commissure).

### Voxel sizes

The voxel sizes (x, y & z, in mm) of the written normalised images.

### Interpolation

The method by which the images are sampled when being written in a different space. (Note that Inf or NaN values are treated as zero, rather than as missing data)

Nearest Neighbour:

- Fastest, but not normally recommended.

Trilinear Interpolation:

- OK for PET, realigned fMRI, or segmentations

B-spline Interpolation:

- Better quality (but slower) interpolation [93], especially with higher degree splines. Can produce values outside the original range (e.g. small negative values from an originally all positive image).

### Filename Prefix

Specify the string to be prepended to the filenames of the normalised image file(s). Default prefix is 'w'.

## 6.3 Normalise: Estimate & Write

Computes the warp that best aligns the template (atlas) to the individual's image, inverting it and writing the result to the file 'y\_'imasename'.nii'. This option also allows the contents of the 'y\_'imasename'.nii' files to be applied to a series of images.

Note that if you encounter problems with spatial normalisation, it is often advisable to use the Check reg button to see how well aligned the original data are with the MNI-space templates released with SPM. If mis-alignment is greater than about 3cm and 15 degrees, you could try to manually re-position the images. This may be done using the Display button.

### 6.3.1 Data

List of subjects. Images of each subject should be warped differently.

#### Subject

Data for this subject. The same parameters are used within subject.



**Image to Align** The image that the template (atlas) data is warped into alignment with. The result is a set of warps, which can be applied to this image, or any other image that is in register with it.

**Images to Write** These are the images for warping according to the estimated parameters. They can be any images that are in register with the image used to generate the deformation.

### 6.3.2 Estimation Options

Various settings for estimating deformations.

#### Bias regularisation

MR images are usually corrupted by a smooth, spatially varying artifact that modulates the intensity of the image (bias). These artifacts, although not usually a problem for visual inspection, can impede automated processing of the images.

An important issue relates to the distinction between intensity variations that arise because of bias artifact due to the physics of MR scanning, and those that arise due to different tissue properties. The objective is to model the latter by different tissue classes, while modelling the former with a bias field. We know a priori that intensity variations due to MR physics tend to be spatially smooth, whereas those due to different tissue types tend to contain more high frequency information. A more accurate estimate of a bias field can be obtained by including prior knowledge about the distribution of the fields likely to be encountered by the correction algorithm. For example, if it is known that there is little or no intensity non-uniformity, then it would be wise to penalise large values for the intensity non-uniformity parameters. This regularisation can be placed within a Bayesian context, whereby the penalty incurred is the negative logarithm of a prior probability for any particular pattern of non-uniformity.

Knowing what works best should be a matter of empirical exploration. For example, if your data has very little intensity non-uniformity artifact, then the bias regularisation should be increased. This effectively tells the algorithm that there is very little bias in your data, so it does not try to model it.

#### Bias FWHM

FWHM of Gaussian smoothness of bias. If your intensity non-uniformity is very smooth, then choose a large FWHM. This will prevent the algorithm from trying to model out intensity variation due to different tissue types. The model for intensity non-uniformity is one of i.i.d. Gaussian noise that has been smoothed by some amount, before taking the exponential. Note also that smoother bias fields need fewer parameters to describe them. This means that the algorithm is faster for smoother intensity non-uniformities.

#### Tissue probability map

Select the tissue probability atlas. These should contain probability maps of all the various tissues found in the image data (such that probabilities are greater than or equal to zero, and they sum to one at each voxel. A nonlinear deformation field is estimated that best overlays the atlas on the individual subjects' image.

#### Affine Regularisation

The procedure is a local optimisation, so it needs reasonable initial starting estimates. Images should be placed in approximate alignment using the Display function of SPM before beginning. A Mutual Information affine registration with the tissue probability maps (D'Agostino et al, 2004) is used to achieve approximate alignment. Note that this step does not include any model for intensity non-uniformity. This means that if the procedure is to be initialised with the affine registration, then the data should not be too corrupted with this artifact. If there is a lot of intensity non-uniformity, then manually position your image in order to achieve closer starting estimates, and turn off the affine registration.

Affine registration into a standard space can be made more robust by regularisation (penalising excessive stretching or shrinking). The best solutions can be obtained by knowing the approximate amount of stretching that is needed (e.g. ICBM templates are slightly bigger than typical brains, so greater zooms are likely to be needed). For example, if registering to an image in ICBM/MNI space, then choose this option. If registering to a template that is close in size, then select the appropriate option for this.

### Warping Regularisation

The objective function for registering the tissue probability maps to the image to process, involves minimising the sum of two terms. One term gives a function of how probable the data is given the warping parameters. The other is a function of how probable the parameters are, and provides a penalty for unlikely deformations. Smoother deformations are deemed to be more probable. The amount of regularisation determines the tradeoff between the terms. Pick a value around one. However, if your normalised images appear distorted, then it may be an idea to increase the amount of regularisation (by an order of magnitude). More regularisation gives smoother deformations, where the smoothness measure is determined by the bending energy of the deformations.

### Smoothness

For PET or SPECT, set this value to about 5 mm, or more if the images have smoother noise. For MRI, you can usually use a value of 0 mm. This is used to derive a fudge factor to account for correlations between neighbouring voxels. Smoother data have more spatial correlations, rendering the assumptions of the model inaccurate.

### Sampling distance

This encodes the approximate distance between sampled points when estimating the model parameters. Smaller values use more of the data, but the procedure is slower and needs more memory. Determining the “best” setting involves a compromise between speed and accuracy.

## 6.3.3 Writing Options

Various options for writing normalised images.

### Bounding box

The bounding box (in mm) of the volume which is to be written (relative to the anterior commissure).

### Voxel sizes

The voxel sizes (x, y & z, in mm) of the written normalised images.

### Interpolation

The method by which the images are sampled when being written in a different space. (Note that Inf or NaN values are treated as zero, rather than as missing data)

Nearest Neighbour:

- Fastest, but not normally recommended.

Trilinear Interpolation:

- OK for PET, realigned fMRI, or segmentations

B-spline Interpolation:

- Better quality (but slower) interpolation [93], especially with higher degree splines. Can produce values outside the original range (e.g. small negative values from an originally all positive image).

**Filename Prefix**

Specify the string to be prepended to the filenames of the normalised image file(s). Default prefix is 'w'.



# Chapter 7

## Smooth

### Contents

---

<b>7.1 Images to Smooth</b>	<b>57</b>
<b>7.2 FWHM</b>	<b>57</b>
<b>7.3 Data Type</b>	<b>57</b>
<b>7.4 Implicit masking</b>	<b>57</b>
<b>7.5 Filename Prefix</b>	<b>57</b>

---

This is for smoothing (or convolving) image volumes with a Gaussian kernel of a specified width. It is used as a preprocessing step to suppress noise and effects due to residual differences in functional and gyral anatomy during inter-subject averaging.

### 7.1 Images to Smooth

Specify the images to smooth. The smoothed images are written to the same subdirectories as the original images and are prefixed with a 's'. The prefix can be changed by an option setting.

### 7.2 FWHM

Specify the full-width at half maximum (FWHM) of the Gaussian smoothing kernel in mm. Three values should be entered, denoting the FWHM in the x, y and z directions.

### 7.3 Data Type

Data-type of output images. SAME indicates the same datatype as the original images.

### 7.4 Implicit masking

An "implicit mask" is a mask implied by a particular voxel value (0 for images with integer type, NaN for float images).

If set to 'Yes', the implicit masking of the input image is preserved in the smoothed image.

### 7.5 Filename Prefix

Specify the string to be prepended to the filenames of the smoothed image file(s). Default prefix is 's'.



Part III

fMRI Statistics





## Chapter 8

# fMRI model specification

Statistical analysis of fMRI data uses a mass-univariate approach based on General Linear Models (GLMs). It comprises the following steps (1) specification of the GLM design matrix, fMRI data files and filtering (2) estimation of GLM parameters using classical or Bayesian approaches and (3) interrogation of results using contrast vectors to produce Statistical Parametric Maps (SPMs) or Posterior Probability Maps (PPMs).

The design matrix defines the experimental design and the nature of hypothesis testing to be implemented. The design matrix has one row for each scan and one column for each effect or explanatory variable. (eg. regressor or stimulus function). You can build design matrices with separable session-specific partitions. Each partition may be the same (in which case it is only necessary to specify it once) or different.

Responses can be either event- or epoch related, the only distinction is the duration of the underlying input or stimulus function. Mathematically they are both modeled by convolving a series of delta (stick) or box functions (u), indicating the onset of an event or epoch with a set of basis functions. These basis functions model the hemodynamic convolution, applied by the brain, to the inputs. This convolution can be first-order or a generalized convolution modeled to second order (if you specify the Volterra option). The same inputs are used by the Hemodynamic model or Dynamic Causal Models which model the convolution explicitly in terms of hidden state variables.

Event-related designs may be stochastic or deterministic. Stochastic designs involve one of a number of trial-types occurring with a specified probability at successive intervals in time. These probabilities can be fixed (stationary designs) or time-dependent (modulated or non-stationary designs). The most efficient designs obtain when the probabilities of every trial type are equal. A critical issue in stochastic designs is whether to include null events. If you wish to estimate the evoked response to a specific event type (as opposed to differential responses) then a null event must be included (even if it is not modeled explicitly).

In SPM, analysis of data from multiple subjects typically proceeds in two stages using models at two “levels”. The “first level” models are used to implement a within-subject analysis. Typically there will be as many first level models as there are subjects. Analysis proceeds as described using the “Specify first level” and “Estimate” options. The results of these analyses can then be presented as “case studies”. More often, however, one wishes to make inferences about the population from which the subjects were drawn. This is an example of a “Random-Effects (RFX) analysis” (or, more properly, a mixed-effects analysis). In SPM, RFX analysis is implemented using the “summary-statistic” approach where contrast images from each subject are used as summary measures of subject responses. These are then entered as data into a “second level” model.

Figure 8.1 shows how the SPM graphics window appears during fMRI model specification.

### 8.1 Timing parameters

Specify various timing parameters needed to construct the design matrix. This includes the units of the design specification and the interscan interval.

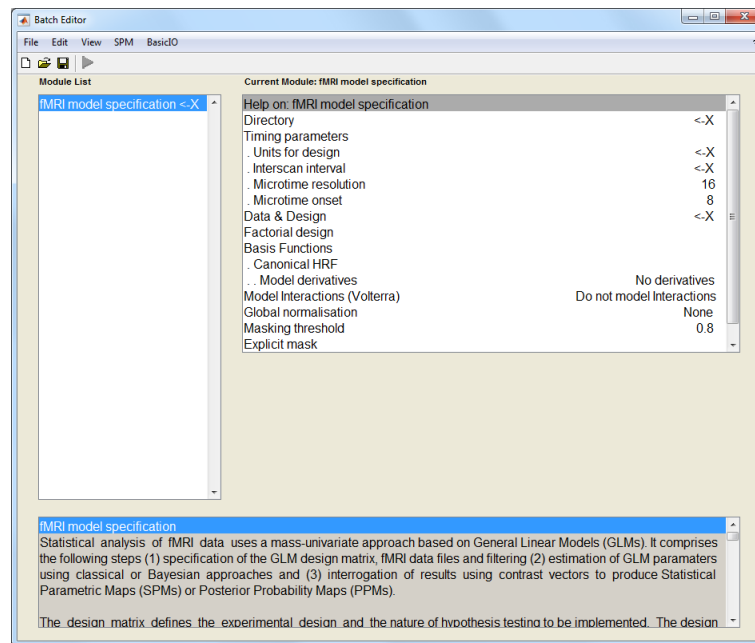


Figure 8.1: After starting SPM in fMRI mode and pressing the “Specify 1st-level” button, the SPM batch editor window should appear as above. The options for “fMRI model specification” can be examined by clicking on them. A single click will bring up some help text in the lower subwindow (not shown in the above graphic). Options highlighted with a “<-X” are mandatory and must be filled in by the user. Each of the options shown above is described in this chapter.

Also, with long TRs you may want to shift the regressors so that they are aligned to a particular slice. This is effected by changing the microtime resolution and onset.

### 8.1.1 Units for design

The onsets of events or blocks can be specified in either scans or seconds.

### 8.1.2 Interscan interval

Interscan interval, TR, (specified in seconds). This is the time between acquiring a plane of one volume and the same plane in the next volume. It is assumed to be constant throughout.

### 8.1.3 Microtime resolution

In Echo-Planar Imaging (EPI), data is acquired a plane at a time. To acquire a whole volume of data takes at least a second or two.

It is possible, however, that experimental events may occur between scan (volume) acquisition times. This can be specified when building your design matrix either by (i) specifying your design in scans and using non-integer values or (ii) specifying your design in seconds at a resolution greater than the TR.

SPM takes these timing specifications and builds its regressors using a ‘microtime’ time-scale. The microtime resolution,  $t$ , is the number of time-bins per scan.

Do not change this parameter unless you have a long TR and wish to shift regressors so that they are aligned to a particular slice.

### 8.1.4 Microtime onset

The microtime onset,  $t_0$ , is the first time-bin at which the regressors are resampled to coincide with data acquisition. If  $t_0 = 1$  then the regressors will be appropriate for the first slice. If you

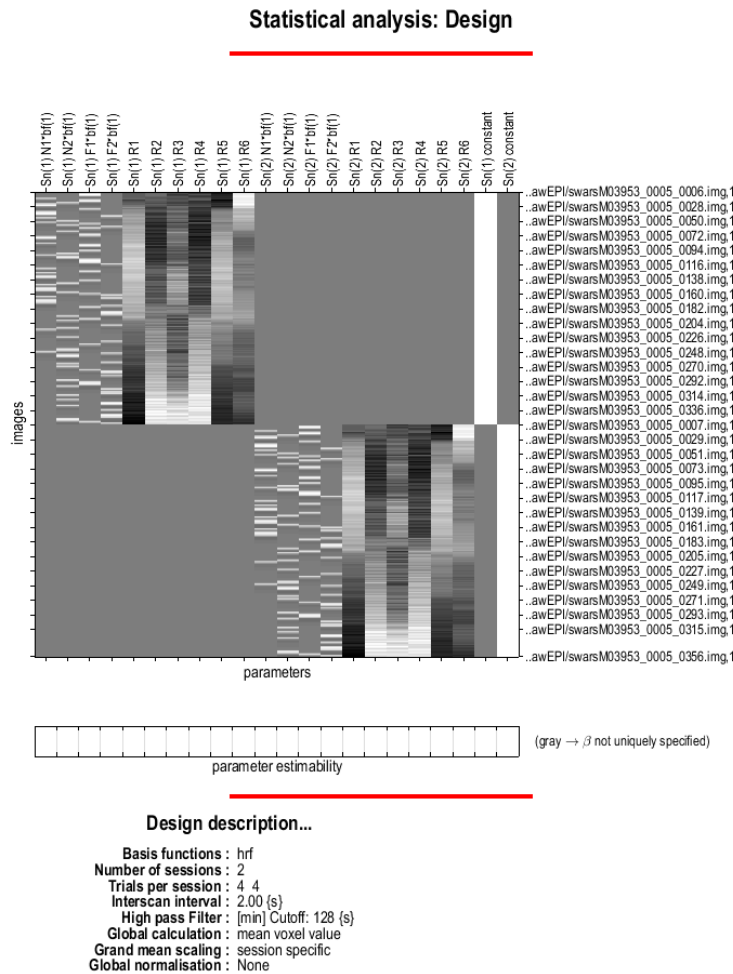


Figure 8.2: *Design matrix for fMRI data from two sessions. There are 4 experimental conditions and 6 regressors modelling the movement parameters for each session. The last two columns model the average activity in each session, giving a total of 22 regressors. There are 351 fMRI scans for each session. The overall design matrix therefore has 702 rows and 22 columns.*

want to temporally realign the regressors so that they match responses in the middle slice then make  $t_0 = t/2$  (assuming there is a negligible gap between volume acquisitions).

Do not change the default setting unless you have a long TR.

A typical use of the  $t$  and  $t_0$  parameters is to set them to correspond to the results of any slice timing correction you have made eg. if you have 24 slices and have made slice 12 the reference slice you would set  $t=24$ ,  $t_0=12$ .

## 8.2 Data & Design

The design matrix defines the experimental design and the nature of hypothesis testing to be implemented. The design matrix has one row for each scan and one column for each effect or explanatory variable. (e.g. regressor or stimulus function). Figure 8.2 shows an example of a design matrix.

You can build design matrices with separable session-specific partitions. Each partition may be the same (in which case it is only necessary to specify it once) or different. Responses can be either event- or epoch related, where the latter model involves prolonged and possibly time-varying responses to state-related changes in experimental conditions. Event-related response are

modelled in terms of responses to instantaneous events. Mathematically they are both modelled by convolving a series of delta (stick) or box-car functions, encoding the input or stimulus function, with a set of hemodynamic basis functions.

### 8.2.1 Subject/Session

The design matrix for fMRI data consists of one or more separable, session-specific partitions. These partitions are usually either one per subject, or one per fMRI scanning session for that subject.

#### Scans

Select the fMRI scans for this session. They must all have the same image dimensions, orientation, voxel size etc. This is implemented using SPM's file selector.

#### Conditions

You are allowed to combine both event- and epoch-related responses in the same model and/or regressor. Any number of condition (event or epoch) types can be specified. Epoch and event-related responses are modeled in exactly the same way by specifying their onsets [in terms of onset times] and their durations. Events are specified with a duration of 0. If you enter a single number for the durations it will be assumed that all trials conform to this duration. For factorial designs, one can later associate these experimental conditions with the appropriate levels of experimental factors.

**Condition** An array of input functions is constructed, specifying occurrence events or epochs (or both). These are convolved with a basis set at a later stage to give regressors that enter into the design matrix. Interactions of evoked responses with some parameter (time or a specified variate) enter at this stage as additional columns in the design matrix with each trial multiplied by the [expansion of the] trial-specific parameter. The 0th order expansion is simply the main effect in the first column.

**Name** Condition Name

**Onsets** Specify a vector of onset times for this condition type. This can be entered using the keyboard eg. typing in "100 300" and then hitting return or "100;300" or "[100,300]" or "[100,300]".

More usually, however, this specification takes place using variables that have been created before and loaded into matlab. For example, an `my_onsets` cell array<sup>1</sup> might exist in a file you created earlier called `my_design.mat`. You would then type `load my_design` at the matlab command prompt before pressing the 'Specify 1st-level' button.

You could then specify the onsets for condition 2 by typing in eg. `my_onsets{2}` instead of entering the numbers via the keyboard.

**Durations** Specify the event durations. Epoch and event-related responses are modeled in exactly the same way but by specifying their different durations. Events are specified with a duration of 0. If you enter a single number for the durations it will be assumed that all trials conform to this duration. If you have multiple different durations, then the number must match the number of onset times.

**Time Modulation** This option allows for the characterisation of nonstationary responses. Specifically, you can model either linear or nonlinear time effects. For example, 1st order modulation would model the stick functions and a linear change of the stick function heights over time. Higher order modulation will introduce further columns that contain the stick functions scaled by time squared, time cubed etc.

---

<sup>1</sup>Cell arrays are usually used in preference to matrices as different event types can then have different numbers of events.

**Parametric Modulations** The stick function itself can be modulated by some parametric variate (this can be time or some trial-specific variate like reaction time) modeling the interaction between the trial and the variate. The events can be modulated by zero or more parameters.

See [16, 14] for further details of parametric modulations.

### Multiple conditions

If you have multiple conditions then entering the details a condition at a time is very inefficient. This option can be used to load all the required information in one go.

You will need to create a `*.mat` file containing the relevant information. This `*.mat` file must include the following cell arrays: `names`, `onsets` and `durations` eg. `names{2}='SSent-DSpeak'`, `onsets{2}=[3 5 19 222]`, `durations{2}=[0 0 0 0]` contain the required details of the second condition. These cell arrays may be made available by your stimulus delivery program eg. CO-GENT. The duration vectors can contain a single entry if the durations are identical for all events.

You then need to use SPM's file selector to select this `*.mat` file.

### Regressors

Regressors are additional columns included in the design matrix, which may model effects that would not be convolved with the haemodynamic response. One such example would be the estimated movement parameters, which may confound the data.

#### Regressor

**Name** Enter name of regressor eg. First movement parameter

**Value** Enter the values that the regressor takes. This could also be, for example, the name of a variable in MATLAB's work space that you have previously loaded in from a file. This might be a subjects movement parameters or reaction times.

#### Multiple regressors

If you have multiple regressors eg. realignment parameters, then entering the details a regressor at a time is very inefficient. This option can be used to load all the required information in one go.

You will first need to create a `*.mat` file containing a matrix `R`. Each column of `R` will contain a different regressor. When SPM creates the design matrix the regressors will be named `R1`, `R2`, `R3`, ..etc.

You then need to use SPM's file selector to select this `*.mat` file.

#### High-pass filter

The default high-pass filter cutoff is 128 seconds. Slow signal drifts with a period longer than this will be removed. Use "Explore design" to ensure this cut-off is not removing too much experimental variance. This is described later in section 8.10. High-pass filtering is implemented using a residual forming matrix (i.e. it is not a convolution) and is simply a way to remove confounds without estimating their parameters explicitly. The constant term is also incorporated into this filter matrix.

## 8.3 Factorial design

If you have a factorial design then SPM can automatically generate the contrasts necessary to test for the main effects and interactions.

This includes the F-contrasts necessary to test for these effects at the within-subject level (first level) and the simple contrasts necessary to generate the contrast images for a between-subject (second-level) analysis.

To use this option, create as many factors as you need and provide a name and number of levels for each. SPM assumes that the condition numbers of the first factor change slowest, the second factor next slowest etc. It is best to write down the contingency table for your design to ensure this condition is met. This table relates the levels of each factor to the conditions.

For example, if you have 2-by-3 design your contingency table has two rows and three columns where the the first factor spans the rows, and the second factor the columns. The numbers of the conditions are 1,2,3 for the first row and 4,5,6 for the second.

See [52] for more information on SPM and factorial designs.

### 8.3.1 Factor

Add a new factor to your experimental design.

#### Name

Name of factor, eg. 'Repetition'

#### Levels

Enter number of levels for this factor, eg. 2

## 8.4 Basis Functions

SPM uses basis functions to model the hemodynamic response. This could be a single basis function or a set of functions. The most common choice is the 'Canonical HRF' with or without time and dispersion derivatives.

### 8.4.1 Canonical HRF

Canonical Hemodynamic Response Function (HRF). This is the default option. Contrasts of these effects have a physical interpretation and represent a parsimonious way of characterising event-related responses. This option is also useful if you wish to look separately at activations and deactivations. This is implemented using a t-contrast with a +1 or -1 entry over the canonical regressor.

#### Model derivatives

Model HRF Derivatives. The canonical HRF combined with time and dispersion derivatives comprise an 'informed' basis set, as the shape of the canonical response conforms to the hemodynamic response that is commonly observed. The incorporation of the derivative terms allow for variations in subject-to-subject and voxel-to-voxel responses. The time derivative allows the peak response to vary by plus or minus a second and the dispersion derivative allows the width of the response to vary by a similar amount.

A positive estimate of the time-derivative regression coefficient implies that the peak hemodynamic response occurs earlier than usual ie. than would be expected using just the canonical regressor. A positive estimate for the dispersion derivative implies a less dispersed response than usual.

The informed basis set requires an SPMF for inference. T-contrasts over just the canonical are perfectly valid but assume constant delay/dispersion. The informed basis set compares favourably with eg. FIR bases on many data sets [54].

### 8.4.2 Other basis sets

The other basis sets supported by SPM are

1. Fourier Set
2. Fourier Set (Hanning)

3. Gamma Functions
4. Finite Impulse Response (FIR)

For each of these options you must also specify the **window length** which is the length in seconds of the post-stimulus time window that the basis functions span. You must also specify the **order**, that is, how many basis functions to use.

Usually, an informed basis set should be sufficient for most data sets. If this does not provide a good fit to the data it may be worthwhile re-considering how the neuronal events are modelled ie. is the timing correct ? should events be split into subsets ?

Alternatively, the gamma basis functions are an interesting choice as a particular linear combination of them is actually used to specify the canonical HRF. The FIR approach is of interest as it is equivalent to the method of ‘selective averaging’. See [50] for further details.

## 8.5 Model Interactions (Volterra)

Generalized convolution of inputs,  $U$ , with basis set,  $bf$ .

For first order expansions the causes are simply convolved (e.g. stick functions) in  $U$  by the basis functions in  $bf$  to create a design matrix  $X$ . For second order expansions new entries appear that correspond to the interaction among the original causes. The basis functions for these effects are two dimensional and are used to assemble the second order kernel.

Interactions or response modulations can enter at two levels. Firstly the stick function itself can be modulated by some parametric variate. This can be time or some trial-specific variate like reaction time modeling the interaction between the trial and the variate. Secondly interactions among the trials themselves can be modeled using a Volterra series formulation that accommodates interactions over time (and therefore within and between trial types).

This last option is useful for accommodating nonlinearities in the hemodynamic response. For example, if two events occur within a second or so of each other then the hemodynamic response to the pair may be less than the sum of the responses to each event when occurring in isolation. This type of ‘sub-linear’ response can be modelled using Volterra kernels. See [38] for further details.

## 8.6 Directory

Select a directory where the SPM.mat file containing the specified design matrix will be written. If this directory already contains an SPM.mat file then SPM will warn you of this before overwriting it, when the specification job is run.

## 8.7 Global normalisation

SPM can normalise fMRI data in one of two ways. These are selected using the options ‘None’ (the default) and ‘Scaling’.

Both methods are based on first estimating the average within-brain fMRI signal,  $g_{ns}$ , where  $n$  denotes scan and  $s$  denotes session. If you select ‘Scaling’, SPM will multiply each fMRI value in scan  $n$  and session  $s$  by  $100/g_{ns}$ .

If you select “None” then SPM computes the grand mean value,  $g_s = \frac{\sum_{n=1}^N g_{ns}}{N}$  where  $N$  is the number of scans in that session. This is the fMRI signal averaged over all voxels within the brain and all time points within session  $s$ . SPM then implements “Session-specific grand mean scaling” by multiplying each fMRI data point in session  $s$  by  $100/g_s$ .

See [1] for further discussion of this issue.

## 8.8 Explicit mask

Specify an image for explicitly masking the analysis. A sensible option here is to use a segmentation of structural images to specify a within-brain mask. If you select that image as an explicit

mask then only those voxels in the brain will be analysed. This both speeds the estimation and restricts SPMs/PPMs to within-brain voxels. Alternatively, if such structural images are unavailable or no masking is required, then leave this field empty.

## 8.9 Serial correlations

Serial correlations in fMRI time series due to aliased biorhythms and unmodelled neuronal activity can be accounted for using an autoregressive AR(1) model during Classical (ReML) parameter estimation.

This estimate assumes the same correlation structure for each voxel, within each session. ReML estimates are then used to correct for non-sphericity during inference by adjusting the statistics and degrees of freedom appropriately. The discrepancy between estimated and actual correlations are greatest at low frequencies. Therefore specification of the high-pass filter is particularly important.

Serial correlation can be ignored if you choose the “none” option. Note that the above options only apply if you later specify that your model will be estimated using the Classical (ReML) approach. If you choose Bayesian estimation these options will be ignored. For Bayesian estimation, the choice of noise model (AR model order) is made under the estimation options. See [41, 83] for further discussion of these issues.

## 8.10 Reviewing your design

After you have completed the SPM “job” file for specifying your fMRI design, and have run it, you will then be able to review your design by pressing the “Review” button in SPM’s button window (the top-left window). This is particularly useful, for example, for checking that your experimental variance has not been removed by high-pass filtering, as shown in Figure 8.3.



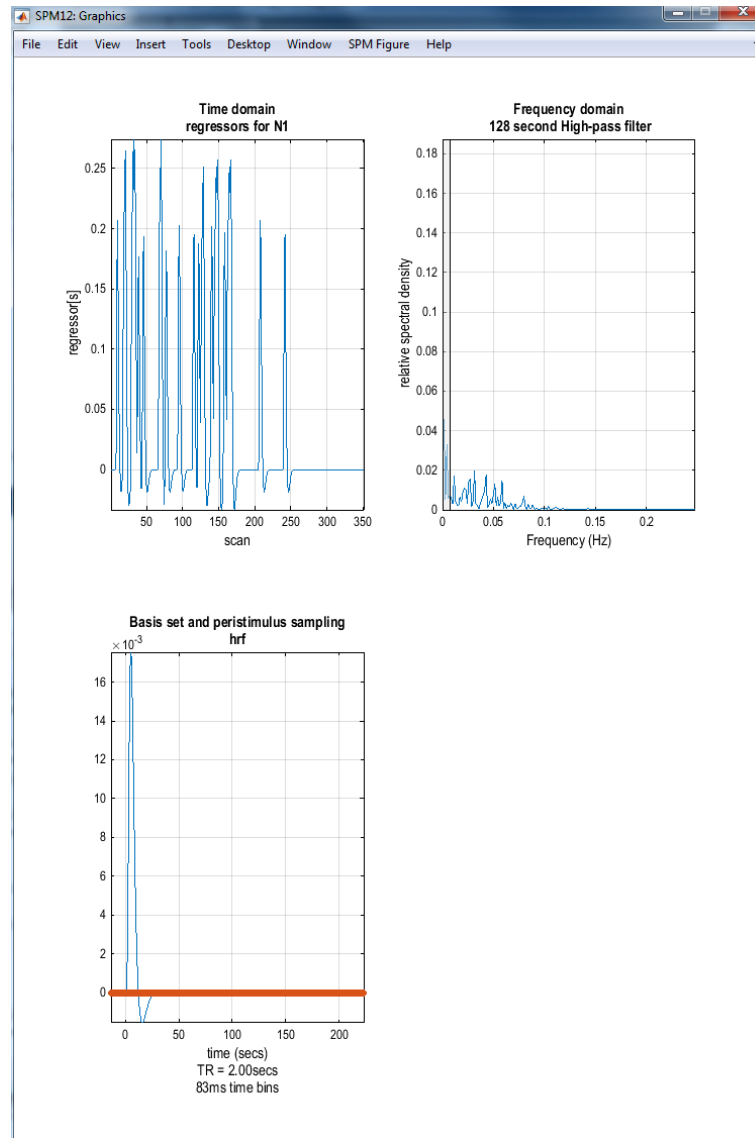


Figure 8.3: After pressing “Review”, selecting the pull-down ‘Design’ menu, *Explore->Session*, and selecting the regressor you wish to look at, you should get a plot similar to the one above. The top row shows time and frequency domain plots of the time-series corresponding to this regressor. In this particular case we have four events. Each event or “stick function” has been convolved with the hemodynamic response function shown in the bottom panel. The frequency domain graph is useful for checking that experimental variance is not removed by high-pass filtering. The grayed out section of the frequency plot shows those frequencies which are removed. For this regressor we have plenty of remaining experimental variance.



## Chapter 9

# fMRI model estimation

Model parameters can be estimated using classical (ReML - Restricted Maximum Likelihood) or Bayesian algorithms. After parameter estimation, the RESULTS button can be used to specify contrasts that will produce Statistical Parametric Maps (SPMs), Effect Size Maps (ESMs) or Posterior Probability Maps (PPMs) and tables of statistics.

### 9.1 Select SPM.mat

Select the `SPM.mat` file that contains the design specification. SPM will output the results of its analysis into this directory. This includes overwriting the `SPM.mat` file. When the estimation job is run, no warning will be given that the `SPM.mat` file will be overwritten. A warning is given at the specification stage. When it comes to estimation, SPM assumes that you've now sorted out your directory structures.

### 9.2 Method

There are three possible estimation procedures for fMRI models (1) classical (ReML) estimation of first or second level models, (2) Bayesian estimation of first level models and (3) Bayesian estimation of second level models. Option (2) uses a Variational Bayes (VB) algorithm introduced in SPM5. Option (3) uses the Empirical Bayes algorithm with global shrinkage priors that was also in SPM2.

To use option (3) you must have already estimated the model using option (1). That is, for second-level models you must run a ReML estimation before running a Bayesian estimation. This is not necessary for option (2). Bayesian estimation of 1st-level models using VB does not require a prior ReML estimation.

#### 9.2.1 Classical

Model parameters are estimated using Restricted Maximum Likelihood (ReML). This assumes the error correlation structure is the same at each voxel. This correlation can be specified using either an AR(1) or an Independent and Identically Distributed (IID) error model. These options are chosen at the model specification stage. ReML estimation should be applied to spatially smoothed functional images. See [41, 32] for further details of the ReML estimation scheme. After estimation, specific profiles of parameters are tested using a linear compound or contrast with the T or F statistic. The resulting statistical map constitutes an SPM. The SPMT/F is then characterised in terms of focal or regional differences by assuming that (under the null hypothesis) the components of the SPM (ie. residual fields) behave as smooth stationary Gaussian fields.

The rest of this chapter describes the Bayesian estimation options. So, please skip to the next chapter if you are interested only in classical estimation and inference.

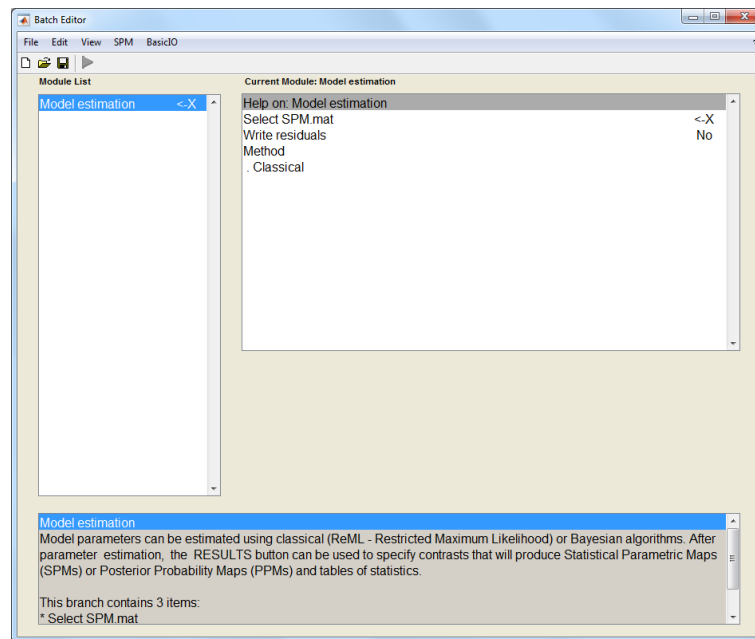


Figure 9.1: After starting SPM in fMRI mode and pressing the “Estimate” button, the SPM batch editor window should appear as above. The options for “fMRI model estimation” can be examined by clicking on them. A single click will bring up some help text in the lower subwindow (not shown in the above graphic). Options highlighted with a ‘<-X’ are mandatory and must be filled in by the user. Each of the options shown above is described in this chapter.

### 9.2.2 Bayesian 1st-level

Model parameters are estimated using Variational Bayes (VB). This allows you to specify spatial priors for regression coefficients and regularised voxel-wise AR(P) models for fMRI noise processes. The algorithm does not require functional images to be spatially smoothed. Estimation will take about 5 times longer than with the classical approach. This is why VB is not the default estimation option. The VB approach has been described in a number of papers [83, 87, 80, 81].

After estimation, contrasts are used to find regions with effects larger than a user-specified size eg. 1 per cent of the global mean signal. These effects are assessed statistically using a Posterior Probability Map (PPM) [39].

#### Analysis Space

Because estimation can be time consuming options are provided to analyse selected slices or clusters rather than the whole volume.

**Volume** A volume of data is analysed in “blocks”, which can be a slice or 3D subvolume, where the extent of each subvolume is determined using a graph partitioning algorithm. Enter the block type, i.e. “Slices” or “Subvolumes”.

**Block type** Enter the block type, i.e. “Slices” or “Subvolumes”.

**Slices** Enter Slice Numbers. This can be a single slice or multiple slices. If you select a single slice or only a few slices you must be aware of the interpolation options when, after estimation, displaying the estimated images eg. images of contrasts or AR maps. The default interpolation option may need to be changed to nearest neighbour (NN) (see bottom right hand of graphics window) for you slice maps to be visible.

**Slice numbers** Enter Slice Numbers.

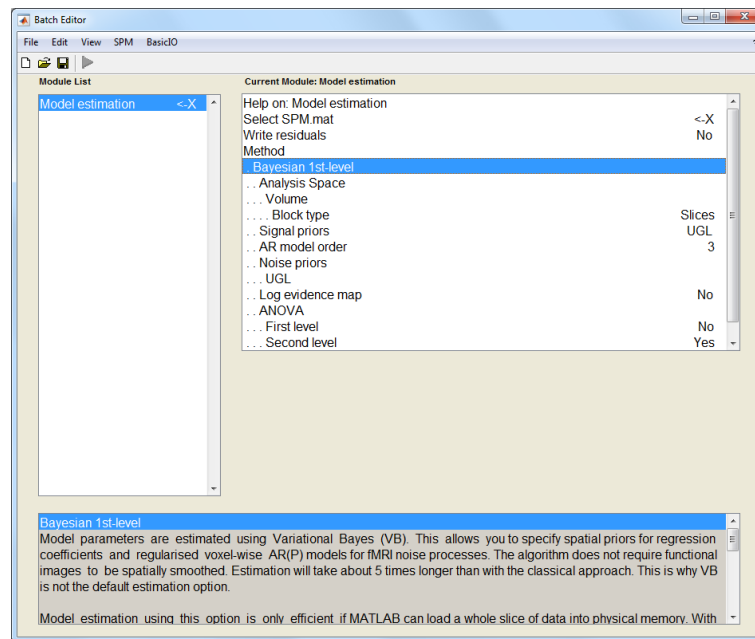


Figure 9.2: After choosing “Bayesian 1st-level” under “Method”, the SPM batch editor window should appear as above. Each of the options shown above is described in this chapter.

**Block type** Enter the block type, i.e. “Slices” or “Subvolume”.

**Clusters** Because estimation can be time consuming an option is provided to analyse selected clusters rather than the whole volume.

**Cluster mask** Select cluster image.

**Block type** Enter the block type, i.e. “Slices” or “Subvolumes”.

### Signal priors

- **[UGL] Unweighted Graph Laplacian.** This spatial prior is the recommended option. Regression coefficients at a given voxel are (softly) constrained to be similar to those at nearby voxels. The strength of this constraint is determined by a spatial precision parameter that is estimated from the data. Different regression coefficients have different spatial precisions allowing each putative experimental effect to have its own spatial regularity.
- **[GMRF] Gaussian Markov Random Field.** This is equivalent to a normalized UGL.
- **[LORETA] Low resolution Tomography Prior.** This is equivalent to UGL squared. It is a standard choice for EEG source localisation algorithms.
- **[WGL] Weighted Graph Laplacian.** This is a generalization of the UGL, where weights can be used to preserve “edges” of functional responses.
- **[Global] Global Shrinkage prior.** This is not a spatial prior in the sense that regression coefficients are constrained to be similar to neighboring voxels. Instead, the average effect over all voxels (global effect) is assumed to be zero and all regression coefficients are shrunk towards this value in proportion to the prior precision. This is the same prior that is used for Bayesian estimation at the second level models, except that here the prior precision is estimated separately for each slice.

- **[Uninformative] A flat prior.** Essentially, no prior information is used. If you select this option then VB reduces to Maximum Likelihood (ML) estimation. This option is useful if, for example, you do not wish to use a spatial prior but wish to take advantage of the voxel-wise AR(P) modelling of noise processes. In this case, you would apply the algorithm to images that have been spatially smoothed. For  $P=0$ , ML estimation in turn reduces to Ordinary Least Squares (OLS) estimates, and for  $P>0$  ML estimation is equivalent to a weighted least squares (WLS) but where the weights are different at each voxel (reflecting the different noise correlation at each voxel).

### AR model order

An AR model order of 3 is the default. Cardiac and respiratory artifacts are periodic in nature and therefore require an AR order of at least 2. In previous work, voxel-wise selection of the optimal model order showed that a value of 3 was the highest order required.

Higher model orders have little effect on the estimation time. If you select a model order of zero this corresponds to the assumption that the errors are IID. This AR specification overrides any choices that were made in the model specification stage.

Voxel-wise AR models are fitted separately for each session of data. For each session this therefore produces maps of AR(1), AR(2) etc coefficients in the output directory.

### Noise priors

There are five noise prior options here (1) UGL, (2) GMRF, (3) LORETA, (4) Tissue-type and (5) Robust.

**UGL** [UGL] Unweighted graph-Laplacian. This is the default option. This spatial prior is the same as that used for the regression coefficients. Spatial precisions are estimated separately for each AR coefficient eg. the AR(1) coefficient over space, AR(2) over space etc.

**GMRF** [GMRF] Gaussian Markov Random Field. See comments on GMRF priors for regression coefficients.

**LORETA** [LORETA] Low resolution Tomography Prior. See comments on LORETA priors for regression coefficients.

**Tissue-type** [Tissue-type] AR estimates at each voxel are biased towards typical values for that tissue type (eg. gray, white, CSF). If you select this option you will need to then select files that contain tissue type maps (see below). These are typically chosen to be Grey Matter, White Matter and CSF images derived from segmentation of registered structural scans.

Previous work has shown that there is significant variation in AR values with tissue type. However, GMRF priors have previously been favoured by Bayesian model comparison.

**Robust** Robust GLM. Uses Mixture of Gaussians noise model.

### Log evidence map

Computes the log evidence for each voxel

### ANOVA

Perform 1st or 2nd level Analysis of Variance.

**First level** This is implemented using Bayesian model comparison. For example, to test for the main effect of a factor two models are compared, one where the levels are represented using different regressors and one using the same regressor. This therefore requires explicit fitting of several models at each voxel and is computationally demanding (requiring several hours of computation). The recommended option is therefore NO.

To use this option you must have already specified your factorial design during the model specification stage.

**Second level** This option tells SPM to automatically generate the simple contrasts that are necessary to produce the contrast images for a second-level (between-subject) ANOVA. Naturally, these contrasts can also be used to characterise simple effects for each subject.

With the Bayesian estimation option it is recommended that contrasts are computed during the parameter estimation stage (see 'simple contrasts' below). The recommended option here is therefore YES.

To use this option you must have already specified your factorial design during the model specification stage.

If you wish to use these contrast images for a second-level analysis then you will need to spatially smooth them to take into account between-subject differences in functional anatomy ie. the fact that one persons V5 may be in a different position than anothers.

### Simple contrasts

“Simple” contrasts refers to a contrast that spans one-dimension ie. to assess an effect that is increasing or decreasing.

If you have a factorial design then the contrasts needed to generate the contrast images for a 2nd-level ANOVA (or to assess these simple effects within-subject) can be specified automatically using the ANOVA->Second level option.

When using the Bayesian estimation option it is computationally more efficient to compute the contrasts when the parameters are estimated. This is because estimated parameter vectors have potentially different posterior covariance matrices at different voxels and these matrices are not stored. If you compute contrasts post-hoc these matrices must be recomputed (an approximate reconstruction based on a Taylor series expansion is used). It is therefore recommended to specify as many contrasts as possible prior to parameter estimation.

If you wish to use these contrast images for a second-level analysis then you will need to spatially smooth them to take into account between-subject differences in functional anatomy ie. the fact that one persons V5 may be in a different position than anothers.

### Simple contrast

**Name** Name of contrast eg. “Positive Effect”.

**Contrast vector** These contrasts are used to generate PPMs which characterise effect sizes at each voxel. This is in contrast to SPMs in which eg. maps of t-statistics show the ratio of the effect size to effect variability (standard deviation). SPMs are therefore a-dimensional. This is not the case for PPMs as the size of the effect is of primary interest. Some care is therefore needed about the scaling of contrast vectors. For example, if you are interested in the differential effect size averaged over conditions then the contrast 0.5 0.5 -0.5 -0.5 would be more suitable than the 1 1 -1 -1 contrast which looks at the differential effect size summed over conditions.

### 9.2.3 Bayesian 2nd-level

Bayesian estimation of 2nd level models. This option uses the Empirical Bayes algorithm with global shrinkage priors that was previously implemented in SPM2. Use of the global shrinkage prior embodies a prior belief that, on average over all voxels, there is no net experimental effect. Some voxels will respond negatively and some positively with a variability determined by the prior precision. This prior precision can be estimated from the data using Empirical Bayes.

## 9.3 Output files

After estimation a number of files are written to the output directory. These are

- An `SPM.mat` file containing specification of the design and estimated model parameters

### 9.3.1 Classical 1st-level

For classical 1st-level models the following files are also produced

- Images of estimated regression coefficients `beta_000k.img` where  $k$  indexes the  $k$ th regression coefficient.
- An image of the variance of the error `ResMS.img`.
- An image `mask.img` indicating which voxels were included in the analysis.
- The image `RPV.img`, the estimated resels per voxel.
- If contrasts have been specified SPM also writes `con_000i.img` if the  $i$ th contrast is a t-contrast and the extra sum of squares image `ess_000i.img` if it is an F-contrast.

Type `help spm_spm` at the matlab command prompt for further information.

### 9.3.2 Bayesian 1st-level

For Bayesian 1st-level models the following files are also produced

- Images of estimated regression coefficients `Cbeta_000k.img` where  $k$  indexes the  $k$ th regression coefficient. These filenames are prefixed with a “C” indicating that these are the mean values of the ‘Conditional’ or ‘Posterior’ density.
- Images of error bars/standard deviations on the regression coefficients `SDbeta_000k.img`.
- An image of the standard deviation of the error `Sess1_SDerror.img`.
- An image `mask.img` indicating which voxels were included in the analysis.
- If a non-zero AR model order is specified then SPM also writes images `Sess1_AR_000p.img` where  $p$  indexes the  $p$ th AR coefficient.
- If contrasts have been specified SPM also writes `con_000i.img` and `con_sd_000i.img` which are the mean and standard deviation of the  $i$ th pre-defined contrast.

Each of these images can be inspected using the “Display” button. Type `help spm_spm_vb` at the MATLAB command prompt for further information.

## 9.4 Model comparison

Once you have estimated a model you can use SPM’s results button to look at the results. You can also extract fMRI data from regions of interest using the ROI button. You can then compare GLMs based on different hemodynamic basis sets using the Bayesian model evidence.

This is described in [81] and implemented using the command line option `spm_vb_roi_basis`. This requires a VOI filename (created using the ROI button) and an SPM data structure. Type `help spm_vb_roi_basis` at the MATLAB command prompt for further information. Figure 9.3 shows an example output from the function indicating that, for the data in this brain region, an informed basis set has the highest model evidence.



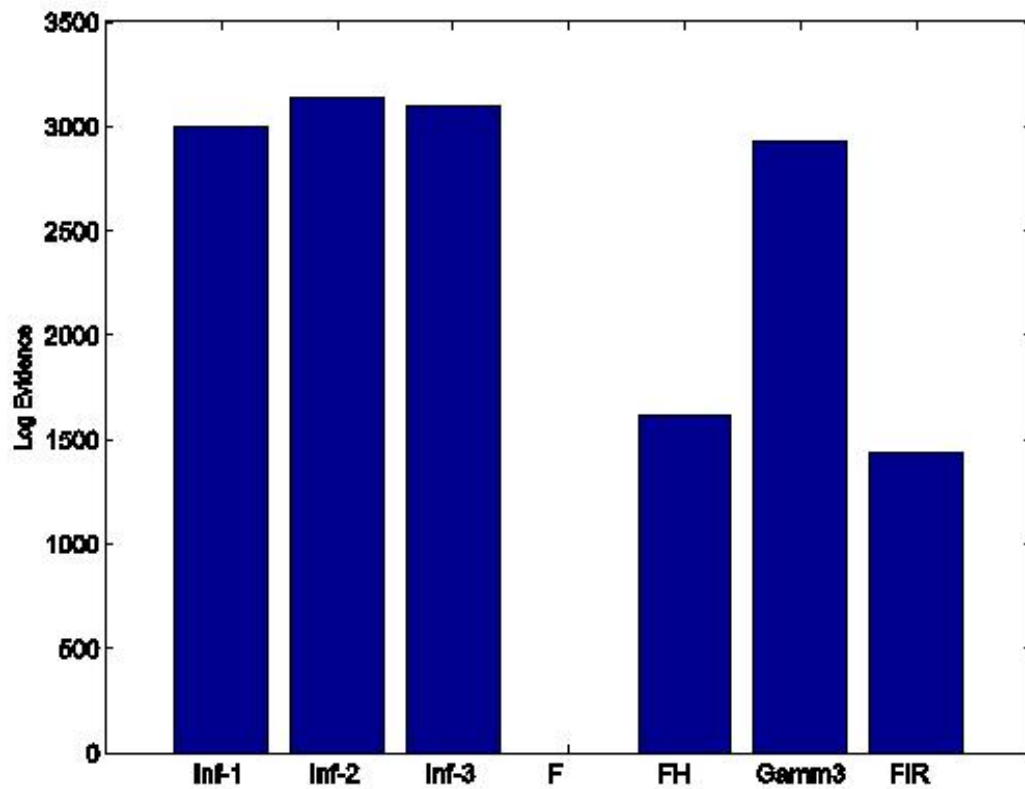


Figure 9.3: This plot shows the model evidence for a number of different hemodynamic basis sets: *Inf1* - Canonical HRF, *Inf2* - Canonical plus temporal derivative, *Inf3* - Canonical plus temporal and dispersion derivatives, *F* - Fourier, *FH* - Fourier with a Hanning Window, *Gamm3* - 3 Gamma basis functions and *FIR* - a Finite Impulse Response function. An informed basis set provides the best model of the data for the selected region.



# Chapter 10

## Factorial design specification

### Contents

---

<b>10.1 Directory</b> . . . . .	<b>80</b>
<b>10.2 Design</b> . . . . .	<b>80</b>
10.2.1 One-sample t-test . . . . .	80
10.2.2 Two-sample t-test . . . . .	80
10.2.3 Paired t-test . . . . .	81
10.2.4 Multiple regression . . . . .	81
10.2.5 One-way ANOVA . . . . .	82
10.2.6 One-way ANOVA - within subject . . . . .	83
10.2.7 Full factorial . . . . .	84
10.2.8 Flexible factorial . . . . .	86
<b>10.3 Covariates</b> . . . . .	<b>88</b>
10.3.1 Covariate . . . . .	88
<b>10.4 Multiple covariates</b> . . . . .	<b>88</b>
10.4.1 Covariates . . . . .	89
<b>10.5 Masking</b> . . . . .	<b>89</b>
10.5.1 Threshold masking . . . . .	89
10.5.2 Implicit Mask . . . . .	90
10.5.3 Explicit Mask . . . . .	90
<b>10.6 Global calculation</b> . . . . .	<b>90</b>
10.6.1 Omit . . . . .	90
10.6.2 User . . . . .	90
10.6.3 Mean . . . . .	91
<b>10.7 Global normalisation</b> . . . . .	<b>91</b>
10.7.1 Overall grand mean scaling . . . . .	91
10.7.2 Normalisation . . . . .	91

---

This interface configures the design matrix, describing the general linear model, data specification, and other parameters necessary for the statistical analysis. These parameters are saved in a configuration file (SPM.mat), which can then be passed on to `spm_spm.m` which estimates the design. This is achieved by pressing the 'Estimate' button. Inference on these estimated parameters is then handled by the SPM results section.

This interface is used for setting up analyses of PET data, morphometric data, or 'second level' ('random effects') fMRI data, where first level models can be used to produce appropriate summary data that are then used as raw data for the second-level analysis. For example, a simple

t-test on contrast images from the first-level turns out to be a random-effects analysis with random subject effects, inferring for the population based on a particular sample of subjects.

A separate interface handles design configuration for first level fMRI time series.

Various data and parameters need to be supplied to specify the design (1) the image files, (2) indicators of the corresponding condition/subject/group (2) any covariates, nuisance variables, or design matrix partitions (3) the type of global normalisation (if any) (4) grand mean scaling options (5) thresholds and masks defining the image volume to analyse. The interface supports a comprehensive range of options for all these parameters.

## 10.1 Directory

Select a directory where the SPM.mat file containing the specified design matrix will be written.

## 10.2 Design

### 10.2.1 One-sample t-test

One-sample t-test.

#### Scans

Select the images. They must all have the same image dimensions, orientation, voxel size etc.

### 10.2.2 Two-sample t-test

Two-sample t-test.

#### Group 1 scans

Select the images from sample 1. They must all have the same image dimensions, orientation, voxel size etc.

#### Group 2 scans

Select the images from sample 2. They must all have the same image dimensions, orientation, voxel size etc.

#### Independence

By default, the measurements are assumed to be independent between levels.

If you change this option to allow for dependencies, this will violate the assumption of sphericity. It would therefore be an example of non-sphericity. One such example would be where you had repeated measurements from the same subjects - it may then be the case that, over subjects, measure 1 is correlated to measure 2.

Restricted Maximum Likelihood (REML): The ensuing covariance components will be estimated using ReML in `spm_spm` (assuming the same for all responsive voxels) and used to adjust the statistics and degrees of freedom during inference. By default `spm_spm` will use weighted least squares to produce Gauss-Markov or Maximum likelihood estimators using the non-sphericity structure specified at this stage. The components will be found in `SPM.xVi` and enter the estimation procedure exactly as the serial correlations in fMRI models.

#### Variance

By default, the measurements in each level are assumed to have unequal variance.

This violates the assumption of 'sphericity' and is therefore an example of 'non-sphericity'.

This can occur, for example, in a 2nd-level analysis of variance, one contrast may be scaled differently from another. Another example would be the comparison of qualitatively different

dependent variables (e.g. normals vs. patients). Different variances (heteroscedasticity) induce different error covariance components that are estimated using restricted maximum likelihood (see below).

Restricted Maximum Likelihood (REML): The ensuing covariance components will be estimated using ReML in `spm_spm` (assuming the same for all responsive voxels) and used to adjust the statistics and degrees of freedom during inference. By default `spm_spm` will use weighted least squares to produce Gauss-Markov or Maximum likelihood estimators using the non-sphericity structure specified at this stage. The components will be found in `SPM.xVi` and enter the estimation procedure exactly as the serial correlations in fMRI models.

### Grand mean scaling

This option is for PET or VBM data (not second level fMRI).

Selecting YES will specify 'grand mean scaling by factor' which could be eg. 'grand mean scaling by subject' if the factor is 'subject'.

Since differences between subjects may be due to gain and sensitivity effects, AnCova by subject could be combined with "grand mean scaling by subject" to obtain a combination of between subject proportional scaling and within subject AnCova.

### ANCOVA

This option is for PET or VBM data (not second level fMRI).

Selecting YES will specify 'ANCOVA-by-factor' regressors. This includes eg. 'Ancova by subject' or 'Ancova by effect'. These options allow eg. different subjects to have different relationships between local and global measurements.

## 10.2.3 Paired t-test

Paired t-test.

### Pairs

**Pair** Add a new pair of scans to your experimental design.

**Scans [1,2]** Select the pair of images.

### Grand mean scaling

This option is for PET or VBM data (not second level fMRI).

Selecting YES will specify 'grand mean scaling by factor' which could be eg. 'grand mean scaling by subject' if the factor is 'subject'.

Since differences between subjects may be due to gain and sensitivity effects, AnCova by subject could be combined with "grand mean scaling by subject" to obtain a combination of between subject proportional scaling and within subject AnCova.

### ANCOVA

This option is for PET or VBM data (not second level fMRI).

Selecting YES will specify 'ANCOVA-by-factor' regressors. This includes eg. 'Ancova by subject' or 'Ancova by effect'. These options allow eg. different subjects to have different relationships between local and global measurements.

## 10.2.4 Multiple regression

Multiple regression.

### Scans

Select the images. They must all have the same image dimensions, orientation, voxel size etc.

**Covariates**

Covariates

**Covariate** Add a new covariate to your experimental design.

**Vector** Vector of covariate values.

**Name** Name of covariate.

**Centering** Centering refers to subtracting the mean (central) value from the covariate values, which is equivalent to orthogonalising the covariate with respect to the constant column.

Subtracting a constant from a covariate changes the beta for the constant term, but not that for the covariate. In the simplest case, centering a covariate in a simple regression leaves the slope unchanged, but converts the intercept from being the modelled value when the covariate was zero, to being the modelled value at the mean of the covariate, which is often more easily interpretable. For example, the modelled value at the subjects' mean age is usually more meaningful than the (extrapolated) value at an age of zero.

If a covariate value of zero is interpretable and/or you wish to preserve the values of the covariate then choose 'No centering'. You should also choose not to center if you have already subtracted some suitable value from your covariate, such as a commonly used reference level or the mean from another (e.g. larger) sample.

**Intercept**

By default, an intercept is always added to the model. If the covariates supplied by the user include a constant effect, the intercept may be omitted.

**10.2.5 One-way ANOVA**

One-way Analysis of Variance (ANOVA).

**Cells**

Enter the scans a cell at a time.

**Cell** Enter data for a cell in your design.

**Scans** Select the images for this cell. They must all have the same image dimensions, orientation, voxel size etc.

**Independence**

By default, the measurements are assumed to be independent between levels.

If you change this option to allow for dependencies, this will violate the assumption of sphericity. It would therefore be an example of non-sphericity. One such example would be where you had repeated measurements from the same subjects - it may then be the case that, over subjects, measure 1 is correlated to measure 2.

Restricted Maximum Likelihood (REML): The ensuing covariance components will be estimated using ReML in `spm_spm` (assuming the same for all responsive voxels) and used to adjust the statistics and degrees of freedom during inference. By default `spm_spm` will use weighted least squares to produce Gauss-Markov or Maximum likelihood estimators using the non-sphericity structure specified at this stage. The components will be found in `SPM.xVi` and enter the estimation procedure exactly as the serial correlations in fMRI models.

### Variance

By default, the measurements in each level are assumed to have unequal variance.

This violates the assumption of 'sphericity' and is therefore an example of 'non-sphericity'.

This can occur, for example, in a 2nd-level analysis of variance, one contrast may be scaled differently from another. Another example would be the comparison of qualitatively different dependent variables (e.g. normals vs. patients). Different variances (heteroscedasticity) induce different error covariance components that are estimated using restricted maximum likelihood (see below).

Restricted Maximum Likelihood (REML): The ensuing covariance components will be estimated using ReML in `spm_spm` (assuming the same for all responsive voxels) and used to adjust the statistics and degrees of freedom during inference. By default `spm_spm` will use weighted least squares to produce Gauss-Markov or Maximum likelihood estimators using the non-sphericity structure specified at this stage. The components will be found in `SPM.xVi` and enter the estimation procedure exactly as the serial correlations in fMRI models.

### Grand mean scaling

This option is for PET or VBM data (not second level fMRI).

Selecting YES will specify 'grand mean scaling by factor' which could be eg. 'grand mean scaling by subject' if the factor is 'subject'.

Since differences between subjects may be due to gain and sensitivity effects, AnCova by subject could be combined with "grand mean scaling by subject" to obtain a combination of between subject proportional scaling and within subject AnCova.

### ANCOVA

This option is for PET or VBM data (not second level fMRI).

Selecting YES will specify 'ANCOVA-by-factor' regressors. This includes eg. 'Ancova by subject' or 'Ancova by effect'. These options allow eg. different subjects to have different relationships between local and global measurements.

## 10.2.6 One-way ANOVA - within subject

One-way Analysis of Variance (ANOVA) - within subject.

### Subjects

**Subject** Enter data and conditions for a new subject.

**Scans** Select the images to be analysed. They must all have the same image dimensions, orientation, voxel size etc.

### Conditions

### Independence

By default, the measurements are assumed to be dependent between levels.

If you change this option to allow for dependencies, this will violate the assumption of sphericity. It would therefore be an example of non-sphericity. One such example would be where you had repeated measurements from the same subjects - it may then be the case that, over subjects, measure 1 is correlated to measure 2.

Restricted Maximum Likelihood (REML): The ensuing covariance components will be estimated using ReML in `spm_spm` (assuming the same for all responsive voxels) and used to adjust the statistics and degrees of freedom during inference. By default `spm_spm` will use weighted least squares to produce Gauss-Markov or Maximum likelihood estimators using the non-sphericity structure specified at this stage. The components will be found in `SPM.xVi` and enter the estimation procedure exactly as the serial correlations in fMRI models.

## Variance

By default, the measurements in each level are assumed to have unequal variance.

This violates the assumption of 'sphericity' and is therefore an example of 'non-sphericity'.

This can occur, for example, in a 2nd-level analysis of variance, one contrast may be scaled differently from another. Another example would be the comparison of qualitatively different dependent variables (e.g. normals vs. patients). Different variances (heteroscedasticity) induce different error covariance components that are estimated using restricted maximum likelihood (see below).

Restricted Maximum Likelihood (REML): The ensuing covariance components will be estimated using ReML in `spm_spm` (assuming the same for all responsive voxels) and used to adjust the statistics and degrees of freedom during inference. By default `spm_spm` will use weighted least squares to produce Gauss-Markov or Maximum likelihood estimators using the non-sphericity structure specified at this stage. The components will be found in `SPM.xVi` and enter the estimation procedure exactly as the serial correlations in fMRI models.

## Grand mean scaling

This option is for PET or VBM data (not second level fMRI).

Selecting YES will specify 'grand mean scaling by factor' which could be eg. 'grand mean scaling by subject' if the factor is 'subject'.

Since differences between subjects may be due to gain and sensitivity effects, AnCova by subject could be combined with "grand mean scaling by subject" to obtain a combination of between subject proportional scaling and within subject AnCova.

## ANCOVA

This option is for PET or VBM data (not second level fMRI).

Selecting YES will specify 'ANCOVA-by-factor' regressors. This includes eg. 'Ancova by subject' or 'Ancova by effect'. These options allow eg. different subjects to have different relationships between local and global measurements.

### 10.2.7 Full factorial

This option is best used when you wish to test for all main effects and interactions in one-way, two-way or three-way ANOVAs. Design specification proceeds in 2 stages. Firstly, by creating new factors and specifying the number of levels and name for each. Nonsphericity, ANOVA-by-factor and scaling options can also be specified at this stage. Secondly, scans are assigned separately to each cell. This accommodates unbalanced designs.

For example, if you wish to test for a main effect in the population from which your subjects are drawn and have modelled that effect at the first level using K basis functions (eg. K=3 informed basis functions) you can use a one-way ANOVA with K-levels. Create a single factor with K levels and then assign the data to each cell eg. canonical, temporal derivative and dispersion derivative cells, where each cell is assigned scans from multiple subjects.

SPM will also automatically generate the contrasts necessary to test for all main effects and interactions.

## Factors

Specify your design a factor at a time.

**Factor** Add a new factor to your experimental design.

**Name** Name of factor, eg. 'Repetition'.

**Levels** Enter number of levels for this factor, eg. 2.



**Independence** By default, the measurements are assumed to be independent between levels.

If you change this option to allow for dependencies, this will violate the assumption of sphericity. It would therefore be an example of non-sphericity. One such example would be where you had repeated measurements from the same subjects - it may then be the case that, over subjects, measure 1 is correlated to measure 2.

**Restricted Maximum Likelihood (REML):** The ensuing covariance components will be estimated using ReML in `spm_spm` (assuming the same for all responsive voxels) and used to adjust the statistics and degrees of freedom during inference. By default `spm_spm` will use weighted least squares to produce Gauss-Markov or Maximum likelihood estimators using the non-sphericity structure specified at this stage. The components will be found in `SPM.xVi` and enter the estimation procedure exactly as the serial correlations in fMRI models.

**Variance** By default, the measurements in each level are assumed to have unequal variance. This violates the assumption of 'sphericity' and is therefore an example of 'non-sphericity'.

This can occur, for example, in a 2nd-level analysis of variance, one contrast may be scaled differently from another. Another example would be the comparison of qualitatively different dependent variables (e.g. normals vs. patients). Different variances (heteroscedasticity) induce different error covariance components that are estimated using restricted maximum likelihood (see below).

**Restricted Maximum Likelihood (REML):** The ensuing covariance components will be estimated using ReML in `spm_spm` (assuming the same for all responsive voxels) and used to adjust the statistics and degrees of freedom during inference. By default `spm_spm` will use weighted least squares to produce Gauss-Markov or Maximum likelihood estimators using the non-sphericity structure specified at this stage. The components will be found in `SPM.xVi` and enter the estimation procedure exactly as the serial correlations in fMRI models.

**Grand mean scaling** This option is for PET or VBM data (not second level fMRI).

Selecting YES will specify 'grand mean scaling by factor' which could be eg. 'grand mean scaling by subject' if the factor is 'subject'.

Since differences between subjects may be due to gain and sensitivity effects, AnCova by subject could be combined with "grand mean scaling by subject" to obtain a combination of between subject proportional scaling and within subject AnCova.

**ANCOVA** This option is for PET or VBM data (not second level fMRI).

Selecting YES will specify 'ANCOVA-by-factor' regressors. This includes eg. 'Ancova by subject' or 'Ancova by effect'. These options allow eg. different subjects to have different relationships between local and global measurements.

## Cells

Enter the scans a cell at a time.

**Cell** Enter data for a cell in your design.

**Levels** Enter a vector or scalar that specifies which cell in the factorial design these images belong to. The length of this vector should correspond to the number of factors in the design

For example, length 2 vectors should be used for two-factor designs eg. the vector [2 3] specifies the cell corresponding to the 2nd-level of the first factor and the 3rd level of the 2nd factor.

**Scans** Select the images for this cell. They must all have the same image dimensions, orientation, voxel size etc.

## Generate contrasts

Automatically generate the contrasts necessary to test for all main effects and interactions.

### 10.2.8 Flexible factorial

Create a design matrix a block at a time by specifying which main effects and interactions you wish to be included.

This option is best used for one-way, two-way or three-way ANOVAs but where you do not wish to test for all possible main effects and interactions. This is perhaps most useful for PET where there is usually not enough data to test for all possible effects. Or for 3-way ANOVAs where you do not wish to test for all of the two-way interactions. A typical example here would be a group-by-drug-by-task analysis where, perhaps, only (i) group-by-drug or (ii) group-by-task interactions are of interest. In this case it is only necessary to have two-blocks in the design matrix - one for each interaction. The three-way interaction can then be tested for using a contrast that computes the difference between (i) and (ii).

Design specification then proceeds in 3 stages. Firstly, factors are created and names specified for each. Nonsphericity, ANOVA-by-factor and scaling options can also be specified at this stage.

Secondly, a list of scans is produced along with a factor matrix, I. This is an  $n_{\text{scan}} \times 4$  matrix of factor level indicators (see `xX.I` below). The first factor must be 'replication' but the other factors can be anything. Specification of I and the scan list can be achieved in one of two ways (a) the 'Specify All' option allows I to be typed in at the user interface or (more likely) loaded in from the matlab workspace. All of the scans are then selected in one go. (b) the 'Subjects' option allows you to enter scans a subject at a time. The corresponding experimental conditions (ie. levels of factors) are entered at the same time. SPM will then create the factor matrix I. This style of interface is similar to that available in SPM2.

Thirdly, the design matrix is built up a block at a time. Each block can be a main effect or a (two-way) interaction.

#### Factors

Specify your design a factor at a time.

**Factor** Add a new factor to your design.

If you are using the 'Subjects' option to specify your scans and conditions, you may wish to make use of the following facility. There are two reserved words for the names of factors. These are 'subject' and 'repl' (standing for replication). If you use these factor names then SPM will automatically create replication and/or subject factors without you having to type in an extra entry in the condition vector.

For example, if you wish to model Subject and Task effects (two factors), under Subjects->Subject->Conditions you should simply type in eg. `[1 2 1 2]` to specify just the 'Task' factor level, instead of, eg. for the 4th subject the matrix `[4 1;4 2;4 1;4 2]`.

**Name** Name of factor, eg. 'Repetition'.

**Independence** By default, the measurements are assumed to be independent between levels.

If you change this option to allow for dependencies, this will violate the assumption of sphericity. It would therefore be an example of non-sphericity. One such example would be where you had repeated measurements from the same subjects - it may then be the case that, over subjects, measure 1 is correlated to measure 2.

**Restricted Maximum Likelihood (REML):** The ensuing covariance components will be estimated using ReML in `spm_spm` (assuming the same for all responsive voxels) and used to adjust the statistics and degrees of freedom during inference. By default `spm_spm` will use weighted least squares to produce Gauss-Markov or Maximum likelihood estimators using the non-sphericity structure specified at this stage. The components will be found in `SPM.xVi` and enter the estimation procedure exactly as the serial correlations in fMRI models.

**Variance** By default, the measurements in each level are assumed to have unequal variance. This violates the assumption of 'sphericity' and is therefore an example of 'non-sphericity'.

This can occur, for example, in a 2nd-level analysis of variance, one contrast may be scaled differently from another. Another example would be the comparison of qualitatively different dependent variables (e.g. normals vs. patients). Different variances (heteroscedasticity) induce different error covariance components that are estimated using restricted maximum likelihood (see below).

**Restricted Maximum Likelihood (REML):** The ensuing covariance components will be estimated using ReML in `spm_spm` (assuming the same for all responsive voxels) and used to adjust the statistics and degrees of freedom during inference. By default `spm_spm` will use weighted least squares to produce Gauss-Markov or Maximum likelihood estimators using the non-sphericity structure specified at this stage. The components will be found in `SPM.xVi` and enter the estimation procedure exactly as the serial correlations in fMRI models.

**Grand mean scaling** This option is for PET or VBM data (not second level fMRI).

Selecting YES will specify 'grand mean scaling by factor' which could be eg. 'grand mean scaling by subject' if the factor is 'subject'.

Since differences between subjects may be due to gain and sensitivity effects, AnCova by subject could be combined with "grand mean scaling by subject" to obtain a combination of between subject proportional scaling and within subject AnCova.

**ANCOVA** This option is for PET or VBM data (not second level fMRI).

Selecting YES will specify 'ANCOVA-by-factor' regressors. This includes eg. 'Ancova by subject' or 'Ancova by effect'. These options allow eg. different subjects to have different relationships between local and global measurements.

## Specify Subjects or all Scans & Factors

### Subjects

**Subject** Enter data and conditions for a new subject.

**Scans** Select the images to be analysed. They must all have the same image dimensions, orientation, voxel size etc.

### Conditions

**Specify all** Specify (i) all scans in one go and (ii) all conditions using a factor matrix, `I`. This option is for 'power users'. The matrix `I` must have four columns and as many rows as scans. It has the same format as SPM's internal variable `SPM.xX.I`.

The first column of `I` denotes the replication number and entries in the other columns denote the levels of each experimental factor.

So, for eg. a two-factor design the first column denotes the replication number and columns two and three have entries like 2 3 denoting the 2nd level of the first factor and 3rd level of the second factor. The 4th column in `I` would contain all 1s.

**Scans** Select the images to be analysed. They must all have the same image dimensions, orientation, voxel size etc.

**Factor matrix** Specify factor/level matrix as a `nscan-by-4` matrix. Note that the first column of `I` is reserved for the internal replication factor and must not be used for experimental factors.

## Main effects & Interactions

**Main effect** Add a main effect to your design matrix.

**Factor number** Enter the number of the factor.

**Interaction** Add an interaction to your design matrix.

**Factor numbers** Enter the numbers of the factors of this (two-way) interaction.

## 10.3 Covariates

This option allows for the specification of covariates and nuisance variables (note that SPM does not make any distinction between effects of interest (including covariates) and nuisance effects).

### 10.3.1 Covariate

Add a new covariate to your experimental design.

#### Vector

Vector of covariate values.

Enter the covariate values "per subject" (i.e. all for subject 1, then all for subject 2, etc). Importantly, the ordering of the cells of a factorial design has to be the same for all subjects in order to be consistent with the ordering of the covariate values.

#### Name

Name of covariate.

#### Interactions

For each covariate you have defined, there is an opportunity to create an additional regressor that is the interaction between the covariate and a chosen experimental factor.

#### Centering

Centering, in the simplest case, refers to subtracting the mean (central) value from the covariate values, which is equivalent to orthogonalising the covariate with respect to the constant column.

Subtracting a constant from a covariate changes the beta for the constant term, but not that for the covariate. In the simplest case, centering a covariate in a simple regression leaves the slope unchanged, but converts the intercept from being the modelled value when the covariate was zero, to being the modelled value at the mean of the covariate, which is often more easily interpretable. For example, the modelled value at the subjects' mean age is usually more meaningful than the (extrapolated) value at an age of zero.

If a covariate value of zero is interpretable and/or you wish to preserve the values of the covariate then choose 'No centering'. You should also choose not to center if you have already subtracted some suitable value from your covariate, such as a commonly used reference level or the mean from another (e.g. larger) sample. Note that 'User specified value' has no effect, but is present for compatibility with earlier SPM versions.

Other centering options should only be used in special cases. More complicated centering options can orthogonalise a covariate or a covariate-factor interaction with respect to a factor, in which case covariate values within a particular level of a factor have their mean over that level subtracted. As in the simple case, such orthogonalisation changes the betas for the factor used to orthogonalise, not those for the covariate/interaction being orthogonalised. This therefore allows an added covariate/interaction to explain some otherwise unexplained variance, but without altering the group difference from that without the covariate/interaction. This is usually \*inappropriate\* except in special cases. One such case is with two groups and covariate that only has meaningful values for one group (such as a disease severity score that has no meaning for a control group); centering the covariate by the group factor centers the values for the meaningful group and (appropriately) zeroes the values for the other group.

## 10.4 Multiple covariates

This option allows for the specification of multiple covariates from TXT/MAT files.

### 10.4.1 Covariates

Add a new set of covariates to your experimental design.

#### File(s)

Select the \*.mat/\*.txt file(s) containing details of your multiple covariates.

You will first need to create a \*.mat file containing a matrix R or a \*.txt file containing the covariates. Each column of R will contain a different covariate. Unless the covariates names are given in a cell array called 'names' in the MAT-file containing variable R, the covariates will be named R1, R2, R3, ..etc.

#### Interactions

For each covariate you have defined, there is an opportunity to create an additional regressor that is the interaction between the covariate and a chosen experimental factor.

#### Centering

Centering, in the simplest case, refers to subtracting the mean (central) value from the covariate values, which is equivalent to orthogonalising the covariate with respect to the constant column.

Subtracting a constant from a covariate changes the beta for the constant term, but not that for the covariate. In the simplest case, centering a covariate in a simple regression leaves the slope unchanged, but converts the intercept from being the modelled value when the covariate was zero, to being the modelled value at the mean of the covariate, which is often more easily interpretable. For example, the modelled value at the subjects' mean age is usually more meaningful than the (extrapolated) value at an age of zero.

If a covariate value of zero is interpretable and/or you wish to preserve the values of the covariate then choose 'No centering'. You should also choose not to center if you have already subtracted some suitable value from your covariate, such as a commonly used reference level or the mean from another (e.g. larger) sample. Note that 'User specified value' has no effect, but is present for compatibility with earlier SPM versions.

Other centering options should only be used in special cases. More complicated centering options can orthogonalise a covariate or a covariate-factor interaction with respect to a factor, in which case covariate values within a particular level of a factor have their mean over that level subtracted. As in the simple case, such orthogonalisation changes the betas for the factor used to orthogonalise, not those for the covariate/interaction being orthogonalised. This therefore allows an added covariate/interaction to explain some otherwise unexplained variance, but without altering the group difference from that without the covariate/interaction. This is usually \*inappropriate\* except in special cases. One such case is with two groups and covariate that only has meaningful values for one group (such as a disease severity score that has no meaning for a control group); centering the covariate by the group factor centers the values for the meaningful group and (appropriately) zeroes the values for the other group.

## 10.5 Masking

The mask specifies the voxels within the image volume which are to be assessed. SPM supports three methods of masking (1) Threshold, (2) Implicit and (3) Explicit. The volume analysed is the intersection of all masks.

### 10.5.1 Threshold masking

Images are thresholded at a given value and only voxels at which all images exceed the threshold are included.

#### None

No threshold masking

**Absolute**

Images are thresholded at a given value and only voxels at which all images exceed the threshold are included.

This option allows you to specify the absolute value of the threshold.

**Threshold** Enter the absolute value of the threshold.

**Relative**

Images are thresholded at a given value and only voxels at which all images exceed the threshold are included.

This option allows you to specify the value of the threshold as a proportion of the global value.

**Threshold** Enter the threshold as a proportion of the global value.

**10.5.2 Implicit Mask**

An "implicit mask" is a mask implied by a particular voxel value. Voxels with this mask value are excluded from the analysis.

For image data-types with a representation of NaN (see `spm_type.m`), NaN's is the implicit mask value, (and NaN's are always masked out).

For image data-types without a representation of NaN, zero is the mask value, and the user can choose whether zero voxels should be masked out or not.

By default, an implicit mask is used.

**10.5.3 Explicit Mask**

Explicit masks are other images containing (implicit) masks that are to be applied to the current analysis.

All voxels with value NaN (for image data-types with a representation of NaN), or zero (for other data types) are excluded from the analysis.

Explicit mask images can have any orientation and voxel/image size. Nearest neighbour interpolation of a mask image is used if the voxel centers of the input images do not coincide with that of the mask image.

**10.6 Global calculation**

This option is for PET or VBM data (not second level fMRI).

There are three methods for estimating global effects (1) Omit (assuming no other options requiring the global value chosen) (2) User defined (enter your own vector of global values) (3) Mean: SPM standard mean voxel value (within per image `fullmean/8` mask)

**10.6.1 Omit**

Omit

**10.6.2 User**

User defined global effects (enter your own vector of global values).

**Global values**

Enter the vector of global values.

### 10.6.3 Mean

SPM standard mean voxel value.

This defines the global mean via a two-step process. Firstly, the overall mean is computed. Voxels with values less than 1/8 of this value are then deemed extra-cranial and get masked out. The mean is then recomputed on the remaining voxels.

## 10.7 Global normalisation

These options are for PET or VBM data (not second level fMRI).

'Overall grand mean scaling' simply scales all the data by a common factor such that the mean of all the global values is the value specified.

'Normalisation' refers to either proportionally scaling each image or adding a covariate to adjust for the global values.

### 10.7.1 Overall grand mean scaling

Scaling of the overall grand mean simply scales all the data by a common factor such that the mean of all the global values is the value specified. For qualitative data, this puts the data into an intuitively accessible scale without altering the statistics.

When proportional scaling global normalisation is used each image is separately scaled such that its global value is that specified (in which case the grand mean is also implicitly scaled to that value). So, to proportionally scale each image so that its global value is eg. 20, select <Yes> then type in 20 for the grand mean scaled value.

When using AnCova or no global normalisation, with data from different subjects or sessions, an intermediate situation may be appropriate, and you may be given the option to scale group, session or subject grand means separately.

#### No

No overall grand mean scaling.

#### Yes

Scaling of the overall grand mean simply scales all the data by a common factor such that the mean of all the global values is the value specified. For qualitative data, this puts the data into an intuitively accessible scale without altering the statistics.

**Grand mean scaled value** The default value of 50, scales the global flow to a physiologically realistic value of 50ml/dl/min.

### 10.7.2 Normalisation

This option is for PET or VBM data (not second level fMRI).

Global nuisance effects (such as average values for PET images, or total tissue volumes for VBM) can be accounted for either by dividing the intensities in each image by the image's global value (proportional scaling), or by including the global covariate as a nuisance effect in the general linear model (AnCova).

Much has been written on which to use, and when. Basically, since proportional scaling also scales the variance term, it is appropriate for situations where the global measurement predominantly reflects gain or sensitivity. Where variance is constant across the range of global values, linear modelling in an AnCova approach has more flexibility, since the model is not restricted to a simple proportional regression.

'Ancova by subject' or 'Ancova by effect' options are implemented using the ANCOVA options provided where each experimental factor (eg. subject or effect), is defined. These allow eg. different subjects to have different relationships between local and global measurements.

Since differences between subjects may be due to gain and sensitivity effects, AnCova by subject could be combined with "grand mean scaling by subject" (an option also provided where each experimental factor is originally defined) to obtain a combination of between subject proportional scaling and within subject AnCova.



**Part IV**  
**EEG/MEG**



# Chapter 11

## SPM for MEG/EEG overview

### 11.1 Welcome to SPM for M/EEG

SPM functionality for M/EEG data analysis consists of three major parts.

- Statistical analysis of voxel-based images. For statistical analysis, we use exactly the same routines as SPM for fMRI users would. These are robust and validated functions based on the General Linear Model<sup>1</sup> (GLM) and Random Field Theory<sup>2</sup> (RFT). The statistical methods are equally applicable to multi- (or single-) subject M/EEG studies.
- Source Reconstruction<sup>3</sup>. Our group has invested heavily in establishing Bayesian approaches to the source reconstruction of M/EEG data. Good source reconstruction techniques are vital for the M/EEG field, otherwise it would be very difficult to relate sensor data to neuroanatomy or findings from other modalities like fMRI. Bayesian source reconstruction provides a principled way of incorporating prior beliefs about how the data were generated, and enables principled methods for model comparison. With the use of priors and Bayesian model comparison, M/EEG source reconstruction is a very powerful neuroimaging tool, which has a unique macroscopic view on neuronal dynamics.
- Dynamic Causal Modelling<sup>4</sup> (DCM), which is a spatio-temporal network model to estimate effective connectivity in a network of sources. For M/EEG, DCM is a powerful technique, because the data are highly resolved in time and this makes the identifiability of neurobiologically inspired network models feasible. This means that DCM can make inferences about temporal precedence of sources and can quantify changes in feedforward, backward and lateral connectivity among sources on a neuronal time-scale of milliseconds.

In order to make it possible for the users to prepare their data for SPM analyses we also implemented a range of tools for the full analysis pipeline starting with raw data from the MEG or EEG machine.

Our overall goal is to provide an academic M/EEG analysis software package that can be used by everyone to apply the most recent methods available for the analysis of M/EEG data. Although SPM development is focusing on a set of specific methods pioneered by our group, we aim at making it straightforward for the users to combine data processing in SPM and other software packages. We have a formal collaboration with the excellent FieldTrip package (head developer: Robert Oostenveld, F.C. Donders centre in Nijmegen/Netherlands)<sup>5</sup> on many analysis issues. For example, SPM and FieldTrip share routines for converting data to MATLAB, forward modelling for M/EEG source reconstruction and the SPM distribution contains a version of FieldTrip so that one can combine FieldTrip and SPM functions in custom scripts. SPM and FieldTrip complement each other well, as SPM is geared toward specific analysis tools, whereas FieldTrip is a more

---

<sup>1</sup>GLM: <http://www.fil.ion.ucl.ac.uk/spm/doc/biblio/Keyword/GLM.html>

<sup>2</sup>RFT: <http://www.fil.ion.ucl.ac.uk/spm/doc/biblio/Keyword/RFT.html>

<sup>3</sup>Source Reconstruction: <http://www.fil.ion.ucl.ac.uk/spm/doc/biblio/Keyword/EEG.html>

<sup>4</sup>Dynamic Causal Modelling: <http://www.fil.ion.ucl.ac.uk/spm/doc/biblio/Keyword/DCM.html>

<sup>5</sup>FieldTrip: <http://fieldtrip.fcdonders.nl/>

general repository of different methods that can be put together in flexible ways to perform a variety of analyses. This flexibility of FieldTrip, however, comes at the expense of accessibility to a non-expert user. FieldTrip does not have a graphical user interface (GUI) and its functions are used by writing custom MATLAB scripts. By combining SPM and FieldTrip the flexibility of FieldTrip can be complemented by SPM's GUI tools and batching system. Within this framework, power users can easily and rapidly develop specialized analysis tools with GUIs that can then also be used by non-proficient MATLAB users. Some examples of such tools are available in the MEEG toolbox distributed with SPM. We will also be happy to include in this toolbox new tools contributed by other users as long as they are of general interest and applicability.

## 11.2 Changes from SPM8 to SPM12

SPM8 introduced major changes to the initial implementation of M/EEG analyses in SPM5. The main change was a different data format that used an object to ensure internal consistency and integrity of the data structures and provide a consistent interface to the functions using M/EEG data. The use of the object substantially improved the stability and robustness of SPM code. The changes in data format and object details from SPM8 to SPM12 are relatively minor. The aims of those changes were to rationalise the internal data structures and object methods to remove some 'historical' design mistakes and inconsistencies. For instance, the methods `meegchannels`, `eogchannels`, `ecgchannels` from SPM8 have been replaced with method `indchantype` that accepts as an argument the desired channel type and returns channel indices. `indchantype` is one of several methods with similar functionality, the others being `indsample`, `indchannel`, `indtrial` (that replaces `pickconditions`) and `indfrequency`.

Another major change in data preprocessing functionality was removal of interactive GUI elements and switch to the use of SPM batch system. This should make it easy to build processing pipelines for performing complete complicated data analyses without programming. The use of batch has many advantages but can also complicate some of the operations because a batch must be configured in advance and cannot rely on information available in the input file. For instance, the batch tool cannot know the channel names for a particular dataset and thus cannot generate a dialog box for the user to choose the channels. To facilitate the processing steps requiring this kind of information additional functionalities have been added to the 'Prepare' tool under 'Batch inputs' menu. One can now make the necessary choices for a particular dataset using an uninteractive GUI and then save the results in a mat file and use this file as an input to batch.

The following chapters go through all the EEG/MEG related functionality of SPM. Most users will probably find the tutorial (chapter 40) useful for a quick start. A further detailed description of the conversion, preprocessing functions, and the display is given in chapter 12. In chapter 13, we explain how one would use SPM's statistical machinery to analyse M/EEG data. The 3D-source reconstruction routines, including dipole modelling, are described in chapter 14. Finally, in chapter 16, we describe the graphical user interface for dynamical causal modelling, for evoked responses, induced responses, and local field potentials.

## Chapter 12

# EEG/MEG preprocessing – Reference

In this chapter we will describe the function and syntax of all SPM/MEEG preprocessing and display functions. This will be the most detailed description of the functions in this manual. Our goal is to provide a comprehensive description of how the software can be used to preprocess M/EEG data up to the point where one would use one of the source reconstruction techniques or statistical analysis of M/EEG channel data.

These functions can be called either from the MATLAB command line and scripts, or via the batch input system. The batch input system is designed for repetitive analyses of data (eg. from multiple subjects) . Once the user becomes familiar with the batch tools necessary for their analysis it is very easy to chain them using batch dependencies and run them as one pipeline. The principles of using the batch tool are described in 47. The command line facilities are very useful for writing scripts, or using SPM’s history-to-script functionality to generate scripts automatically.

For scripts we follow the concept of providing only one input argument to each function. This input argument is usually a structure (struct) that contains all input arguments as fields. This approach has the advantage that the input does not need to follow a specific input argument order. For some arguments default values can be provided. When an obligatory argument is missing, this will cause an error.

Below we will describe the parameters available in the batch tool and the names of the corresponding low-level SPM functions. The interface for calling these functions from a script is described in function headers.

We will go through the conversion of the data, specifics of the M/EEG format in SPM, how to properly enter additional information about the channels, how to call FieldTrip-functions from SPM, a complete reference of all methods and functions, how to use the display, and finally how to script and batch the preprocessing.

### 12.1 Conversion of data

The first step of any analysis is the conversion of data from its native machine-dependent format to a MATLABbased, common SPM format. This format stores the data in a \*.dat file and all other information in a \*.mat file. The \*.mat file contains the data structure D and the \*.dat is the M/EEG data. The conversion facility of SPM is based on the “fileio” toolbox<sup>1</sup>, which is shared between SPM, FieldTrip and EEGLAB toolboxes and jointly developed by the users of these toolboxes. At the moment most common EEG and MEG data formats are supported. For some cases, it might be necessary to install additional MATLAB toolboxes. In this case an error message will be displayed with a link where the appropriate toolbox can be downloaded. If your data format is not recognized by “fileio”, you can extend the “fileio” toolbox and contribute your

---

<sup>1</sup>fileio: <http://fieldtrip.fcdonders.nl/development/fileio>

code to us. See “fileio” page for details.

After selecting on the CONVERT from the CONVERT dropdown menu of the M/EEG GUI you will be asked (“Define settings?”) to choose whether to define some settings for the conversion or “just read”. The latter option was introduced to enable a simple and convenient conversion of the data with no questions asked. The resulting SPM M/EEG data file can then be explored with SPM’s reviewing tool to determine the appropriate conversion parameters for the future. If the “just read” option is chosen, SPM will try to convert the whole dataset preserving as much data as possible. The other option - “yes” - opens the batch tool for conversion

In either case you will need to select the file to be converted. As a rule of thumb, if the dataset consists of several files, the file containing the data (which is usually the largest) should be selected. SPM can usually automatically recognize the data format and apply the appropriate conversion routine. However, in some cases there is not enough information in the data file for SPM to recognize the format. This will typically be the case for files with non-specific extensions (\*.dat, \*.bin, \*.eeg, etc). In these cases the header-, and not the data-, file should be chosen for conversion and if it is recognized, SPM will locate the data file automatically. In some rare cases automatic recognition is not possible or there are several possible low-level readers available for the same format. For these cases there is an option to force SPM to use a particular low-level reader available with the batch tool or in a script (see below).

The other options in the conversion batch are as follows:

- Reading mode - a file can be read either as continuous or epoched. In the continuous case either the whole file or a contiguous time window can be read. In the epoched case trials should be defined (see ‘Epoching’ below). The advantage of defining trials at conversion is that only the necessary subset of the raw data is converted. This is useful when the trials of interest are only a small subset of the whole recording (e.g. some events recorded during sleep). Note that some datasets do not contain continuous data to begin with. These datasets should usually be converted with the “Epoched” option. There is also a possibility to only convert the header without the data. This can be useful if the information of interest is in the header (e.g. sensor locations).
- Channel selection - a subset of channels can be selected. There are several options for defining this subset that can be combined: by channel type, by names or using a .mat file containing a list of channel labels. Note that channel selection branch is available in many batch tools and its functionality is the same everywhere.
- Output filename - the name for the output dataset. Note that here any name can be given whereas in other preprocessing tools the user can only define a prefix to be appended to the existing name (this limitation can be circumvented using the ‘Copy’ tool). By default SPM will append ‘spmeeeg\_’ prefix to the raw data file name.
- Event padding - usually when epoching at conversion only events occurring within trials are included with the trials. This option makes it possible to also include events occurring earlier and later within the specified time window.
- Block size - this is size of blocks used internally to read large files. Does not usually need to be modified unless you have an old system with memory limitations.
- Check trial boundaries - SPM will not usually read data as continuous if it is clear from the raw data file that it is not the case and will give an error. In some rare cases this might need to be circumvented (e.g. if truly continuous data are stored in chunks (pseudo-epoched) and SPM does not recognise it automatically).
- Save original header - the generic fileio interface does not let through all the possible header fields available for specific formats. Sometimes those missing header fields are necessary for particular functionality and this option allows to keep the complete original header as a subfield of the converted header. A particular case where this is useful is processing of continuous head localisation data in CTF MEG system which requires some information from the original header to interpret it.

- Input data format - this option allows to force a particular low-level reader to convert the data. It is not usually necessary. Power users can find possible values for this field in the code of `ft_read_header` function.

## 12.2 Converting arbitrary data

It might be the case that your data is not in any standard format but is only available as an ASCII or Excel file or as a variable in the MATLAB workspace. Then you have two options depending on whether you would be willing to use a MATLAB script or want to only use the GUI.

'Prepare' interface in SPM has an option to convert a variable in the MATLAB workspace to SPM format. Only a few question will be asked to determine the dimensions of the data and the time axis. The other information (e.g. channel labels) can be provided via the SPM reviewing tool.

If you are willing to write a simple MATLAB script, the most straightforward way to convert your data would be to create a quite simple FieldTrip raw data structure (MATLAB `struct`) and then use SPM's `spm_eeg_ft2spm.m` function to convert this structure to SPM dataset. Missing information can then be supplemented using `meeeg` methods and SPM functions.

FieldTrip raw struct must contain the following fields:

- `.trial` - cell array of trials containing matrices with identical dimensions (channels  $\times$  time).
- `.time` - cell array of time vectors (in sec) - one cell per trial, containing a time vector the same length as the second dimension of the data. For SPM, the time vectors must be identical.
- `.label` - cell array of strings, list of channel labels. Same length as the first dimension of the data.

If your data only has one trial (e.g. it is already an average or it is raw continuous data) you should only have one cell in `.trial` and `.time` fields of the raw struct.

An example script for converting LFP data can be found under `man\example1_scripts\spm_eeg_convert_a`

As some of third party toolboxes whose format SPM can convert also support converting arbitrary data via GUI (e.g. EEGLAB), it is also possible to use one these toolboxes first to build a dataset and then convert it to SPM.

## 12.3 The M/EEG SPM format

SPM8 introduced major changes to the initial implementation of M/EEG analyses in SPM5. The main change was a different data format that used an object to ensure internal consistency and integrity of the data structures and provide a consistent interface to the functions using M/EEG data. The use of the object substantially improved the stability and robustness of SPM code. The changes in data format and object details from SPM8 to SPM12 were relatively minor. The aims of those changes were to rationalise the internal data structures and object methods to remove some 'historical' design mistakes and inconsistencies.

SPM M/EEG format consists of two files: header file with extension `.mat` and data file with extension `.dat`. The header is saved in the mat file as a struct called 'D'. Description of the struct fields can be found in the header of `meeeg.m`. When a dataset is loaded into memory by SPM using the `spm_eeg_load` function (see below) the header is converted to `@meeeg` object and the data are linked to the object using memory mapping so they are not actually kept in memory unnecessarily. The object can only be manipulated using standardized functions (called methods), which makes it very hard to introduce any inconsistency into SPM M/EEG data. Also, using methods simplifies internal book-keeping, which makes it much easier to program functions operating on the M/EEG object. SPM functions only access the header data via the object interface and we strongly encourage the power users to become familiar with this interface and also use it in their own code. Using the object can make your code simpler as many operations requiring multiple commands when working with the struct directly are already implemented in `@meeeg` methods. When converting from struct to an object an automatic integrity check is done.

Many problems can be fixed on the fly and there will only be an error if SPM does not know how to fix a problem. Messages from the automatic consistency checks will sometimes appear during conversion or other processing steps. They do not usually indicate a problem, unless an error is generated.

## 12.4 Preparing the data after conversion and specifying batch inputs

SPM does its best to extract information automatically from the various data formats. In some cases it can also supplement the converted dataset with information not directly present in the raw data. For instance, SPM can recognize common EEG channel setups (extended 1020, Biosemi, EGI) based on channel labels and assigns 'EEG' channel type and default electrode locations for these cases. However, there are data types which are either not yet supported in this way or do not contain sufficient information for SPM to make the automatic choices. Also the channel labels do not always correctly describe the actual electrode locations in an experiment. In these cases, further information needs to be supplied by the user. Reading and linking this additional information with the data was the original purpose of the **Prepare** interface. In SPM12 with removal of interactive GUI elements from all preprocessing functions some of those elements were added to 'Prepare' so that the users will be able to prepare inputs for batch tool using interactive GUI. These tools can be found in the 'Batch inputs menu'.

'Prepare' interface is accessed by selecting **Prepare** from the **Convert** drop-down menu in the GUI. A menu (easily overlooked) will appear at the top of SPM's interactive window. The same functionality can also be accessed by pressing "Prepare SPM file" in the SPM M/EEG reviewing tool. Note that in the latest Mac OS versions the menu can appear at the top of the screen when clicking on the interactive window rather than in the window itself.

In this menu, an SPM M/EEG file can be loaded and saved using the "File" submenu. 'Load header' option makes it possible to only load the header information from a raw data file without converting any data. This is useful to subsequently use this header information (e.g. channel labels) for specifying batch inputs. 'Import from workspace' is a basic GUI functionality for converting any data to SPM M/EEG format. It will scan the workspace for any numeric arrays and list them for the user to choose the right one. It will then ask to choose the number of channels and trials to correctly identify the dimensions of the data and also to specify the time axis by providing the sampling rate and time of the first sample (in ms). Finally it will ask the user to name the dataset. Then the dataset will be created and opened in SPM reviewing tool (see below) where the rest of the information (e.g. channel labels) can be supplemented.

The 'Batch inputs' submenu contains tools to interactively specify and save some pieces of information that can be then used as inputs to different batch tools.

'Channel selection' as the name suggests is for making channel lists. A list of all channels in the dataset is shown and the user can select a subset of them and save in a mat-file. Channel set selection is necessary in many batch tools and choosing a pre-saved list is a convenient way of doing it.

'Trial definition' tool makes it possible to interactively define trials based on the events in the dataset. You will first need to specify the time window (in ms) to be cut around the triggers and the number of different conditions you want to have. A list will then pop up, and present the found triggers with their type and value entries. These can sometimes look strange, but if you want to run a batch or script to do the epoching, you have to first find out what the type and value of your event of interest are. Fortunately, these tend to be the same over scanning sessions, so that you can batch multi-subject epoching using the types and values found in one subject. You also have to come up with a "condition label" for each trial type, which can be anything you choose. This is the label that SPM will use to indicate the trial type of a trial at later processing stages. It is possible to use several types of triggers for defining trials with the same label - in the GUI, just select several events using **Shift** or **Ctrl** key. Finally, you can specify a shift for each condition so that the zero time of the trial will be shifted with respect to the trigger (e.g. to account for projector delay). When all conditions are specified, you can choose to review a list of epochs and can edit the list by unselecting some of them. Note that for such manual tweaking to work you must define trials on the same dataset that you will later epoch. You can then save



the completed trial definition.

'Montage' menu allows one to specify custom montages. The most generic way to do it is via 'Custom montage' option which brings up montage editing GUI. On the left hand side, you will find the montage-matrix, where each row stands for a new channel. This means the labels in the left column describe the new labels. The old labels are on top, that means, each row contains weights for how the old channels must be weighted to produce new channels in the montage. On the right hand side, you see a graphical representation of the current matrix. The default is the identity matrix, i.e., the montage will not change anything. The concept is very general. For example, if you want to remove channels from the data, just delete the corresponding row from the montage matrix. To re-reference to a particular channel the column for this channel should be -1 for all rows, except the row corresponding to itself which should be 0, whereas the other channels should have 1 in the intersection of their column and row (the diagonal of the matrix) and 0 elsewhere. For average reference the matrix should have  $(N - 1)/N$  (where  $N$  is number of channels) at the diagonal and  $-1/N$  elsewhere. In principle, any montage can be represented this way. The specification will only need to be done once for your setup and then you can save the montage and use it routinely. After changing the weights of the matrix, you can visualise the montage by pressing the button in the lower right below the figure.

Simpler way to specify some common montages is via 'Re-reference' and 'ROI' options. 'Re-reference' is for rereferencing montage specified by choosing one or more (possibly all) channels as reference from the EEG channel list. 'ROI' is for averaging across channel groups (e.g. reducing the data to 'Frontal channels', 'Occipital channels' etc.).

In many cases in SPM the order of the conditions in the file is important (for instance in 3D source reconstruction and in DCM). The 'Sort conditions' functionality makes it possible to change the specification of the order (without actually changing the data file). Subsequently every time the order of the conditions is important, the order thereby specified will be used. For instance, if you sort conditions in an epoched file and then average it, the conditions in the average file will be ordered as you specified. If you originally defined the trials by selecting events from a list then the order in which you made the selection will be preserved. You can see the present order in a file using the `condlist` method (`condlist(D)`). The specified order can be saved in a mat-file and used for batch processing (in batch version of 'Prepare', see below).

The "Channel types" submenu allows reviewing and changing the channel types. Use the "Review" option to examine the presently set channel types. During conversion, SPM will make an informed *guess* at the correct channel types but this can sometimes go wrong, especially for EEG data. To set a particular channel group to some channel type, select this type from the menu. A list of all channels will appear. Select the subset whose type you would like to set. **Ctrl** and **Shift** buttons can be used to refine the selection. Press OK to apply your choice. It is especially important to correctly specify which are the EEG channels. MEG types are assigned automatically by SPM and cannot be modified using the GUI.

The "Sensors" submenu can be used to supply information about the sensor positions to the file. This information is needed to perform 3D source reconstruction and DCM analysis for EEG and MEG data. Sensor positions for MEG are extracted from the raw data automatically and are already present. For EEG, sensor positions are usually measured by a special device (such as Polhemus) and are not part of the dataset. Even if you do not measure electrode positions routinely in your lab, we recommend to perform at least one initial measurement with the electrode cap you use and use the result as your standard template. In order for SPM to provide a meaningful interpretation of the results of source reconstruction, it should link the coordinate system in which sensor positions are originally represented to the coordinate system of a structural MRI image (MNI coordinates). In general to link between two coordinate systems you will need a set of at least 3 points whose coordinates are known in both systems. This is a kind of *Rosetta stone* that can be used to convert a position of any point from one system to the other. These points are called "fiducials" and the process of providing SPM with all the necessary information to create the *Rosetta stone* for your data is called "coregistration". The most commonly used fiducials are the nose bridge and the two pre-auricular points. The coordinates of these points for SPM's standard template image are hard-coded in SPM code. So if you provide the coordinates of these specific points with your sensor positions, it will be enough for SPM. If you do not have these fiducials but have other anatomical landmarks (for instance 3 EEG electrodes whose positions can be easily marked on a structural image) it will be possible to use

them for coregistration as well, but that will require additional input from you. In addition, or as a replacement of fiducials a headshape measurement may be used. This measurement is done by an operator moving his digitizer pen around on the subject’s scalp and generates many more data points than just 3 fiducials. EEG sensor and fiducial positions can be added to an SPM file using the “Load EEG sensors” menu. There are 3 options:

- “Assign default” - assigning default sensor positions. If this is possible, it will be done automatically at conversion but this option can be used to revert to default sensor positions after making some changes.
- “From a \*.mat file” - this option is for the kind of files that were used in SPM5 and can also be used for any kind of locations without trying to get them into one of the standard formats. SPM will ask for two files. The sensors file should contain an  $N \times 3$  matrix, where  $N$  is the same as the number of channels whose type is set to “EEG” and the order of the rows matches the order of these channels in the SPM file. The fiducials file should contain a  $K \times 3$  matrix, where  $K$  (usually 3) is the number of fiducials. You will then be asked to provide labels for these fiducials. They should appear in the same order as the rows in the file.
- “Convert locations file” - this option uses a function from the internal “fileio” toolbox that supports several common formats for EEG channel position specification such as \*.sfp and BESA’s \*.elp. It can also read Polhemus files from FIL and FCDC. In general Polhemus devices do not have a standard data format so if you are using Polhemus at a different site is is most likely that your Polhemus file will not be recognized by SPM directly. You will need to convert it to another format. An \*.sfp file is the easiest to create (for instance in Excel). It is just an ASCII file containing a column of channel labels and 3 columns of cartesian coordinates. Check “fileio” website<sup>2</sup> for a complete list of supported formats. The file you are importing can also contain positions of fiducial points or any other named points that do not necessarily correspond to channels. You can also include multiple headshape points with the label “headshape”. The important thing is that there are coordinates for each channel that was assigned “EEG” type.

The fiducials for MEG are automatically loaded from the dataset. However, in some MEG setups the situation is more complicated. For instance, it might be convenient to attach the coils marking MEG fiducials to the top of the head, where there are no clear anatomical landmarks. In this case there should be an additional file measured with a Polhemus-like device that contains the positions of MEG fiducials and something that can be linked to a structural image (either anatomical landmarks or a headshape) in the same coordinate system. The way SPM handles this situation is in two steps. First, this additional file is converted into the same coordinate system in which MEG sensors are represented and it replaces the original MEG fiducials. At a later stage having MEG sensors and fiducials/headshape in the same coordinate system, SPM uses the fiducials/headshape for coregistration with standard MRI template or subject’s own structural image. If you can mark the points where your MEG fiducial coils were located on a structural image, the step described below is not necessary. It is also possible that the digitizer measurement is stored with the raw data. Then SPM will read it automatically. Otherwise, the additional fiducials/headshape file can be loaded using the “Load MEG Fiducials/Headshape” menu. The supported formats are the same as for electrode locations. It is also possible to create a fiducials/headshape MATLAB struct and store it in a \*.mat file. This file will also be recognized by SPM. The struct should be called `shape` and it should contain the following fields: `shape.pnt` - a  $K \times 3$  matrix (possibly empty) with headshape points i.e. points that are on the surface of the head and have no labels, `shape.fid.pnt` -  $M \times 3$  matrix with fiducial points i.e. points that have labels, `shape.fid.label` -  $M \times 1$  cell array of strings with the labels of the points in `shape.fid.pnt`. As mentioned above,  $M$  should be at least 3 for the coregistration to work.

If you did not use default 3D positions, after loading the sensor positions you can perform coregistration of your sensors with SPM’s template head model. This initial alignment is helpful to verify that the sensor information you supplied were interpreted correctly and should also be

<sup>2</sup>fileio: <http://fieldtrip.fcdonders.nl/dataformat>

done if you would like to generate a 2D sensor template based on your 3D sensor positions (see below). The 2D-coordinates will be used for displaying the data in a topologically meaningful way. This is implemented using the “Coregister” option. For details of how this option works see the 3D source reconstruction chapter 14.

‘Define EEG referencing’ menu makes it possible to specify the original recording reference for EEG data. This is necessary for source reconstruction and DCM to work correctly. Most commonly the reference one of the sensors and it can be chosen from the list. It can also be a combination of two sensors (e.g. averaged ears) or average reference, specified by selecting the relevant or all sensors respectively. It could be possible to support even more complicated referencing schemes (rarely used in research setting) like double banana. That would require loading a special montage file where the referencing arrangement is specified.

The “2D Projection” menu deals with the generation of representative 2D-coordinates for the sensors. Note that generating 2D-coordinates is not obligatory. If the 2D-coordinates are not specified, the sensors will be, when displaying, presented in a default square grid. Missing out on topographically meaningful 2D-coordinates might be useful when working on few channels. The 2D-coordinates are also used for producing scalp-level SPMs in voxel space when converting M/EEG data to images for later statistical analysis (see below). If you are planning to do 3D source reconstruction or DCM, 2D-coordinates are not necessarily required. Also, you can load 2D-coordinates from a file (several example files are available in the `EEGtemplates` SPM directory). 2D-coordinates can also be generated by projecting the 3D sensor positions to a plane. This is done automatically when default 3D coordinates can be assigned, and also for MEG. In case of custom EEG sensor positions coregistration should be performed first (see above). The resulting 2D-coordinates are displayed in SPM’s graphics window. You can modify these projected 2D-coordinates manually by adding, deleting and moving sensors. To select a sensor, click on its label. The label will change its color to green. If you then click at a different location, the sensor will be moved to this position. Note that, at this stage, SPM does not check whether there is any correspondence between the labels of the coordinates and the labels of the channels stored in the SPM file. When you are satisfied with the 2D-coordinates, select “Apply” from the menu and the coordinates will be assigned to EEG or MEG channels according to their labels. Note that 2D-coordinates cannot be assigned to channels of other types than M/EEG.

Remember to save the file using “File/Save” after you finished modifying it using the `Prepare` interface. Your changes will not be saved automatically. In case of invoking `Prepare` from the reviewing tool you should press the ‘OK’ button that will appear at the bottom left of the interactive window, and then save the file with the “Save” button of the reviewing tool.

In the rare case that you neither have measured sensor locations, or fiducials, and the supplied standard templates do not work for you, you can also supply a so-called channel template file, which contains all information necessary. However, remember, that if you do not supply any 2D-coordinates, you can still use all SPM functions, however, SPM will use 2D-coordinates laid out in a topographically unmeaningful rectangular pattern.

A channel template file contains four variables:

<code>Nchannels</code>	-	The number of channels
<code>Cnames</code>	-	A cell vector of channel names. Each cell can contain either a string or a cell vector of strings. The latter allows for multiple versions of a given channel name. Case can be ignored, i.e., it doesn’t matter whether channel names are in small or capital letters.
<code>Cpos</code>	-	A $2 \times Nchannels$ -matrix of channel coordinates on a 2D plane. In $x$ - and $y$ -direction the minimum coordinate must be $\leq 0.05$ and the maximum coordinate must be $\geq 0.95$ .
<code>Rxy</code>	-	A factor that determines the ratio of the width of the display plots to their height when displaying the data. Standard is 1.5.

Note that the channel template files and 3D coordinate files with labels (such as `*.sfp`) can contain many more channel labels than your data file. SPM searches, for each channel in the data, through the labels in the channel template file. If the labels match, the coordinate is used.

### 12.4.1 Prepare (batch)

Many of the operations of the 'Prepare' tool can be batched. The relevant batch tool is accessed by selecting 'Prepare (batch)' from 'Convert' menu. One or more tasks can be selected (note that the order can matter). The configuration of each task should be clear based on the descriptions above and will not be describe in detail here.

## 12.5 Integration of SPM and Fieldtrip

The SPM distribution includes the latest version of the FieldTrip toolbox<sup>3</sup>. FieldTrip is a MATLAB toolbox for MEG and EEG analysis that is being developed at the Donders Institute for Brain, Cognition and Behaviour at the Radboud University Nijmegen together with collaborating institutes. FieldTrip functions can be used for many kinds of analysis which are not supported in SPM proper. However, FieldTrip does not have extensive graphical user interface and its functionality should be accessed by writing scripts. Full reference documentation for FieldTrip including example scripts is available at the FieldTrip website. The SPM distribution also contains some documentation, contained as help comments in FieldTrip functions. These can be found in the directory `external\fieldtrip`.

Fieldtrip data structures can be converted to SPM EEG files using the `spm_eeg_ft2spm` function. SPM M/EEG data, once loaded with the function `spm_eeg_load` can be converted to FieldTrip format using the methods `ftraw` (with syntax `D.ftraw` or `ftraw(D)`) and `fttimelock` (with syntax `D.fttimelock` or `fttimelock(D)`). For SPM time-frequency datasets `fttimelock` method converts the data to Fieldtrip time-frequency structure.

## 12.6 Loading data into workspace

If you use the GUI only, there is no need to read this section because the functions called by the GUI will read the data automatically. However, if you plan to write scripts and access the data and header information more directly, this section should contain all the necessary information to do so.

An SPM M/EEG file can be read using the `spm_eeg_load` function. Without any arguments a file requester asks for the name of the file. With a string argument *P*, `spm_eeg_load(P)` will attempt to read a file with the name *P*. The SPM-format stores the binary data in a `*.dat` file. All header information are stored in a `*.mat` file. This `*.mat` file contains a single struct named *D* which contains several fields. When using `spm_eeg_load`, the struct is transformed into an object, and the data are linked into this object. The linking is done via memory mapping using `file_array` objects. Note that the data should always be read using the routine `spm_eeg_load`. The memory mapped data can be addressed like a matrix (see below) which is convenient for accessing the data in a random access way. However, a word of caution: If you write new values to this matrix, the matrix is not only changed in the object (in memory), but also physically on the hard disk.

## 12.7 The meeg object

This section describes the `meeg` object and its methods. This information is intended for power users who would like to write their own scripts and high level functions integrated with SPM. `meeg` methods are functions that operate on an `meeg` object, loaded with `spm_eeg_load`. The code for all methods can be found in the `@meeg` SPM directory. Most methods provide some minimalist help text. In the following, we will assume that the object variable is called *D* and was loaded by using `D = spm_eeg_load;`. Methods can be called in two ways, either as standard function call with *D* as the first input (e.g. `chanlabels(D, 1)` returns the label of the first channel) or with struct-like syntax `D.chanlabels(1)`.

<sup>3</sup>FieldTrip: <http://fieldtrip.fcdonders.nl/>

### 12.7.1 Constructor `meeg`

The `meeg` method is a constructor. Called without any arguments it will produce a consistent, but empty object. It is also possible to provide data dimensions as inputs and create a dataset with default labels etc. that can be subsequently updated using other methods. Most functions in SPM create new datasets in a different, more convenient way using `clone` method (see below). In SPM, the constructor is called when a struct has been loaded into memory by `spm_eeg_load`, and is transformed into an `meeg` object. Importantly, the constructor also checks the consistency of the object.

### 12.7.2 Array-like interface

The actual M/EEG data are memory mapped and can be accessed directly using something like `d = D(:, :, 1)`. This command would put the first trial over all channels and time points into the variable `d`. The first dimension of `D` is channels, the second peri-stimulus time, and the third is trials. If the data are time-frequency transformed, there would be four dimensions, where the frequency dimension is squeezed in at the second position (i.e., channels/frequencies/time/trials). If you wanted to change the values of the data, you would write something like `D(1,2,3) = 1;`, which would change the value of the first channel, second time-point, and third trial to 1.

### 12.7.3 `display`

This method will return, in the MATLAB window, some information about the object, e.g., `display(D)`. The same will happen when just writing `D` in the command line and pressing Enter.

### 12.7.4 Number methods

These are methods which return the number of something; they count the number of channels, etc. For example, to find out how many channels an MEEG object contains, you would use `D.nchannels`, where `D` is the object. Number functions are `nchannels`, `nfrequencies`, `nsamples`, `ntrials`. You can also use `size(D)` to get all the dimensions of the data array at once.

### 12.7.5 Reading and manipulation of information

There are a large number of methods that can be used to either read or write some information. The method name is the same but it depends on the arguments whether something is read or stored. For example, when you use the method `badchannels`, you can either type `D.badchannels`, which returns the indices of all bad channels. You could also change information about specific bad channels, e.g., `D.badchannels([43:55], 1)` will flag channels 43 to 55 as bad. You could also use `D.badchannels([43:55], ones(1,13))`, i.e. you can either use a scalar to change all channels listed, or supply a 0/1-flag for each channel. There are other functions which use the same logic. In the following we will list these functions and describe briefly what they do but won't go into much detail. We believe that you can work it out using the `badchannels`-example.

#### `selectdata`

With this method the data can be indexed using channel labels, times and condition labels instead of indices which you would usually need to find out in your code. For instance `D.selectdata('Cz', [0.1 0.12], 'Oddball')` will return the waveforms of channel `Cz` between 100 and 120 ms in peristimulus time for the condition "Oddball".

#### `badchannels`

Flags/unflags channels as bad.

#### `badtrials`

Flags/unflags trials as bad.

**chanlabels**

This method reads or writes the label of the channels (string). Note that the channel labels must be unique.

**chantype**

This method reads or writes the type of a channel (string). Currently, the types recognized by SPM are: “EEG”, “MEG”, “EMG”, “EOG”, or “Other”, but in principle type can be any string.

**clone**

This is a useful method for creating new datasets by copying the header information from an existing dataset and creating a new blank data file. Optionally the data dimensions can be changed when cloning. This method is used by SPM preprocessing functions to create new datasets where the processed data is written out.

**conditions**

This method reads or writes the name of the condition of an epoch (string).

**events**

This method returns the events stored with each trial. Events are records of things that happened during the experiment - stimuli, responses, etc. Before a file is epoched all the events are stored with the only trial and they can be used by the epoching function. For an epoched file SPM stores with each trial the events that occurred within that trial or possibly in some time window around it (this is a parameter of the epoching function that can be specified). You can use this information for your analysis (for instance to sort trials by reaction time). Events are represented by a structure array with the following fields:

- **.type** - string (e.g. “front panel trigger”)
- **.value** - number or string, can be empty (e.g. “Trig 1”).
- **.time** - in seconds in terms of the original file
- **.duration** - in seconds (optional)

Note that in order to find out the time of an event in peristimulus time you will need additional information provided by “trialonset” method.

**fname**

This method reads or sets the name of the **mat**-file, in which the header information are stored.

**fnamedat**

This method returns the name of the **dat**-file, in which the data are stored. Most commonly the **dat**-file will have the same name as the **mat**-file and will be stored in the same folder. However for some less common uses there is a possibility to link an meeg header to a binary datafile located elsewhere. (See also **link** method).

**frequencies**

If the data has been transformed to time-frequency, this method reads or writes the frequencies (Hz) of the data.

**fsample**

This method reads or writes the sampling rate of the data. In SPM, all data must have the same sampling frequency.

**fullfile**

Returns the full path to the dataset mat file. This is a shortcut for commonly used `fullfile(D.path, D.fname)`.

**history**

This method can read or add to the history of a file. Usually, each time a SPM function (e.g. like converting) does something to a data set, the function name and arguments (possibly after collecting them with the GUI) are stored in the history. Effectively, the history is a documentation of how exactly the data were processed. Of course, the history function can also be used to replicate the processing, or generate (modifiable) scripts for processing other data in the same way.

**montage**

This method makes it possible to define online montages to apply linear transforms to the data without writing it out as a new dataset. See the method code for more documentation.

**path**

This method reads or writes the path, under which the `mat`- and `dat`-files are stored.

**repl**

This method returns the number of replications measured for a condition. This method is usually only used on single trial data.

**timeonset**

This method reads and writes the time of the first sample in a trial in peristimulus time (in seconds). In SPM all trials should have the same time axis. Therefore there is only one `timeonset` in a file. For instance, if you have a pre-stimulus baseline of 100 ms and the stimulus comes at time zero, `timeonset` will be -0.1. In general it is possible to define the time axis any way you like and there is no requirement that the stimulus comes at 0 or that there is baseline with negative times (which was the case in SPM5).

**trialonset**

This method should not be confused with the more commonly used `timeonset` (see above). It returns the times of the first sample of each trial in the original raw data file time. This information is not always available to begin with. It may also be invalidated during processing (for instance if you merge two epoched files). When this happens the information is discarded. For SPM analysis `trialonset` is not usually necessary. However it may be useful if you want to relate something in your analysis to the timing of your experiment, for instance create a regressor for GLM analysis of single trials to account for fatigue. `trialonset` is also necessary for interpretation of events in epoched files.

**transformtype**

This method reads and writes the type of the data transform (string). For example, when the data are transformed to a time-frequency representation, `transformtype` is set to "TF". For time data, this is "time".

**type**

This method reads and writes the type of the data (string). Currently, this string can be "continuous", "single", "evoked", or "grandmean".

**units**

This method reads and writes the units of the measurements (string). The units are channel-specific, i.e., each channel can have its own units.

**12.7.6 Reading of information**

Some methods can only read information but not change them. These are:

**condlist**

This method returns a list of unique condition labels in the file. The order of this list is important as SPM functions rely on it many cases. For instance, when averaging an epoched dataset the conditions in the averaged file will be in the order of condlist. The order of condlist does not have to be the same as physical order of trial on disk and can be changed (See 'Sort conditions' below).

**coor2D**

This method returns the 2D-coordinates used for displaying or writing sensor data to voxel-based images. These coordinates can also be useful e.g. to find all the frontal channels (*y*-coordinate above 0.5) or all the left channels (*x*-coordinate below 0.5) etc.

**ind-methods**

This is a group of methods that return indices into the data array based on labels of the relevant data dimension. These include:

- `indchannel` - returns indices given channel labels. Several labels can be provided together as a cell array.
- `indchantype` - returns indices given channel types. Several types can be provided together as a cell array. An additional flag 'GOOD' or 'BAD' can be provided to return only good or bad channels.
- `indfrequency` - returns indices given frequencies (for TF datasets).
- `indsample` - returns indices given times (in sec).
- `indtrial` - returns trial indices given condition labels. Several labels can be provided together as a cell array. An additional flag 'GOOD' or 'BAD' can be provided to return only good or bad trials.

**modality**

This method returns the modality of the dataset (MEG, EEG, etc.). There can be datasets with multiple modalities and in this case the method returns 'Multimodal' with a list of modalities as the second output.

**time**

This method returns the time axis of the dataset (in sec). When given sample indices as input it will return the corresponding times.



**sensors**

This method returns the sensor locations structure. There is an additional argument for modality ('EEG' or 'MEG') as SPM supports datasets with more than one sensor type. The exact way sensors are represented depends on the modality and you can find more information in Fieldtrip documentation as the sensors structure is produced and used by code originally developed at by the Fieldtrip team. Note that in SPM, sensors are not directly linked with channels, unlike for instance in EEGLAB. So there is no requirement for the number of sensors and channels to match or even for any relation between them. Of course loading sensors completely unrelated to your data will not be very useful and will eventually lead to an error. This kind of representation is more powerful than a simple correspondence.

**fiducials**

This method returns the fiducials. They are represented as **shape** struct (see the discussion of loading fiducials by the **Prepare** function) with an additional field for units that is assigned automatically.

**ftraw**

This method converts an object to a FieldTrip structure. Additional arguments can be provided to only convert a subset of the data.

**fttimelock**

Similar to **ftraw** but converts the data to a different kind of Fieldtrip structure.

### 12.7.7 Manipulations of the data on disk

**delete**

This function deletes the **mat**- and **dat**-files from the disk. This is useful, for instance, in a script to delete the intermediate datasets after the next processing step has been completed.

**link**

Links a header in the workspace to a binary data file on disk. This is usually done automatically when loading the dataset. The dimensions of the datafile should match the header.

**unlink**

Unlinks the header from the data. This can be useful e.g. for working with the header when the datafile is not available.

**blank**

Creates a new empty datafile matching header dimensions.

**move**

Renames or moves a dataset

**copy**

Makes a copy of the dataset.

**save**

This method saves the object to the **mat**- and **dat**-files.

### 12.7.8 Struct-like interface

In addition to pre-defined internal fields that should only be manipulated using methods, the `meeg` object also allows storing additional information in it as long as the names of additional fields do not clash with the names of existing methods. This functionality is used by some SPM functions. For instance, the results of 3D source reconstructions are stored in `D.inv` field for which no methods are necessary to access and modify it. You can use this functionality in your scripts (try commands like `D.myfield = 'hello world'; disp(D.myfield);`). The methods `rmfield` and `isfield` work for these extra-fields as they would if the `meeg` object was a struct. Several of the methods support the struct-like interface functionality: `fieldnames`, `getfield`, `rmfield`, `isfield`. The struct-like interface only allows to access those extra-fields that were added using it and not the core fields of the object.

## 12.8 SPM functions

In this section we will describe the high-level SPM functions which are used for preprocessing M/EEG data. These functions are fairly standard and should allow a simple preprocessing of the data (e.g., epoching, filtering, averaging, etc.). Here, we will just describe what each function roughly does and what the batch input arguments mean. More detailed information about the syntax for scripts can be found in the help text of the code. For example, to get detailed help on epoching, type `help spm_eeg_epochs`. The general syntax is the same for all functions. Input arguments are provided in a struct (by convention called `S`), whose fields contain the arguments. A typical call, e.g., from a script would be: `D = spm_eeg_epochs(S)`, where `S` is the input struct, and `D` contains the return argument, the epoched `meeg` object. Note that, with all SPM functions, the object is also always written to hard disk. The filenames of the `mat`- and `dat`-files are generated by prepending (by default) a single letter to the old file name. In the example of epoching this would be an `'e'`. The idea is that by calling a sequence of functions on a file, the list of first letters of the file name shows (roughly) which preprocessing steps were called to produce this file. Note that another way of calling SPM functions and specifying all input parameters is to use the batch interface.

### 12.8.1 Epoching the data: `spm_eeg_epochs`

Epoching cuts out little chunks of continuous data and saves them as “single trials”. In M/EEG research, this is a standard data selection procedure to remove long gaps between trials and extract time segments with the same time relation to the event of interest. The first input to epoching is a continuous M/EEG dataset. It can be either data in the time domain or time-frequency data.

The epoching function can deal with three different ways of specifying trials (chosen under ‘How to define trials’). The first way (‘Define trial’) is to specify trials based on events stored in the dataset. One should define the time window in peristimulus time (which will be the same for all trials). In addition it is necessary to specify the events (triggers) around which the epochs will be “cut”. The user can add multiple entries for all the event types of interest. SPM identifies events by their “event type” and “event value”. These are strings or numbers which the software run by the EEG or MEG vendor uses when generating the measurement file. If you don’t know what they are for your system the interactive GUI in ‘Prepare’ will present the found triggers with their type and value entries. These tend to be the same over scanning sessions, so that you can batch multi-subject epoching using the types and values found in one subject. You also have to come up with a “condition label” for each trial type, which can be anything you choose. This is the label that SPM will use to indicate the trial type at later processing stages. It is possible to use several types of triggers for defining trials with the same label. Using the ‘Shift’ parameter it is possible to shift ‘time zero’ of the trial relative to the original event. This can be useful e.g. to account for known projector delay.

The second choice is to load a trial definition file. This file can be produced by an interactive GUI tool in ‘Prepare’ (under ‘Batch inputs’/‘Trial definition’) or by the user’s custom code. Trial definition file is a `mat`-file containing either variables named `'trialdef'` and `'timewin'` or the variables `'trl'` and `'conditionlabels'`. `'trialdef'` and `'timewin'` are analogous to the specification described above. `trl` is a  $N \times 2$  matrix, where each row contains the start and end of a trial

(in samples). Optionally, there can be a third column containing the offset of the trigger with respect to the trial. An offset of 0 (default) means that the first sample of the trial corresponds to the trigger. A positive offset indicates that the first sample is later than the trigger, a negative offset indicates that the trial begins before the trigger. In SPM the offset should be the same for all trials. The need to specify a whole column is for interoperability with FieldTrip where trials can have different time axes. In addition you have to specify `conditionlabels` (a single string or a cell array of strings), either one for each trial or one for all trials. Using `trl` and `conditionlabels` the user can have full control over the way the data are epoched. So if your conditions are defined in a more complicated way than just based on a single trigger you should write your own code that will output a file with `trl` and `conditionlabels` variables and that file can then be used as input to epoching. When a trial definition is created in 'Prepare' GUI both 'trialdef' and 'trl' versions are saved. 'trl' takes precedence if the mat file is later used to epoch the same dataset on which it was defined (recognised based on the file name). Otherwise 'trialdef' will be used and 'trl' - ignored. This makes it possible to define trials using GUI on a single file and then use the same trial definition mat-file on other datasets with the same triggers.

The third trial definition option is only relevant for studies of steady-state data (i.e. spectra). It is possible to break the data into arbitrary segments with length defined by the user.

If the peri-stimulus time starts before zero, the epoched data will be baseline-corrected by default, i.e. the mean of the pre-stimulus time is subtracted from the whole trial. The baseline correction can be removed if undesirable by setting 'Baseline correction' to 'no'.

The 'Event padding' option makes it possible to store with each trial some events that occur around it within the specified interval. This is useful e.g. when a response comes long time after a stimulus but needs to be included with the trial e.g. to compute the reaction time at a later stage.

'Filename Prefix' option is for specifying the prefix to be added to the dataset name after epoching. The default prepended output letter is 'e'.

### 12.8.2 Filtering the data: `spm_eeg_filter`

Continuous or epoched data can be filtered, over time, with a low-, high-, stop- or bandpass-filter. SPM uses a Butterworth filter to do this. Phase delays are minimised by using MATLAB's `filtfilt` function which filters the data twice, forwards and backwards. The batch interface makes it possible to define filter type, band, cutoff frequencies, direction and order. The default prepended output letter is 'f'.

### 12.8.3 Baseline correction: `spm_eeg_bc`

This function subtracts the baseline from channel data. You will be asked to specify the baseline period in ms (e.g. [-100 0]). A new dataset will be written out with the name prepended by 'b'.

### 12.8.4 Artefact detection and rejection: `spm_eeg_artefact`

Some trials not only contain neuronal signals of interest, but also a large amount of signal from other sources like eye movements or muscular activity. These signal components are referred to as artefacts. There are many kinds of artefacts and many methods for detecting them. The artefact detection function in SPM is, therefore, extendable and can automatically detect and use plugin functions that implement particular detection algorithms. Simple algorithms presently implemented include thresholding of the data, thresholding of the difference between adjacent samples (to detect jumps), thresholding peak-to-peak amplitude and detection of flat segments. Channels containing artefacts in large proportion of the trials are automatically marked as bad.

Note that the function only indicates which trials are artefactual or clean and subsequent processing steps (e.g. averaging) will take this information into account. However, no data is actually removed from the \*.dat file. The \*.dat file is actually copied over without any change. The default prepended output letter is 'a'.

Click on "File name" and select the dataset. Double click "How to look for artefacts" and a new branch will appear. It is possible to define several sets of channels to scan and one of the

several different methods for artefact detection. For each detection method there are specific configuration parameter (e.g. for thresholding - the threshold value).

### 12.8.5 Downsampling: `spm_eeg_downsample`

The data can be downsampled to any sampling rate. This is useful if the data were acquired at a higher sampling rate than one needs for making inferences about low-frequency components. For example, resampling from 1000 Hz to 200 Hz would cut down the resulting file size to 20% of the original file size. The prepended output letter is `'d'`.

### 12.8.6 Rereferencing: `spm_eeg_montage`

Sometimes it is necessary to re-reference the data to a new reference. For sensor level analysis it can be useful to use a reference that emphasizes the effects of interest. In SPM this is done by specifying a weight matrix, which pre-multiplies the data. This is a general approach which allows one to re-reference to the average over channels, to single channels, or any linear combination of channels, e.g. the average over a pair of channels. The prepended output letter is `'M'`.

The montage function has several 'modes' of action which have to do with the fact that it is possible to use online montages which do not actually change the data on disk but apply the montage on line when reading the data. M/EEG objects with online montage appear as if the montage has been applied (e.g they have channel labels corresponding to the post-montage state). The most common mode 'Write' does not use online montage but rather applies montage to the data and generates a new dataset. 'Switch' can switch between previously defined montages, 'Add' can add a montage to the set of montages in the dataset without switching to it and 'Clear' removes online montages and returns back to the original channel set.

A montage is specified by a `textttmat`-file, which contains a struct with 3 fields: `labelnew` (labels of new channels), `labelorg` (labels of original channels), and the montage-matrix `tra` ("tra" as in transform). Montages files can be generated and edited using 'Prepare' GUI.

Finally, you need to specify whether you want to "keep the other channels". There may be channels that are not involved in the montage. For instance, if you the apply montage defined for your EEG channels but there are also EOG or trigger channels in the file. If you answer "yes", they will just be copied to the new file unaffected. If you answer "no" they will not be included in the new file.

### 12.8.7 Grand mean: `spm_eeg_grandmean`

The grand mean is usually understood as the average of evoked responses over subjects. The grand mean function in SPM is typically used to do exactly this, but can also be used to average over multiple EEG files, e.g. multiple sessions of a single subject. There is an option to either do averaging weighted by the number of trials in each file (suitable for averaging accross sessions within a subject) or do unweighted averaging (suitable for averaging accross subjects).

You will need to specify the name of the output file.

### 12.8.8 Merge: `spm_eeg_merge`

Merging several MEEG files can be useful for concatenating multiple sessions of a single subject. Another use is to merge files and then use the display tool on the concatenated file to be able to display in the same graph data coming from different files. This is the preferred way in SPM to display data together that is split up into several files. The merged file will be written into the same directory as the first selected file. The prepended output letter is `'c'`.

You should specify what to do with condition labels. The simplest option is to keep them the same. This might be useful for instance when you have several sessions for one subject with the same conditions in all files. In other cases, however, it might be helpful to rename the conditions like "condition A" to something like "condition A, session 1", etc. The simplest way to do it is to append the name of the original file to the condition labels. There is also a possibility to specify more sophisticated 'recoding rules' (see the documentation in the function header). This is mostly useful for writing scripts.

The function will first check whether there are at least two files, and whether all data are consistent with each other, i.e., have the same number of channels, time points, and sampling rate.

### 12.8.9 Multimodal fusion: `spm_eeg_fuse`

SPM supports datasets containing simultaneously recorded MEG and EEG. For imaging source reconstruction it is possible to use both modalities to inform the source solution. Usually combined MEG/EEG data is contained within the same raw dataset and can be pre-processed together from the beginning. If this is not the case `spm_eeg_fuse` makes it possible to combine two datasets with different channels into a single dataset given that the sets of channels do not overlap and the datasets are identical in the other dimensions (i.e. have the same sampling rate and time axis, the same number of trials and the same condition labels in the same order). This function can be used to create a multimodal dataset also from separately recorded MEG and EEG which is a valid thing to do in the case an experiment with highly reproducible ERP/ERF.

### 12.8.10 Cropping: `spm_eeg_crop`

The 'Crop' function can be used to remove part of the data, particularly the edges of trials. This is useful if the trials were originally made longer than necessary to absorb edge artefacts like filter ringing and at a later stage of processing the padding can be removed. It is also possible to remove frequencies and channels.

### 12.8.11 Combine planar: `spm_eeg_combineplanar`

This functionality is specific to MEG systems with planar gradiometers (of which most common is the Neuromag system). It can also be applied to MEG datasets converted to synthetic planar gradient. Planar gradiometers come in pairs corresponding to two directions of the magnetic field in a plain parallel to the head surface. To interpret the data from this sensors it is convenient to combine them. This can be done for data in the time domain, in which case RMS is computed or time-frequency data in which case the data for the two directions are summed. Note that it is important to put the 'Combine planar' step in the right place in the pipeline. For instance it would not make sense to filter the combined planar data, so filtering must be done first. For time-frequency case the 'Combine planar' step would usually precede the rescaling step. Note also that combining planar channels is a nonlinear step so these channels cannot be used for source reconstruction or DCM.

You should choose whether to replace the original planar channels with combined, add them in addition to the original ones, replace all MEG channels with combined planar or only keep combined planar and discard all others.

### 12.8.12 Data reduction: `spm_eeg_reduce`

This function can be used to reduce data dimensionality by projecting the data on a small number of spatial components (e.g. PCA). This is an extendable function where new reduction methods can be added.

### 12.8.13 Time-frequency decomposition: `spm_eeg_tf`

The time-frequency decomposition is extendable and can automatically detect and use plugin functions that implement particular spectral estimation algorithms. Algorithms presently implemented include continuous Morlet wavelet transform, Hilbert transform and multitaper spectral estimation. The result is written to one or two result files, one containing the instantaneous power and the other, optionally written, the phase estimates (phase estimation is not possible for all algorithms). One can select the channels and frequencies for which power and phase should be estimated. For power, the prepended output letters are `tf_`, for phase `tph_`.

### 12.8.14 Rescaling and baseline correction of time-frequency: `spm_eeg_tf_rescale`

Usually raw event-related power is not the most informative thing to look at (although contrasts of raw power between conditions can be informative). To see the event-related effects better the power should be either transformed or baseline-corrected separately for each frequency. There are several different ways to do this and they are implemented in `spm_eeg_tf_rescale` function. 'LogR' method first computes the log of power and then baseline-corrects and scales the result to produce values in dB. 'Diff' just does simple baseline subtraction. 'Rel' expresses the power in % of the baseline units. Finally 'Log' and 'Sqrt' options just compute the respective functions without baseline-correction. If necessary, you will need to specify the baseline period. Optionally the baseline can come from a different dataset. This can be useful e.g. to baseline correct response-locked data using a pre-stimulus baseline.

### 12.8.15 Averaging over time or frequency: `spm_eeg_avgtime`, `spm_eeg_avgfreq`

These functions can be used to average time-frequency data over time or frequency and save the results as M/EEG datasets. This is useful when subsequently additional processing steps need to be carried out (e.g. rescaling).

### 12.8.16 Averaging: `spm_eeg_average`

Averaging of single trial data is the crucial step to obtain the evoked or induced response. When averaging single trial data, single trials are averaged within condition. The prepended output letter is 'm'.

Optionally you can use robust averaging for your data. This approach estimates weights, lying between 0 and 1, that indicate how artefactual a particular sample in a trial is. Later on, when averaging to produce evoked responses, each sample is weighted by this number. For example, if the weight of a sample is close to zero, it doesn't have much influence in the average, and is effectively treated like an artefact. If you choose robust averaging, you will be given an option to save the weights as a separate dataset which is useful for finding out what parts of the data were downweighted and adjusting the parameters if necessary. Then you should choose whether to compute the weights by condition (as opposed to for all the trials pooled together). When there are approximately equal numbers of trials in each condition, it is probably safer to compute weights across all conditions, so as not to introduce artifactual differences between conditions. However, if one condition has fewer trials than the others, it is likely to be safer to estimate the weights separately for each condition, otherwise evoked responses in the rarer condition will be downweighted so as to become more similar to the more common condition(s). Finally, you will have to choose an offset for the weighting function. This value, default value 3, defines the weighting function used for averaging the data. The value 3 will roughly preserve 95% of data points drawn randomly from a Gaussian distribution. Robust averaging can be applied to either time or time-frequency data. In the case of time data if you applied a low-pass filter before averaging it is advised to apply it again after averaging because differential weighting of adjacent points may re-introduce high-frequencies into the data.

For phase data it is possible to compute phase-locking value (a.k.a. inter-trial coherence) instead of averaging the phase.

### 12.8.17 Contrast over epochs: `spm_eeg_contrast`

As an extension to the averaging functionality, SPM can also be used to compute linear combinations of single trials or evoked responses. For example, if you want to compute the difference between two evoked responses, you supply a contrast vector of  $[-1; 1]$ . Similarly, if you want to remove some trials from the file, you can do this by using a contrast vector like  $[1; 0]$  which would write a new file with only the first evoked response. The prepended output letter is 'w'.

For each contrast, you have to enter a label and a vector of weights of the same length as the number of trial types in the file. Note that SPM will zero-pad this vector if you specify fewer contrast weights than you have trials. You also need to decide whether to "Weight by

replications”. This is important when you use this function on single trials, where, typically, you have a different number of trials for each trial type. If you then choose to average over multiple trials, this option allows you to choose whether you want to form an average that is weighted by the number of measurements within each trial type. The ‘yes’ choice is useful when combining multiple conditions into one. The ‘no’ choice is useful when computing differences between responses.

### 12.8.18 Copy: `spm_eeg_copy`

This function makes it possible to make a copy of a dataset. It won’t work just to copy and rename the files because the name of the data file is stored in the header file and this should be updated. You will be asked to specify the new dataset name.

### 12.8.19 Remove bad trials: `spm_eeg_remove_bad_trials`

This function physically removes trials marked as bad from a dataset. This can be useful, for instance, before time-frequency computation as processing bad trials generates a lot of overhead. Also under any other circumstances when it is necessary to remove trials from a dataset (for instance to get rid of some unused condition) these trials can be first marked as bad and then removed using this function.

## 12.9 Displaying data with SPM M/EEG REVIEW

This tool can be called from the main SPM GUI under “Display” → M/EEG.

SPM M/EEG REVIEW is meant to provide the user with basic visualization (data and source reconstruction) and reviewing (e.g. trial and sensor good/bad status) tools.

When called, SPM M/EEG REVIEW displays in the SPM graphics window information about the SPM data file which is displayed (only for MATLAB versions  $\geq 7.4$ ).

SPM M/EEG REVIEW uses tabs to easily access different fields in the SPM data file structure (see relevant SPM manual section for SPM EEG data format). The main tabs system, at the top of the graphics windows, offers the following alternatives:

- **EEG** displays EEG type data (if any). These are the data associated with “EEG” sensors. The content of this tab is described below, as well as the “MEG” and “OTHER” tabs.
- **MEG** displays MEG type data (if any).
- **MPLANAR** displays MEG data from planar gradiometers (if any).
- **MCOMB** displays RMS MEG data from combined planar gradiometers (if any).
- **OTHER** displays any other type of data (e.g. HEOG, VEOG, etc).
- **info** (active tab by default): displays basic information about the data file. This tab contains three further sub-tabs<sup>4</sup>: “channels”, “trials” and “inv” (the latter shows source reconstructions parameters, if any). Some of this info can be changed by the user (e.g. sensor/trial<sup>5</sup> type, label and status, etc) by editing the table. The changes become effective when clicking on “update”. They are actually saved in the data file when clicking on “SAVE”.
- **source** displays source reconstructions (if any). See below (2- source reconstructions visualization).

In addition, the user can call the SPM PREPARE routine <sup>6</sup> or save any modification in the data file using the top-right buttons “Prepare SPM file” and “SAVE”.

<sup>4</sup>Users can also check sensor coregistration when clicking on “sensor positions”.

<sup>5</sup>Sensor/trial status (good/bad) can also be changed under the EEG/MEG/OTHER tabs, when visualizing trials (sensor: right-click uicontextmenu ; trials: button 10).

<sup>6</sup>This is part of the SPM EEG preprocessing tools. It mainly concerns the coregistration of the sensors onto the normalized SPM space. See relevant section in the SPM manual.

### 12.9.1 Data visualization

The graphics window of SPM REVIEW offers two modes of data visualization: “scalp” and “standard” (default). For continuous (non-epoched) data, only “standard” mode is enabled. For time-frequency data, only “scalp” mode is enabled. For any other type of data, the user can switch to any of these modes using the standard/scalp radio button. These two modes are described below:

- **standard** channels are displayed vertically, within the same axes. A channel uicontextmenu can be accessed by right clicking on any time series (e.g. for changing the channel good/bad status). An additional axis (bottom right) provides the user with the temporal and horizontal scale of the displayed data). The size of the plotted time window can be changed using the top left buttons 1 and 2. User can scroll through the data using the temporal slider, at the bottom of the graphics window. A global display scaling factor can be changed using the top buttons 3 and 4. Zooming within the data is done by clicking on button 5. Clicking on button 6 displays a 2D scalp projection of the data.

When displaying epoched data, the user can select the trial within the list of accessible trials (top right of the window). It is also possible to switch the status of trials (good/bad) by clicking on button 10.

When displaying continuous data, SPM M/EEG REVIEW allows the user to manage events and selections. After having clicked on button 7, the user is asked to add a new event in the data file, by specifying its temporal bounds (two mouse clicks within the display axes). Basic properties of any events can be accessed either in the “info” table, or by right-clicking on the event marker (vertical line or patch superimposed on the displayed data). This gives access to the event uicontextmenu (e.g. for changing the event label). Buttons 8 and 9 allow the user to scroll through the data from marker to marker (backward and forward in time).

- **scalp** channels are displayed vertically, within the same axes. A channel uicontextmenu can be accessed by right clicking on any time series (e.g. for changing the channel good/bad status). An additional axis (bottom right) provides the user with the temporal and horizontal scale of the displayed data). The size of the plotted time window can be changed using the top left buttons 1 and 2. User can scroll through the data using the temporal slider, at the bottom of the graphics window. A global display scaling factor can be changed using the top buttons 3 and 4. Zooming within the data is done by clicking on button 5. Clicking on button 6 displays a 2D scalp projection of the data.

When displaying epoched data, the user can select the trial within the list of accessible trials (top right of the window). It is also possible to switch the status of trials (good/bad) by clicking on button 10.

### 12.9.2 Source reconstructions visualization

SPM M/EEG REVIEW makes use of sub tabs for any source reconstruction that has been stored in the data file<sup>7</sup>. Since these reconstructions are associated with epoched data, the user can choose the trial he/she wants to display using the list of accessible events (top of the main tab). Each sub tab has a label given by the corresponding source reconstruction comment which is specified by the user when source reconstructing the data (see relevant section in the SPM manual). The bottom-left part of each sub tab displays basic infos about the source reconstruction (date, number of included dipoles, number of temporal modes, etc). The top part of the window displays a rendering of the reconstruction on the cortical surface that has been used. User can scroll through peri-stimulus time by using the temporal slider below the rendered surface. Other sliders allow the user to (i) change the transparency of the surface (left slider) and (ii) threshold the colormap (right sliders). In the center, a butterfly plot of the reconstructed intensity of cortical source activity over peri-stimulus time is displayed. If the data file contains more than one source reconstruction, the bottom-right part of the window displays a bar graph of the model evidences of each source reconstruction. This provides the user with a *visual* Bayesian model comparison

<sup>7</sup>This concerns any distributed source reconstruction, i.e. also includes imaging DCM analyses, but not ECD reconstructions (so far).



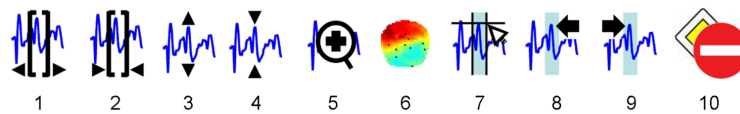


Figure 12.1: *SPM M/EEG REVIEW* buttons legend 1-2: increase/decrease width of plotted time window, 3-4: increase/decrease global scaling display factor, 5: zoom in, 7: add event, 8-9: scroll backward/forward data from marker to marker, 10: declare event as good/bad

tool<sup>8</sup>. SPM M/EEG REVIEW allows quick and easy switching between different models and trials, for a visual comparison of cortical source activities.

### 12.9.3 Script generation

Another way of batching jobs is by using scripts, written in MATLAB . You can generate these scripts automatically. To do this, you first have to analyze one data set using the GUI or batch system. Whenever a preprocessing function is called, all the input arguments, once they have been assembled by the GUI, are stored in a “history”. This history can then be used to not only see in detail which functions have been used on a data set, but also to generate a script that repeats the same analysis steps. The big difference is that, this time, no more GUI interactions are necessary because the script already has all the input arguments which you gave during the first run. The history of an `meeeg` object can be accessed by `D.history`.

To generate a script from the history of an SPM MEEG file, open the file in the M/EEG REVIEW facility and select the `info` tab: a `history` tab is then available that will display all the history of the file. Clicking the `SAVE AS SCRIPT` button will ask for the filename of the MATLAB script to save and the list of processing steps to save (default is all but it is possible to select only a subset of them). This will generate a script, which, when run, repeats the analysis. The script can also be obtained by directly calling the function `spm_eeg_history`.

Of course, this script can not only be used to repeat an analysis, but the script can also be seen as a template that can be re-used for other analyses. One needs minimal MATLAB knowledge for these changes. For example, you can replace the filenames to preprocess a different subject. Or you can change parameters and then re-run the analysis. We have prepared an example, using the same example data set, as in the previous subsection to demonstrate this (see the file `man\example_scripts\history_subject1.m`). With a script, you could also use object methods directly, e.g. add a line like `D=badchannels(D, 23, 1)`, which flags channel 23 as bad (see also our example script after the filtering step). To run the example script on your computer, you need the data set that you can download from the SPM webpage (<sup>9</sup>).

<sup>8</sup>Remember that model evidences  $p(y|m)$  can only be compared for the same data. Therefore, if the source reconstructions have different time windows, filters, number of temporal modes, etc., the comparison does not hold. This is why basic information (bottom-left part of the window) has to be recalled when comparing models.

<sup>9</sup>[http://www.fil.ion.ucl.ac.uk/spm/data/eeg\\_mmn/](http://www.fil.ion.ucl.ac.uk/spm/data/eeg_mmn/)



## Chapter 13

# Analysis in sensor space

This chapter describes how to perform statistical analyses of EEG/MEG data. This requires transforming data from SPM M/EEG format to image files (NIfTI format). Once the data are in image format the analyses for M/EEG are procedurally identical to 2nd level analyses for fMRI. We therefore refer the reader to the fMRI section for further details of this last step.

In the drop down “Images” menu, select the function **Convert to images**. This will open the batch tool for conversion to images. You will need to select the input dataset, that can be either a mat-file on disk or a dependency from a previous processing step.

Then you need to set the ‘mode’ of conversion. M/EEG data in general case can be up to 5-dimensional (3 spatial dimensions, time and frequency). SPM statistical machinery can only handle up to 3 dimensions. Although this is a purely implementational limitation and the theory behind SPM methods can be extended to any dimensionality, in practice high-dimensional statistical results can be very hard to interpret not least due to our inability as humans to visualise them. Furthermore, unconstrained high-dimensional test would incur very severe penalty for multiple comparisons and should in most case be avoided. Thus, our purpose is to reduce our data dimensionality to be 3 or less. The three spatial dimensions in which the sensors reside can be reduced to two by projecting their locations onto a plane. Further reduction of dimensionality will involve averaging over one of the dimensions. The choices for ‘mode’ option correspond to all the different possibilities to average over a subset of data dimensions. Some of the options are only relevant for time-frequency data where the frequency dimension is present.

‘Conditions’ options makes it possible to only convert data for a subset of conditions in the file. This is especially useful for batch pipeline building. The conversion module outputs as a dependency a list of all the generated NIfTI images. These can be used as input to subsequent steps (e.g. statistical design specification). By including the ‘Convert2images’ module several times in batch each condition can have a separate dependency and enter in a different place in the statistical design (e.g. for two-sample t-test between two groups of trials).

The ‘Channels’ option makes it possible to select a subset of channels for conversions. These can be either selected by modality (e.g. ‘EEG’) or chosen by name of by a list in a mat-file (e.g. to average over all occipital channels).

‘Time window’ and ‘Frequency window’ options limit the data range for conversion which is especially important if the data are averaged over this range. Make sure you only include the range of interest.

Finally the ‘Directory prefix’ option specifies the prefix for the directory where images will be written out. This is important if several different groups of images are generated from the same dataset (e.g. from different modalities or different channel groups).

### 13.0.4 Output

When running the tool a directory will be created at the dataset location. Its name will be the name of the dataset with the specified prefix. In this directory there will be a nii-file for each condition. In the case of averaged dataset these will be 3D images (where some dimensions can have size of 1). In the case of an epoched dataset there will be 4D-NIfTI images where every

frame will contain a trial.

### **Averaging over time or frequency**

Although 2D scalp images averaged over time or frequency dimension can be created directly in conversion to images, they can also be generated by averaging over part of the Z dimension of previously created 3D images. This is done via 'Collapse time' tool in the 'Images' menu.

### **Masking**

When you set up your statistical analysis, it might be useful to use an explicit mask to limit your analysis to a fixed time window of interest. Such a mask can be created by selecting **Mask images** from "Images" dropdown menu. You will be asked to provide one unsmoothed image to be used as a template for the mask. This can be any of the images you exported. Then you will be asked to specify the time (or frequency) window of interest and the name for the output mask file. This file can then enter in your statistical design under the 'Explicit mask' option or when pressing the 'small volume' button in the 'Results' GUI and choosing the 'image' option to specify the volume.

### **13.0.5 Smoothing**

The images generated from M/EEG data must be smoothed prior to second level analysis using the **SMOOTH IMAGES** function in the drop down "Images" menu. Smoothing is necessary to accommodate spatial/temporal variability between subjects and make the images better conform to the assumptions of random field theory. The dimensions of the smoothing kernel are specified in the units of the original data (e.g. [mm mm ms] for space-time, [Hz ms] for time-frequency). The general guiding principle for deciding how much to smooth is the matched filter idea, which says that the smoothing kernel should match the data feature one wants to enhance. Therefore, the spatial extent of the smoothing kernel should be more or less similar to the extent of the dipolar patterns that you are looking for (probably something of the order of magnitude of several cm). In practice you can try to smooth the images with different kernels designed according to the principle above and see what looks best. Smoothing in time dimension is not always necessary as filtering the data has the same effect. For scalp images you should set the 'Implicit masking' option to 'yes' in order to keep excluding the areas outside the scalp from the analysis.

Once the images have been smoothed one can proceed to the second level analysis.

# Chapter 14

## 3D source reconstruction: Imaging approach

This chapter describes an Imaging approach to 3D source reconstruction.

### 14.1 Introduction

This chapter focuses on the imaging (or distributed) method for implementing EEG/MEG source reconstruction in SPM. This approach results in a spatial projection of sensor data into (3D) brain space and considers brain activity as comprising a very large number of dipolar sources spread over the cortical sheet, with fixed locations and orientations. This renders the observation model linear, the unknown variables being the source amplitudes or power.

Given epoched and preprocessed data (see chapter 12), the evoked and/or induced activity for each dipolar source can be estimated, for a single time-sample or a wider peristimulus time window.

The obtained reconstructed activity is in 3D voxel space and can be further analyzed using mass-univariate analysis in SPM.

Contrary to PET/fMRI data reconstruction, EEG/MEG source reconstruction is a non trivial operation. Often compared to estimating a body shape from its shadow, inferring brain activity from scalp data is mathematically ill-posed and requires prior information such as anatomical, functional or mathematical constraints to isolate a unique and most probable solution [11].

Distributed linear models have been around for several decades now [21] and the proposed pipeline in SPM for an imaging solution is classical and very similar to common approaches in the field. However, at least two aspects are quite original and should be emphasized here:

- Based on an empirical Bayesian formalism, the inversion is meant to be generic in the sense it can incorporate and estimate the relevance of multiple constraints of varied nature; data-driven relevance estimation being made possible through Bayesian model comparison [41, 88, 72, 35].
- The subject's specific anatomy is incorporated in the generative model of the data, in a fashion that eschews individual cortical surface extraction. The individual cortical mesh is obtained automatically from a canonical mesh in MNI space, providing a simple and efficient way of reporting results in stereotactic coordinates.

The EEG/MEG imaging pipeline is divided into four consecutive steps which characterize any inverse procedure with an additional step of summarizing the results. In this chapter, we go through each of the steps that need completing when proceeding with a full inverse analysis:

1. Source space modeling,
2. Data co-registration,
3. Forward computation,

4. Inverse reconstruction.
5. Summarizing the results of inverse reconstruction as an image.

Whereas the first three steps are part of the whole generative model, the inverse reconstruction step consists in Bayesian inversion, and is the only step involving actual EEG/MEG data.

## 14.2 Getting started

Everything which is described hereafter is accessible from the SPM user-interface by choosing the “EEG” application, **3D Source Reconstruction** button. When you press this button a new window will appear with a GUI that will guide you through the necessary steps to obtain an imaging reconstruction of your data. At each step, the buttons that are not yet relevant for this step will be disabled. When you open the window the only two buttons you can press are **Load** which enables you to load a pre-processed SPM MEEG dataset and the **Group inversion** button that will be described below. You can load a dataset which is either epoched with single trials for different conditions, averaged with one event related potential (ERP) per condition, or grand-averaged. An important pre-condition for loading a dataset is that it should contain sensors and fiducials. This will be checked when you load a file and loading will fail in case of a problem. You should make sure that for each modality present in the dataset as indicated by channel types (either EEG or MEG) there is a sensor description. If, for instance, you have an MEG dataset with some EEG channels that you don’t actually want to use for source reconstruction, change their type to “*LFP*” or “*Other*” before trying to load the dataset (the difference is that *LFP* channels will still be filtered and available for artefact detection whereas *Other* channels won’t). MEG datasets converted by SPM from their raw formats will always contain sensor and fiducial descriptions. In the case of EEG for some supported channel setups (such as extended 10-20 or BioSemi) SPM will provide default channel locations and fiducials that you can use for your reconstruction. Sensor and fiducial descriptions can be modified using the **Prepare** interface and in this interface you can also verify that these descriptions are sensible by performing a coregistration (see chapter 12 and also below for more details about coregistration).

When you successfully load a dataset you are asked to give a name to the present analysis cell. In SPM it is possible to perform multiple reconstructions of the same dataset with different parameters. The results of these reconstructions will be stored with the dataset if you press the **Save** button. They can be loaded and reviewed again using the 3D GUI and also with the SPM EEG REVIEW tool. From the command line you can access source reconstruction results via the `D.inv` field of the `meeg` object. This field (if present) is a cell array of structures and does not require methods to access and modify it. Each cell contains the results of a different reconstruction. In the GUI you can navigate between these cells using the buttons in the second row. You can also create, delete and clear cells. The label you input at the beginning will be attached to the cell for you to identify it.

## 14.3 Source space modeling

After entering the label you will see the **Template** and **MRI** button enabled. The **MRI** button will create individual head meshes describing the boundaries of different head compartments based on the subject’s structural scan. SPM will ask for the subject’s structural image. It might take some time to prepare the model as the image needs to be segmented. The individual meshes are generated by applying the inverse of the deformation field needed to normalize the individual structural image to MNI template to canonical meshes derived from this template. This method is more robust than deriving the meshes from the structural image directly and can work even when the quality of the individual structural images is low.

Presently we recommend the **Template** button for EEG and a head model based on an individual structural scan for MEG. In the absence of individual structural scan combining the template head model with the individual headshape also results in a quite precise head model.

The **Template** button uses SPM's template head model based on the MNI brain. The corresponding structural image can be found under `canonical\single_subj_T1.nii` in the SPM directory. When you use the template, different things will happen depending on whether your data is EEG or MEG. For EEG, your electrode positions will be transformed to match the template head. So even if your subject's head is quite different from the template, you should be able to get good results. For MEG, the template head will be transformed to match the fiducials and headshape that come with the MEG data. In this case having a headshape measurement can be quite helpful in providing SPM with more data to scale the head correctly. From the user's perspective the two options will look quite similar.

No matter whether the MRI or **Template** button was used the cortical mesh, which describes the locations of possible sources of EEG and MEG signal, is obtained from a template mesh. In the case of EEG the mesh is used as is, and in the case of MEG it is transformed with the head model. Three cortical mesh sizes are available "coarse", "normal" and "fine" (5124, 8196 and 20484 vertices respectively). It is advised to work with the "normal" mesh. Choose "coarse" if your computer has difficulties handling the "normal" option. "Fine" will only work on 64-bit systems and is probably an overkill.

## 14.4 Coregistration

In order for SPM to provide a meaningful interpretation of the results of source reconstruction, it should link the coordinate system in which sensor positions are originally represented to the coordinate system of a structural MRI image (MNI coordinates). In general, to link between two coordinate systems you will need a set of at least 3 points whose coordinates are known in both systems. This is a kind of *Rosetta stone* that can be used to convert a position of any point from one system to the other. These points are called "fiducials" and the process of providing SPM with all the necessary information to create the *Rosetta stone* for your data is called "coregistration".

There are two possible ways of coregistrating the EEG/MEG data into the structural MRI space.

1. A Landmark based coregistration (using fiducials only).

The rigid transformation matrices (Rotation and Translation) are computed such that they match each fiducial in the EEG/MEG space into the corresponding one in sMRI space. The same transformation is then applied to the sensor positions.

2. Surface matching (between some headshape in MEG/EEG space and some sMRI derived scalp tessellation).

For EEG, the sensor locations can be used instead of the headshape. For MEG, the headshape is first coregistrated into sMRI space; the inverse transformation is then applied to the head model and the mesh.

Surface matching is performed using an Iterative Closest Point algorithm (ICP). The ICP algorithm [12] is an iterative alignment algorithm that works in three phases:

- Establish correspondence between pairs of features in the two structures that are to be aligned based on proximity;
- Estimate the rigid transformation that best maps the first member of the pair onto the second;
- Apply that transformation to all features in the first structure. These three steps are then reapplied until convergence is concluded. Although simple, the algorithm works quite effectively when given a good initial estimate.

In practice what you will need to do after pressing the **Coregister** button is to specify the points in the sMRI image that correspond to your M/EEG fiducials. If you have more fiducials (which may happen for EEG as in principle any electrode can be used as a fiducial), you will be asked at the first step to select the fiducials you want to use. You can select more than 3, but not less. Then for each M/EEG fiducial you selected you will be asked to specify the corresponding position in the sMRI image in one of 3 ways.

- **select** - locations of some points such as the commonly used nasion and preauricular points and also CTF recommended fiducials for MEG (as used at the FIL) are hard-coded in SPM. If your fiducial corresponds to one of these points you can select this option and then select the correct point from a list.
- **type** - here you can enter the MNI coordinates for your fiducial ( $1 \times 3$  vector). If your fiducial is not on SPM's hard-coded list, it is advised to carefully find the right point on either the template image or on your subject's own image normalized to the template. You can do it by just opening the image using SPM's Display/images functionality. You can then record the MNI coordinates and use them in all coregistrations you need to do using the "type" option.
- **click** - here you will be presented with a structural image where you can click on the right point. This option is good for "quick and dirty" coregistration or to try out different options.

You will also have the option to skip the current fiducial, but remember you can only do it if you eventually specify more than 3 fiducials in total. Otherwise the coregistration will fail.

After you specify the fiducials you will be asked whether to use the headshape points if they are available. For EEG it is advised to always answer "yes". For MEG if you use a head model based on the subject's sMRI and have precise information about the 3 fiducials (for instance by doing a scan with fiducials marked by vitamin E capsules) using the headshape might actually do more harm than good. In other cases it will probably help, as in EEG.

The results of coregistration will be presented in SPM's graphics window. It is important to examine the results carefully before proceeding. In the top plot you will see the scalp, the inner skull and the cortical mesh with the sensors and the fiducials. For EEG make sure that the sensors are on the scalp surface. For MEG check that the head position in relation to the sensors makes sense and the head does not for instance stick outside the sensor array. In the bottom plot the sensor labels will be shown in topographical array. Check that the top labels correspond to anterior sensors, bottom to posterior, left to left and right to right and also that the labels are where you would expect them to be topographically.

## 14.5 Forward computation (*forward*)

This refers to computing for each of the dipoles on the cortical mesh the effect it would have on the sensors. The result is a  $N \times M$  matrix where  $N$  is the number of sensors and  $M$  is the number of mesh vertices (that you chose from several options at a previous step). This matrix can be quite big and it is, therefore, not stored in the header, but in a separate `*.mat` file which has `SPMgainmatrix` in its name and is written in the same directory as the dataset. Each column in this matrix is a so called "lead field" corresponding to one mesh vertex.

The lead fields are computed using the "forwinv" toolbox<sup>1</sup> developed by Robert Oostenveld, which SPM shares with FieldTrip. This computation is based on Maxwell's equations and makes assumptions about the physical properties of the head. There are different ways to specify these assumptions which are known as "forward models".

The "forwinv" toolbox can support different kinds of forward models. When you press **Forward Model** button (which should be enabled after successful coregistration), you will have a choice of several head models depending on the modality of your dataset. We presently recommend using a single shell model for MEG and "EEG BEM" for EEG. You can also try other options and compare them using model evidence (see below). The first time you use the EEG BEM option with a new structural image (and also the first time you use the **Template** option) a lengthy computation will take place that prepares the BEM model based on the head meshes. The BEM will then be saved in a quite large `*.mat` file with ending `_EEG_BEM.mat` in the same directory with the structural image ("canonical" subdirectory of SPM for the template). When the head model is ready, it will be displayed in the graphics window with the cortical mesh and sensor locations you should verify for the final time that everything fits well together.

The actual lead field matrix will be computed at the beginning of the next step and saved. This is a time-consuming step and it takes longer for high-resolution meshes. The lead field file

<sup>1</sup>forwinv: <http://fieldtrip.fcdonders.nl/development/forwinv>



will be used for all subsequent inversions if you do not change the coregistration and the forward model.

## 14.6 Inverse reconstruction

To get started press the **Invert** button. The first choice you will see is between **Imaging**, VB-ECD and DCM. For reconstruction based on an empirical Bayesian approach to localize either the evoked response, the evoked power or the induced power, as measured by EEG or MEG press the **Imaging** button. The other options are explained in greater detail elsewhere.

If you have trials belonging to more than one condition in your dataset then the next choice you will have is whether to invert all the conditions together or to choose a subset. It is recommended to invert the conditions together if you are planning to later do a statistical comparison between them. If you have only one condition, or after choosing the conditions, you will get a choice between “Standard” and “Custom” inversion. If you choose “Standard” inversion, SPM will start the computation with default settings. These correspond to the multiple sparse priors (MSP) algorithm [33] which is then applied to the whole input data segment.

If you want to fine-tune the parameters of the inversion, choose the “Custom” option. You will then have the possibility to choose between several types of inversion differing by their hyperprior models (IID - equivalent to classical minimum norm, COH - smoothness prior similar to methods such as LORETA) or the MSP method .

You can then choose the time window that will be available for inversion. Based on our experience, it is recommended to limit the time window to the activity of interest in cases when the amplitude of this activity is low compared to activity at other times. The reason is that if the irrelevant high-amplitude activity is included, the source reconstruction scheme will focus on reducing the error for reconstructing this activity and might ignore the activity of interest. In other cases, when the peak of interest is the strongest peak or is comparable to other peaks in its amplitude, it might be better not to limit the time window to let the algorithm model all the brain sources generating the response and then to focus on the sources of interest using the appropriate contrast (see below). There is also an option to apply a hanning taper to the channel time series in order to downweight the possible baseline noise at the beginning and end of the trial. There is also an option to pre-filter the data. Finally, you can restrict solutions to particular brain areas by loading a `*.mat` file with a  $K \times 3$  matrix containing MNI coordinates of the areas of interest. This option may initially seem strange, as it may seem to overly bias the source reconstructions returned. However, in the Bayesian inversion framework you can compare different inversions of the same data using Bayesian model comparison. By limiting the solutions to particular brain areas you greatly simplify your model and if that simplification really captures the sources generating the response, then the restricted model will have much higher model evidence than the unrestricted one. If, however, the sources you suggested cannot account for the data, the restriction will result in a worse model fit and depending on how much worse it is, the unrestricted model might be better in the comparison. So using this option with subsequent model comparison is a way, for instance, to integrate prior knowledge from the literature or from fMRI/PET/DTI into your inversion. It also allows for comparison of alternative prior models.

Note that for model comparison to be valid all the settings that affect the input data, like the time window, conditions used and filtering should be identical.

SPM imaging source reconstruction also supports multi-modal datasets. These are datasets that have both EEG and MEG data from a simultaneous recording. Datasets from the “Neuromag” MEG system which has two kinds of MEG sensors are also treated as multimodal. If your dataset is multimodal a dialogue box will appear asking to select the modalities for source reconstruction from a list. If you select more than one modality, multimodal fusion will be performed. This option based on the paper by Henson et al. [51] uses a heuristic to rescale the data from different modalities so that they can be used together.

Once the inversion is completed you will see the time course of the region with maximal activity in the top plot of the graphics window. The bottom plot will show the maximal intensity projection (MIP) at the time of the maximal activation. You will also see the log-evidence value that can be used for model comparison, as explained above. Note that not all the output of the inversion is displayed. The full output consists of time courses for all the sources and conditions

for the entire time window. You can view more of the results using the controls in the bottom right corner of the 3D GUI. These allow focusing on a particular time, brain area and condition. One can also display a movie of the evolution of neuronal activity.

## 14.7 Summarizing the results of inverse reconstruction as an image

SPM offers the possibility of writing the results as 3D NIfTI images, so that you can then proceed with GLM-based statistical analysis using Random Field theory. This is similar to the 2nd level analysis in fMRI for making inferences about region and trial-specific effects (at the between subject level).

This entails summarizing the trial- and subject-specific responses with a single 3-D image in source space. Critically this involves prompting for a time-frequency contrast window to create each contrast image. This is a flexible and generic way of specifying the data feature you want to make an inference about (e.g., gamma activity around 300 ms or average response between 80 and 120 ms). This kind of contrast is specified by pressing the `Window` button. You will then be asked about the time window of interest (in ms, peri-stimulus time). It is possible to specify one or more time segments (separated by a semicolon). To specify a single time point repeat the same value twice. The next question is about the frequency band. If you just want to average the source time course leave that at the default, zero. In this case the window will be weighted by a Gaussian. In the case of a single time point this will be a Gaussian with 8 ms full width half maximum (FWHM). If you specify a particular frequency or a frequency band, then a series of Morlet wavelet projectors will be generated summarizing the energy in the time window and band of interest.

There is a difference between specifying a frequency band of interest as zero, as opposed to specifying a wide band that covers the whole frequency range of your data. In the former case the time course of each dipole will be averaged, weighted by a gaussian. Therefore, if within your time window this time course changes polarity, the activity can average out and in an ideal case even a strong response can produce a value of zero. In the latter case the power is integrated over the whole spectrum ignoring phase, and this would be equivalent to computing the sum of squared amplitudes in the time domain.

Finally, if the data file is epoched rather than averaged, you will have a choice between “evoked”, “induced” and “trials”. If you have multiple trials for certain conditions, the projectors generated at the previous step can either be applied to each trial and the results averaged (induced) or applied to the averaged trials (evoked). Thus it is possible to perform localization of induced activity that has no phase-locking to the stimulus. It is also possible to focus on frequency content of the ERP using the “evoked” option. Clearly the results will not be the same. The projectors you specified (bottom plot) and the resulting MIP (top plot) will be displayed when the operation is completed. “trials” option makes it possible to export an image per trial which might be useful for doing within-subject statistics. The images are exported as 4D-NIfTI with one file per condition including all the trials for that condition.

The `Image` button is used to write out the contrast results. It is possible to export them as either values on a mesh (GIfTI) or volumetric 3D images (NIfTI). Both formats are supported by SPM statistical machinery. When generating an image per trial the images are exported as 4D-NIfTI with one file per condition including all the trials for that condition. The values of the exported images are normalized to reduce between-subject variance. Therefore, for best results it is recommended to export images for all the time windows and conditions that will be included in the same statistical analysis in one step. Note that the images exported from the source reconstruction are a little peculiar because of smoothing from a 2D cortical sheet into 3D volume. SPM statistical machinery has been optimized to deal with these peculiarities and get sensible results. If you try to analyze the images with older versions of SPM or with a different software package you might get different (less focal) results.

## 14.8 Rendering interface

By pressing the **Render** button you can open a new GUI window which will show you a rendering of the inversion results on the brain surface. You can rotate the brain, focus on different time points, run a movie and compare the predicted and observed scalp topographies and time series. A useful option is “virtual electrode” which allows you to extract the time course from any point on the mesh and the MIP at the time of maximal activation at this point. Just press the button and click anywhere in the brain.

An additional tool for reviewing the results is available in the SPM M/EEG REVIEW function.

## 14.9 Group inversion

A problem encountered with MSP inversion is that sometimes it is “too good”, producing solutions that were so focal in each subject that the spatial overlap between the activated areas across subjects was not sufficient to yield a significant result in a between-subjects contrast. This could be improved by smoothing, but smoothing compromises the spatial resolution and thus subverts the main advantage of using an inversion method that can produce focal solutions.

To circumvent this problem we proposed a modification of the MSP method [69] that effectively restricts the activated sources to be the same in all subjects with only the degree of activation allowed to vary. We showed that this modification makes it possible to obtain significance levels close to those of non-focal methods such as minimum norm while preserving accurate spatial localization.

The group inversion can yield much better results than individual inversions because it introduces an additional constraint for the ill-posed inverse problem, namely that the responses in all subjects should be explained by the same set of sources. Thus it should be your method of choice when analyzing an entire study with subsequent GLM analysis of the images.

Group inversion works very similarly to what was described above. You can start it by pressing the “Group inversion” button right after opening the 3D GUI. You will be asked to specify a list of M/EEG data sets to invert together. Then the routine will ask you to perform coregistration for each of the files and specify all the inversion parameters in advance. It is also possible to specify the contrast parameters in advance. Then the inversion will proceed by computing the inverse solution for all the files and will write out the output images. The results for each subject will also be saved in the header of the corresponding input file. It is possible to load this file into the 3D GUI after the inversion and explore the results as described above.

## 14.10 Batching source reconstruction

There is a possibility to run imaging source reconstruction using the SPM batch tool. It can be accessed by pressing the “Batch” button in the main SPM window and then going to “M/EEG source reconstruction” in the “SPM” under “M/EEG”. There are separate tools there for building head models, computing the inverse solution and computing contrasts and generating images. This makes it possible for instance to generate images for several different contrasts from the same inversion. All the three tools support multiple datasets as inputs. In the case of the inversion tool group inversion will be done for multiple datasets.

## 14.11 Appendix: Data structure

The MATLAB object describing a given EEG/MEG dataset in SPM is denoted as  $D$ . Within that structure, each new inverse analysis will be described by a new cell of sub-structure field  $D.inv$  and will be made of the following fields:

- **method**: character string indicating the method, either “ECD” or “Imaging” in present case;
- **mesh**: sub-structure with relevant variables and filenames for source space and head modeling;

- **datareg**: sub-structure with relevant variables and filenames for EEG/MEG data registration into MRI space;
- **forward**: sub-structure with relevant variables and filenames for forward computation;
- **inverse**: sub-structure with relevant variable, filenames as well as results files;
- **comment**: character string provided by the user to characterize the present analysis;
- **date**: date of the last modification made to this analysis.
- **gainmat**: name of the gain matrix file.

## Chapter 15

# Localization of Equivalent Current Dipoles

This chapter describes source reconstruction based on “Variational Bayes Equivalent Current Dipoles” (VB-ECDs). For more details about the implementation, please refer to the help and comments in the routines themselves, as well as the original paper by [62].

### 15.1 Introduction

3D imaging (or distributed) reconstruction methods consider all possible source location simultaneously, allowing for large and widely spread clusters of activity. This is to be contrasted with “Equivalent Current Dipole” (ECD) approaches which rely on two different hypotheses:

- only a few (say less than 5) sources are active simultaneously, and
- those sources are very focal.

This leads to the ECD model where the observed scalp potential will be explained by a handful of discrete current sources, i.e. dipoles, located inside the brain volume.

In contrast to the 3D imaging reconstruction, the number of ECDs considered in the model, i.e. the number of “active locations”, should be defined a priori. This is a crucial step, as the number of sources considered defines the ECD model. This choice should be based on empirical knowledge of the brain activity observed or any other source of information (for example by looking at the scalp potential distribution). In general, each dipole is described by 6 parameters: 3 for its location, 2 for its orientation and 1 for its amplitude. Once the number of ECDs is fixed, a non-linear optimisation algorithm is used to adjust the dipoles parameters (6 times the number of dipoles) to the observed potential.

Classical ECD approaches use a simple best fitting optimisation using “least square error” criteria. This leads to relatively simple algorithms but presents a few drawbacks:

- constraints on the dipoles are difficult to include in the framework;
- the noise cannot be properly taken into account, as its variance should be estimated alongside the dipole parameters;
- it is difficult to define confidence intervals on the estimated parameters, which could lead to over-confident interpretation of the results;
- models with different numbers of dipoles cannot be compared except through their goodness-of-fit, which can be misleading.

As adding dipoles to a model will necessarily improve the overall goodness of fit, one could erroneously be tempted to use as many ECDs as possible and to perfectly fit the observed signal. Through using Bayesian techniques, however, it is possible to circumvent all of the above limitations of classical approaches.

Briefly, a probabilistic generative model is built providing a likelihood model for the data<sup>1</sup>. The model is completed by a set of priors on the various parameters, leading to a Bayesian model, allowing the inclusion of user-specified prior constraints.

A “variational Bayes” (VB) scheme is then employed to estimate the posterior distribution of the parameters through an iterative procedure. The confidence interval of the estimated parameters is therefore directly available through the estimated posterior variance of the parameters. Critically, in a Bayesian context, different models can be compared using their evidence or marginal likelihood. This model comparison is superior to classical goodness-of-fit measures, because it takes into account the complexity of the models (e.g., the number of dipoles) and, implicitly, uncertainty about the model parameters. VB-ECD can therefore provide an objective and accurate answer to the question: Would this data set be better modelled by 2 or 3 ECDs?

## 15.2 Procedure in SPM12

This section aims at describing how to use the VB-ECD approach in SPM12.

### 15.2.1 Head and forward model

The engine calculating the projection of the dipolar sources on the scalp electrode comes from Fieldtrip and is the same for the 3D imaging or DCM. The head model should thus be prepared the same way, as described in the chapter 14. For the same data set, differences between the VB-ECD and imaging reconstructions would therefore be due to the reconstruction approach only.

### 15.2.2 VB-ECD reconstruction

To get started, after loading and preparing the head model, press the ‘Invert’ button<sup>2</sup>. The first choice you will see is between ‘Imaging’, ‘VB-ECD’ and ‘DCM’. The ‘Imaging’ reconstruction corresponds to the imaging solution, as described in chapter 14, and ‘DCM’ is described in chapter 16. Then you are invited to fill in information about the ECD model and click on buttons in the following order:

1. indicate the time bin or time window for the reconstruction, within the epoch length. Note that the data will be averaged over the selected time window! VB-ECD will thus always be calculated for a single time bin.
2. enter the trial type(s) to be reconstructed. Each trial type will be reconstructed separately.
3. add a single (i.e. individual) dipole or a pair of symmetric dipoles to the model. Each “element” (single or pair) is added individually to the model.
4. use “Informativ” or “Non-informativ” location priors. “Non-informativ” means flat priors over the brain volume. With “Informativ”, you can enter the a priori location of the source<sup>3</sup>.
5. use “Informativ” or “Non-informativ” moment priors. “Non-informativ” means flat priors over all possible directions and amplitude. With “Informativ”, you can enter the a priori moment of the source<sup>4</sup>.
6. go back to step 3 and add some more dipole(s) to the model, or stop adding dipoles.

<sup>1</sup>This includes an independent and identically distributed (IID) Normal distribution for the errors, but other distributions could be specified.

<sup>2</sup>The GUI for VB-ECD can also be launched directly from MATLAB command line with the instruction: `D = spm_eeg_inv_vbecd_gui`.

<sup>3</sup>For a pair of dipoles, only the right dipole coordinates are required.

<sup>4</sup>For a pair of dipoles, only the right dipole moment is required.

7. specify the number of iterations. These are repetitions of the fitting procedure with different initial conditions. Since there are multiple local maxima in the objective function, multiple iterations are necessary to get good results especially when non-informative location priors are chosen.

The routine then proceeds with the VB optimization scheme to estimate the model parameters. There is graphical display of the intermediate results. When the best solution is selected the model evidence will be shown at the top of the SPM Graphics window. This number can be used to compare solutions with different priors.

Results are finally saved into the data structure `D` in the field `.inv{D.val}.inverse` and displayed in the graphic window.

### 15.2.3 Result display

The latest VB-ECD results can be displayed again through the function `D = spm_eeg_inv_vbecd_disp`. If a specific reconstruction should be displayed, then use: `spm_eeg_inv_vbecd_disp('Init',D,ind)`. In the GUI you can use the 'dip' button (located under the 'Invert' button) to display the dipole locations.

In the upper part, the 3 main figures display the 3 orthogonal views of the brain with the dipole location and orientation superimposed. The location confidence interval is described by the dotted ellipse around the dipole location on the 3 views. It is not possible to click through the image, as the display is automatically centred on the dipole displayed. It is possible though to zoom into the image, using the right-click context menu.

The lower left table displays the current dipole location, orientation (Cartesian or polar coordinates) and amplitude in various formats.

The lower right table allows for the selection of trial types and dipoles. Display of multiple trial types and multiple dipoles is also possible. The display will center itself on the average location of the dipoles.





## Chapter 16

# Dynamic Causal Modelling for M/EEG

### 16.1 Introduction

Dynamic Causal Modelling (DCM) is based on an idea initially developed for fMRI data: The measured data are explained by a network model consisting of a few sources, which are interacting dynamically. This network model is inverted using a Bayesian approach, and one can make inferences about connections between sources, or the modulation of connections by task.

For M/EEG data, DCM is a powerful technique for inferring about parameters that one doesn't observe with M/EEG directly. Instead of asking 'How does the strength of the source in left superior temporal gyrus (STG) change between condition A and B?', one can ask questions like 'How does the backward connection from this left STG source to left primary auditory cortex change between condition A and B?'. In other words, one isn't limited to questions about source strength as estimated using a source reconstruction approach, but can test hypotheses about what is happening between sources, in a network.

As M/EEG data is highly resolved in time, as compared to fMRI, the inferences are about more neurobiologically plausible parameters. These relate more directly to the causes of the underlying neuronal dynamics.

The key DCM for M/EEG methods paper appeared in 2006, and the first DCM studies about mismatch negativity came out in 2007/2008. At its heart DCM for M/EEG is a source reconstruction technique, and for the spatial domain we use exactly the same leadfields as other approaches. However, what makes DCM unique, is that it combines the spatial forward model with a biologically informed temporal forward model, describing e.g. the connectivity between sources. This critical ingredient not only makes the source reconstruction more robust by implicitly constraining the spatial parameters, but also allows one to infer about connectivity.

Our methods group is continuing to work on further improvements and extensions to DCM. In the following, we will describe the usage of DCM for evoked responses (both MEG and EEG), DCM for induced responses (i.e., based on power data in the time-frequency domain), and DCM for local field potentials (measured as steady-state responses). All three DCMs share the same interface, as many of the parameters that need to be specified are the same for all three approaches. Therefore, we will first describe DCM for evoked responses, and then point out where the differences to the other two DCMs lie.

This manual provides only a procedural guide for the practical use of DCM for M/EEG. If you want to read more about the scientific background, the algorithms used, or how one would typically use DCM in applications, we recommend the following reading. The two key methods contributions can be found in [23] and [63]. Two other contributions using the model for testing interesting hypotheses about neuronal dynamics are described in [64] and [25]. At the time of writing, there were also three application papers published which demonstrate what kind of hypotheses can be tested with DCM [47, 46, 45]. Another good source of background information

is the recent SPM book [28], where Parts 6 and 7 cover not only DCM for M/EEG but also related research from our group. The DCMs for induced responses and steady-state responses are covered in [18, 17] and [78, 74, 75]. Also note that there is a DCM example file, which we put onto the webpage [http://www.fil.ion.ucl.ac.uk/spm/data/eeg\\_mmn/](http://www.fil.ion.ucl.ac.uk/spm/data/eeg_mmn/). After downloading `DCMexample.mat`, you can load (see below) this file using the DCM GUI, and have a look at the various options, or change some, after reading the description below.

## 16.2 Overview

In summary, the goal of DCM is to explain measured data (such as evoked responses) as the output of an interacting network consisting of a several areas, some of which receive input (i.e., the stimulus). The differences between evoked responses, measured under different conditions, are modelled as a modulation of selected DCM parameters, e.g. cortico-cortical connections [23]. This interpretation of the evoked response makes hypotheses about connectivity directly testable. For example, one can ask, whether the difference between two evoked responses can be explained by top-down modulation of early areas [47]. Importantly, because model inversion is implemented using a Bayesian approach, one can also compute Bayesian model evidences. These can be used to compare alternative, equally plausible, models and decide which is the best [65].

DCM for evoked responses takes the spatial forward model into account. This makes DCM a spatiotemporal model of the full data set (over channels and peri-stimulus time). Alternatively, one can describe DCM also as a spatiotemporal source reconstruction algorithm which uses additional temporal constraints given by neural mass dynamics and long-range effective connectivity. This is achieved by parameterising the lead-field, i.e., the spatial projection of source activity to the sensors. In the current version, this can be done using two different approaches. The first assumes that the leadfield of each source is modelled by a single equivalent current dipole (ECD) [63]. The second approach posits that each source can be presented as a 'patch' of dipoles on the grey matter sheet [22]. This spatial model is complemented by a model of the temporal dynamics of each source. Importantly, these dynamics not only describe how the intrinsic source dynamics evolve over time, but also how a source reacts to external input, coming either from subcortical areas (stimulus), or from other cortical sources.

The GUI allows one to enter all the information necessary for specifying a spatiotemporal model for a given data set. If you want to fit multiple models, we recommend using a batch script. An example of such a script (`DCM_ERP_example`), which can be adapted to your own data, can be found in the `man/example_scripts/` folder of the distribution. You can run this script on example data provided by via the SPM webpage ([http://www.fil.ion.ucl.ac.uk/spm/data/eeg\\_mmn/](http://www.fil.ion.ucl.ac.uk/spm/data/eeg_mmn/)). However, you first have to preprocess these data to produce an evoked response by going through the preprocessing tutorial (chapter 40) or by running the `history_subject1.m` script in the `example_scripts` folder.

## 16.3 Calling DCM for ERP/ERF

After calling `spm eeg`, you see SPM's graphical user interface, the top-left window. The button for calling the DCM-GUI is found in the second partition from the top, on the right hand side. When pressing the button, the GUI pops up. The GUI is partitioned into five parts, going from the top to the bottom. The first part is about loading and saving existing DCMs, and selecting the type of model. The second part is about selecting data, the third is for specification of the spatial forward model, the fourth is for specifying connectivity, and the last row of buttons allows you to estimate parameters and view results.

You have to select the data first and specify the model in a fixed order (data selection > spatial model > connectivity model). This order is necessary, because there are dependencies among the three parts that would be hard to resolve if the input could be entered in any order. At any time, you can switch back and forth from one part to the next. Also, within each part, you can specify information in any order you like.

## 16.4 load, save, select model type

At the top of the GUI, you can load an existing DCM or save the one you are currently working on. In general, you can *save* and *load* during model specification at any time. You can also switch between different DCM analyses (the left menu). The default is 'ERP' which is DCM for evoked responses described here. Currently, the other types are cross-spectral densities (CSD), induced responses (IND) and phase coupling (PHA) described later in this chapter. The menu on the right-hand side lets you choose the neuronal model. 'ERP' is the standard model described in most of our older papers, e.g. [23]. 'SEP' uses a variant of this model, however, the dynamics tend to be faster [71]. 'NMM' is a nonlinear neural mass model based on a first-order approximation, and 'MFM', is also nonlinear and is based on a second-order approximation. 'NMDA' is a variant of the 'NMM' model which also includes a model of NMDA receptor. 'CMC' and 'CMM' are canonical microcircuit models [96] used in the more recent paper to link models of neurophysiological phenomena with canonical models of cortical processing based on the idea of predictive coding.

## 16.5 Data and design

In this part, you select the data and model between-trial effects. The data can be either event-related potentials or fields. These data must be in the SPM-format. On the right-hand side you can enter trial indices of the evoked responses in this SPM-file. For example, if you want to model the second and third evoked response contained within an SPM-file, specify indices 2 and 3. The indices correspond to the order specified by the *condlist* method (see 12). If the two evoked responses, for some reason, are in different files, you have to merge these files first. You can do this with the SPM preprocessing function *merge* (*spm\_eeg\_merge*), see 12. You can also choose how you want to model the experimental effects (i.e. the differences between conditions). For example, if trial 1 is the standard and trial 2 is the deviant response in an oddball paradigm, you can use the standard as the baseline and model the differences in the connections that are necessary to fit the deviant. To do that type 0 1 in the text box below trial indices. Alternatively, if you type -1 1 then the baseline will be the average of the two conditions and the same factor will be subtracted from the baseline connection values to model the standard and added to model the deviant. The latter option is perhaps not optimal for an oddball paradigm but might be suitable for other paradigms where there is no clear 'baseline condition'. When you want to model three or more evoked responses, you can model the modulations of a connection strength of the second and third evoked responses as two separate experimental effects relative to the first evoked response. However, you can also choose to couple the connection strength of the first evoked response with the two gains by imposing a linear relationship on how this connection changes over trials. Then you can specify a single effect (e.g. -1 0 1). This can be useful when one wants to add constraints on how connections (or other DCM parameters) change over trials. A compelling example of this can be found in [45]. For each experimental effect you specify, you will be able to select the connections in the model that are affected by it (see below).

Press the button 'data file' to load the M/EEG dataset. Under 'time window (ms)' you have to enter the peri-stimulus times which you want to model, e.g. 1 to 200 ms.

You can choose whether you want to model the mean or drifts of the data at sensor level. Select 1 for 'detrend' to just model the mean. Otherwise select the number of discrete cosine transform terms you want to use to model low-frequency drifts ( $> 1$ ). In DCM, we use a projection of the data to a subspace to reduce the amount of data. The type of spatial projection is described in [25]. You can select the number of modes you wish to keep. The default is 8.

You can also choose to window your data, along peri-stimulus time, with a hanning window (radio button). This windowing will reduce the influence of the beginning and end of the time-series.

If you are happy with your data selection, the projection and the detrending terms, you can click on the > (forward) button, which will bring you to the next stage *electromagnetic model*. From this part, you can press the red < button to get back to the data and design part.

## 16.6 Electromagnetic model

With the present version of DCM, you have three options for how to spatially model your evoked responses. Either you use a single equivalent current dipole (ECD) for each source, or you use a patch on the cortical surface (IMG), or you don't use a spatial model at all (local field potentials (LFP)). In all three cases, you have to enter the source names (one name in one row). For ECD and IMG, you have to specify the prior source locations (in mm in MNI coordinates). Note that by default DCM uses uninformative priors on dipole orientations, but tight priors on locations. This is because tight priors on locations ensure that the posterior location will not deviate too much from its prior location. This means each dipole stays in its designated area and retains its meaning. The prior location for each dipole can be found either by using available anatomical knowledge or by relying on source reconstructions of comparable studies. Also note that the prior location doesn't need to be overly exact, because the spatial resolution of M/EEG is on a scale of several millimeters. You can also load the prior locations from a file ('load'). You can visualize the locations of all sources when you press 'dipoles'.

The onset-parameter determines when the stimulus, presented at 0 ms peri-stimulus time, is assumed to activate the cortical area to which it is connected. In DCM, we usually do not model the rather small early responses, but start modelling at the first large deflection. Because the propagation of the stimulus impulse through the input nodes causes a delay, we found that the default value of 60 ms onset time is a good value for many evoked responses where the first large deflection is seen around 100 ms. However, this value is a prior, i.e., the inversion routine can adjust it. The prior mean should be chosen according to the specific responses of interest. This is because the time until the first large deflection is dependent on the paradigm or the modality you are working in, e.g. audition or vision. You may also find that changing the onset prior has an effect on how your data are fitted. This is because the onset time has strongly nonlinear effects (a delay) on the data, which might cause differences in which maximum was found at convergence, for different prior values. It is also possible to type several numbers in this box (identical or not) and then there will be several inputs whose timing can be optimized separately. These inputs can be connected to different model sources. This can be useful, for instance, for modelling a paradigm with combined auditory and visual stimulation. The 'duration (sd)' box makes it possible to vary the width of the input volley, separately for each of the inputs. This can be used to model more closely the actual input structure (e.g. a long tone or extended presentation of a visual input). By combining several inputs with different durations one can approximate an even more complex input waveform (e.g. speech).

When you want to proceed to the next model specification stage, hit the > (forward) button and proceed to the *neuronal model*.

## 16.7 Neuronal model

There are five (or more) matrices which you need to specify by button presses. The first three are the connection strength parameters for the first evoked response. There are three types of connections, *forward*, *backward* and *lateral*. In each of these matrices you specify a connection *from* a source area *to* a target area. For example, switching on the element (2, 1) in the intrinsic forward connectivity matrix means that you specify a forward connection from area 1 to 2. Some people find the meaning of each element slightly counter-intuitive, because the column index corresponds to the source area, and the row index to the target area. This convention is motivated by direct correspondence between the matrices of buttons in the GUI and connectivity matrices in DCM equations and should be clear to anyone familiar with matrix multiplication.

The one or more inputs that you specified previously can go to any area and to multiple areas. You can select the receiving areas by selecting area indices in the *C input* vector.

The *B* matrix contains all gain modulations of connection strengths as set in the *A*-matrices. These modulations model the difference between the first and the other modelled evoked responses. For example, for two evoked responses, DCM explains the first response by using the *A*-matrix only. The 2nd response is modelled by modulating these connections by the weights in the *B*-matrix.

## 16.8 Estimation

When you are finished with model specification, you can hit the *estimate* button in the lower left corner. If this is the first estimation and you have not tried any other source reconstructions with this file, DCM will build a spatial forward model. You can use the template head model for quick results. DCM will now estimate model parameters. You can follow the estimation process by observing the model fit in the output window. In the matlab command window, you will see each iteration printed out with expected-maximization iteration number, free energy  $F$ , and the predicted and actual change of  $F$  following each iteration step. At convergence, DCM saves the results in a DCM file, by default named 'DCM\_ERP.mat'. You can save to a different name, eg. if you are estimating multiple models, by pressing 'save' at the top of the GUI and writing to a different name.

## 16.9 Results

After estimation is finished, you can assess the results by choosing from the pull-down menu at the bottom (middle).

With *ERPs (mode)* you can plot, for each mode, the data for both evoked responses, and the model fit.

When you select *ERPs (sources)*, the dynamics of each area are plotted. The activity of the pyramidal cells (which is the reconstructed source activity) are plotted in solid lines, and the activity of the two interneuron populations are plotted as dotted lines.

The option *coupling (A)* will take you to a summary about the posterior distributions of the connections in the  $A$ -matrix. In the upper row, you see the posterior means for all intrinsic connectivities. As above, element  $(i, j)$  corresponds to a connection from area  $j$  to  $i$ . In the lower row, you'll find, for each connection, the probability that its posterior mean is different from the prior mean, taking into account the posterior variance.

With the option *coupling(B)* you can access the posterior means for the gain modulations of the intrinsic connectivities and the probability that they are unequal to the prior means. If you specified several experimental effects, you will be asked which of them you want to look at.

With *coupling(C)* you see a summary of the posterior distribution for the strength of the input into the input receiving area. On the left hand side, DCM plots the posterior means for each area. On the right hand side, you can see the corresponding probabilities.

The option *Input* shows you the estimated input function. As described by [23], this is a gamma function with the addition of low-frequency terms.

With *Response*, you can plot the selected data, i.e. the data, selected by the spatial modes, but back-projected into sensor space.

With *Response (image)*, you see the same as under Results but plotted as an image in grey-scale.

And finally, with the option *Dipoles*, DCM displays an overlay of each dipole on an MRI template using the posterior means of its 3 orientation and 3 location parameters. This makes sense only if you have selected an ECD model under *electromagnetic model*.

Before estimation, when you press the button 'Initialise' you can assign parameter values as initial starting points for the free-energy gradient ascent scheme. These values are taken from another already estimated DCM, which you have to select.

The button *BMS* allows you do Bayesian model comparison of multiple models. It will open the SPM batch tool for model selection. Specify a directory to write the output file to. For the "Inference method" you can choose between "Fixed effects" and "Random effects" (see [91] for additional explanations). Choose "Fixed effects" if you are not sure. Then click on "Data" and in the box below click on "New: Subject". Click on "Subject" and in the box below on "New: Session". Click on models and in the selection window that comes up select the DCM mat files for all the models (remember the order in which you select the files as this is necessary for interpreting the results). Then run the model comparison by pressing the green "Run" button. You will see, at the top, a bar plot of the log-model evidences for all models. At the bottom, you will see the probability, for each model, that it produced the data. By convention, a model can be said to

be the best among a selection of other models, with strong evidence, if its log-model evidence exceeds all other log-model evidences by at least 3.

## 16.10 Cross-spectral densities

### 16.10.1 Model specification

DCM for cross-spectral densities can be applied to M/EEG or intracranial data.

The top panel of the DCM for ERP window allows you to toggle through available analysis methods. On the top left drop-down menu, select 'CSD'. The second drop-down menu in the right of the top-panel allows you to specify whether the analysis should be performed using a model which is linear in the states, for this you can choose ERP or CMC. Alternatively you may use a conductance based model, which is non-linear in the states by choosing, 'NMM', 'MFM' or 'NMDA'. (see [71] for a description of the differences).

The steady state (frequency) response is generated automatically from the time domain recordings. The time duration of the frequency response is entered in the second panel in the time-window. The options for detrending allow you to remove either 1st, 2nd, 3rd or 4th order polynomial drifts from channel data. In the subsampling option you may choose to downsample the data before constructing the frequency response. The number of modes specifies how many components from the leadfield are present in channel data. The specification of between trial effects and design matrix entry is the same as for the case of ERPs, described above.

### 16.10.2 The Lead-Field

The cross-spectral density is a description of the dependencies among the observed outputs of these neuronal sources. To achieve this frequency domain description we must first specify the likely sources and their location. If LFP data are used then only source names are required. This information is added in the third panel by selecting 'LFP'. Alternatively, x,y,z coordinates are specified for ECD or IMG solutions.

### 16.10.3 Connections

The bottom panel then allows you to specify the connections between sources and whether these sources can change from trial type to trial type.

On the first row, three connection types may be specified between the areas. For NMM and MFM options these are Excitatory, Inhibitory or Mixed excitatory and inhibitory connections. When using the ERP option the user will specify if connections are 'Forward', 'Backward' or 'Lateral'. To specify a connection, switch on the particular connection matrix entry. For example to specify an Inhibitory connection from source 3 to source 1, turn on the 'Inhib' entry at position (3,1).

On this row the inputs are also specified. These are where external experimental inputs enter the network.

The matrix on the next row allows the user to select which of the connections specified above can change across trial types. For example in a network of two sources with two mixed connections (1,2) and (2,1), you may wish to allow only one of these to change depending on experimental context. In this case, if you wanted the mixed connection from source 2 to source 1 to change depending on trial type, then select entry (2,1) in this final connection matrix.

### 16.10.4 Cross Spectral Densities

The final selection concerns what frequencies you wish to model. These could be part of a broad frequency range e.g. like the default 4 - 48 Hz, or you could enter a narrow band e.g. 8 to 12 Hz, will model the alpha band in 1Hz increments.

Once you hit the 'invert DCM' option the cross spectral densities are computed automatically (using the spectral-toolbox). The data for inversion includes the auto-spectra and cross-spectra between channels or between channel modes. This is computed using a multivariate autoregressive model, which can accurately measure periodicities in the time-domain data. Overall the spectra



are then presented as an upper-triangular,  $s \times s$  matrix, with auto-spectra on the main diagonal and cross-spectra in the off-diagonal terms.

### 16.10.5 Output and Results

The results menu provides several data estimates. By examining the 'spectral data', you will be able to see observed spectra in the matrix format described above. Selecting 'Cross-spectral density' gives both observed and predicted responses. To examine the connectivity estimates you can select the 'coupling (A)' results option, or for the modulatory parameters, the 'coupling (B)' option. Also you can examine the input strength at each source by selecting the 'coupling (C)' option, as in DCM for ERPs. The option 'trial-specific effects' shows the change in connectivity parameter estimates (from B) from trial to trial relative to the baseline connection (from A). To examine the spectral input to these sources choose the 'Input' option; this should look like a mixture of white and pink noise. Finally the 'dipoles' option allows visualisation of the a posteriori position and orientation of all dipoles in your model.

## 16.11 Induced responses

DCM for induced responses aims to model coupling within and between frequencies that are associated with linear and non-linear mechanisms respectively. The procedure to do this is similar to that for DCM for ERP/ERF. In the following, we will just point out the differences in how to specify models in the GUI. Before using the technique, we recommend reading about the principles behind DCM for induced responses [18].

### 16.11.1 Data

The data to be modelled must be single trial, epoched data. We will model the entire spectra, including both the evoked (phase-locked to the stimulus) and induced (non-phase-locked to the stimulus) components.

### 16.11.2 Electromagnetic model

Currently, DCM for induced responses uses only the ECD method to capture the data features. Note that a difference to DCM for evoked responses is that the parameters of the spatial model are not optimized. This means that DCM for induced responses will project the data into source space using the spatial locations provided by you.

### 16.11.3 Neuronal model

This is where you specify the connection architecture. Note that in DCM for induced responses, the  $A$ -matrix encodes the linear and nonlinear coupling strength between sources.

### 16.11.4 Wavelet transform

This function can be called below the connectivity buttons and allows one to transfer data into the time-frequency domain using a Morlet Wavelet transform as part of the feature extraction. There are two parameters: The frequency window defines the desired frequency band and the wavelet number specifies the temporal-frequency resolution. We recommend values greater than 5 to obtain a stable estimation.

### 16.11.5 Results

#### Frequency modes

This will display the frequency modes, identified using singular value decomposition of spectral dynamics in source space (over time and sources).

**Time-Frequency**

This will display the observed time-frequency power data for all pre-specified sources (upper panel) and the fitted data (lower panel).

**Coupling (A-Hz)**

This will display the coupling matrices representing the coupling strength from source to target frequencies.

**16.12 Phase-coupled responses**

DCM for phase-coupled responses is based on a weakly coupled oscillator model of neuronal interactions.

**16.12.1 Data**

The data to be modeled must be multiple trial, epoched data. Multiple trials are required so that the full state-space of phase differences can be explored. This is achieved with multiple trials as each trial is likely to contain different initial relative phase offsets. Information about different trial types is entered as it is with DCM for ERP ie. using a design matrix. DCM for phase coupling is intended to model dynamic transitions toward synchronization states. As these transitions are short it is advisable to use short time windows of data to model and the higher the frequency of the oscillations you are interested in, the shorter this time window should be. DCM for phase coupling will probably run into memory problems if using long time windows or large numbers of trials.

**16.12.2 Electromagnetic model**

Currently, DCM for phase-coupled responses will work with either ECD or LFP data. Note that a difference to DCM for evoked responses is that the parameters of the spatial model are not optimized. This means that DCM for phase-coupled responses will project the data into source space using the spatial locations you provide.

**16.12.3 Neuronal model**

This is where you specify the connection architecture for the weakly coupled oscillator model. If using the GUI, the Phase Interaction Functions are given by  $a_{ij}\sin(\phi_i - \phi_j)$  where  $a_{ij}$  are the connection weights that appear in the  $A$ -matrix and  $\phi_i$  and  $\phi_j$  are the phases in regions  $i$  and  $j$ . DCM for phase coupling can also be run from a MATLAB script. This provides greater flexibility in that the Phase Interaction Functions can be approximated using arbitrary order Fourier series. Have a look in the `example_scripts` to see how.

**16.12.4 Hilbert transform**

Pressing this button does two things. First, source data are bandpass filtered into the specified range. Second, a Hilbert transform is applied from which time series of phase variables are obtained.

**16.12.5 Results****Sin(Data) - Region i**

This plots the sin of the data (ie. sin of phase variable) and the corresponding model fit for the  $i$ th region.



**Coupling (A),(B)**

This will display the intrinsic and modulatory coupling matrices. The  $i, j$ th entry in  $A$  specifies how quickly region  $i$  changes its phase to align with region  $j$ . The corresponding entry in  $B$  shows how these values are changed by experimental manipulation.



Part V  
Utilities



# Chapter 17

## Display Image

### Contents

---

<b>17.1 Image to Display</b> . . . . .	<b>146</b>
--	------------

---

This is an interactive facility that allows orthogonal sections from an image volume to be displayed. Clicking the cursor on either of the three images moves the point around which the orthogonal sections are viewed. The co-ordinates of the cursor are shown both in voxel co-ordinates and millimetres within some fixed framework. The intensity at that point in the image (sampled using the current interpolation scheme) is also given. The position of the cross-hairs can also be moved by specifying the co-ordinates in millimetres to which they should be moved. Clicking on the horizontal bar above these boxes will move the cursor back to the origin (analogous to setting the cross-hair position (in mm) to [0 0 0]).

The images can be re-oriented by entering appropriate translations, rotations and zooms into the panel on the left. The transformations can then be saved by hitting the "Reorient images..." button. The transformations that were applied to the image are saved to the header information of the selected images. The transformations are considered to be relative to any existing transformations that may be stored. Note that the order that the transformations are applied in is the same as in `spm_matrix.m`.

The "Reset..." button next to it is for setting the orientation of images back to transverse. It retains the current voxel sizes, but sets the origin of the images to be the centre of the volumes and all rotations back to zero.

The right panel shows miscellaneous information about the image. This includes:

Dimensions - the x, y and z dimensions of the image.

Datatype - the computer representation of each voxel.

Intensity - scale-factors and possibly a DC offset.

Miscellaneous other information about the image.

Vox size - the distance (in mm) between the centres of neighbouring voxels.

Origin - the voxel at the origin of the co-ordinate system

DIR Cos - Direction cosines. This is a widely used representation of the orientation of an image.

There are also a few options for different resampling modes, zooms etc. You can also flip between voxel space (as would be displayed by Analyze) or world space (the orientation that SPM considers the image to be in). If you are re-orienting the images, make sure that world space is specified. Blobs (from activation studies) can be superimposed on the images and the intensity windowing can also be changed.

If you have put your images in the correct file format, then (possibly after specifying some rigid-body rotations):

The top-left image is coronal with the top (superior) of the head displayed at the top and the left shown on the left. This is as if the subject is viewed from behind.

The bottom-left image is axial with the front (anterior) of the head at the top and the left shown on the left. This is as if the subject is viewed from above.

The top-right image is sagittal with the front (anterior) of the head at the left and the top of the head shown at the top. This is as if the subject is viewed from the left.

## 17.1 Image to Display

Image to display.

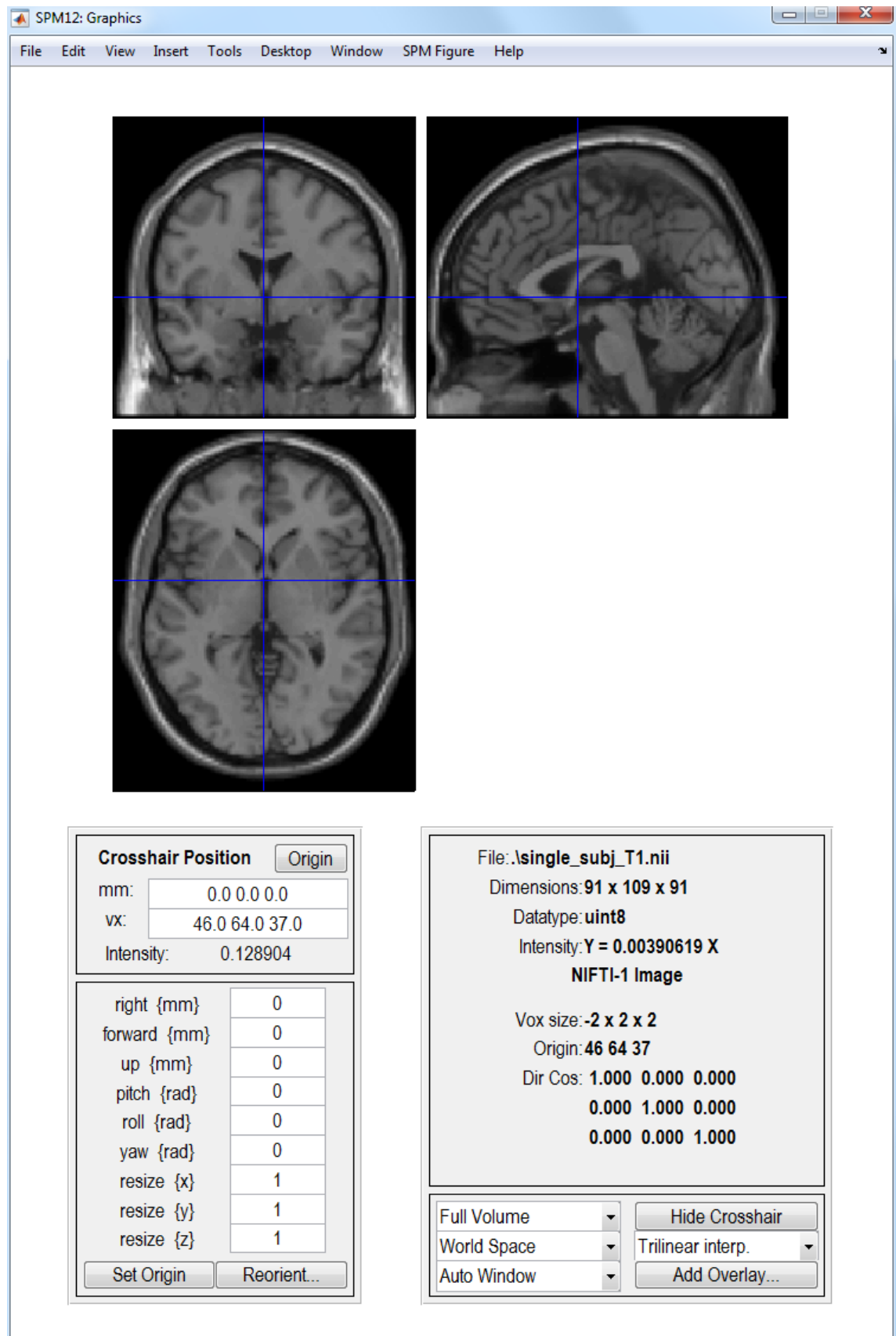


Figure 17.1: The Display routine.





# Chapter 18

## Check Registration

### Contents

---

<b>18.1 Images to Display</b> . . . . .	<b>149</b>
---	------------

---

Orthogonal views of one or more images are displayed. Clicking in any image moves the centre of the orthogonal views. Images are shown in orientations relative to that of the first selected image. The first specified image is shown at the top-left, and the last at the bottom right. The fastest increment is in the left-to-right direction (the same as you are reading this).

If you have put your images in the correct file format, then (possibly after specifying some rigid-body rotations):

The top-left image is coronal with the top (superior) of the head displayed at the top and the left shown on the left. This is as if the subject is viewed from behind.

The bottom-left image is axial with the front (anterior) of the head at the top and the left shown on the left. This is as if the subject is viewed from above.

The top-right image is sagittal with the front (anterior) of the head at the left and the top of the head shown at the top. This is as if the subject is viewed from the left.

### 18.1 Images to Display

Images to display.



# Chapter 19

# Rendering

## Contents

---

<b>19.1 Surface Extraction</b> . . . . .	<b>151</b>
19.1.1 Input Images . . . . .	151
19.1.2 Surfaces . . . . .	151
<b>19.2 Surface Rendering</b> . . . . .	<b>152</b>
19.2.1 Objects . . . . .	152
19.2.2 Lights . . . . .	153

---

This is a toolbox that provides a limited range of surface rendering options. The idea is to first extract surfaces from image data, which are saved in `rend_*.mat` files. These can then be loaded and displayed as surfaces. Note that OpenGL rendering is used, which can be problematic on some computers. The tools are limited - and they do what they do.

## 19.1 Surface Extraction

User-specified algebraic manipulations are performed on a set of images, with the result being used to generate a surface file. The user is prompted to supply images to work on and a number of expressions to evaluate, along with some thresholds. The expression should be a standard matlab expression, within which the images should be referred to as `i1`, `i2`, `i3`,... etc. An isosurface file is created from the results at the user-specified threshold.

### 19.1.1 Input Images

These are the images that are used by the calculator. They are referred to as `i1`, `i2`, `i3`, etc in the order that they are specified.

### 19.1.2 Surfaces

Multiple surfaces can be created from the same image data.

#### Surface

An expression and threshold for each of the surfaces to be generated.

**Expression** Example expressions (f):

\* Mean of six images (select six images)

f = '(i1+i2+i3+i4+i5+i6)/6'

\* Make a binary mask image at threshold of 100

f = 'i1>100'

\* Make a mask from one image and apply to another

f = 'i2.\*(i1>100)'

- here the first image is used to make the mask, which is applied to the second image

\* Sum of n images

f = 'i1 + i2 + i3 + i4 + i5 + ...'

**Surface isovalue(s)** Enter the value at which isosurfaces through the resulting image is to be computed.

## 19.2 Surface Rendering

This utility is for visualising surfaces. Surfaces first need to be extracted and saved in surf\_\*.gii files using the surface extraction routine.

### 19.2.1 Objects

Several surface objects can be displayed together in different colours and with different reflective properties.

#### Object

Each object is a surface (from a surf\_\*.gii file), which may have a number of light-reflecting qualities, such as colour and shininess.

**Surface File** Filename of the surf\_\*.gii file containing the rendering information. This can be generated via the surface extraction routine in SPM. Normally, a surface is extracted from grey and white matter tissue class images, but it is also possible to threshold e.g. an spmT image so that activations can be displayed.

**Color** Specify the colour using a mixture of red, green and blue. For example, white is specified by 1,1,1, black is by 0,0,0 and purple by 1,0,1.

**Red** The intensity of the red colouring (0 to 1).

**Green** The intensity of the green colouring (0 to 1).

**Blue** The intensity of the blue colouring (0 to 1).

**Diffuse Strength** The strength with which the object diffusely reflects light. Mat surfaces reflect light diffusely, whereas shiny surfaces reflect specularly.

**Ambient Strength** The strength with which the object reflects ambient (non-directional) lighting.

**Specular Strength** The strength with which the object specularly reflects light (i.e. how shiny it is). Mat surfaces reflect light diffusely, whereas shiny surfaces reflect specularly.

**Specular Exponent** A parameter describing the specular reflectance behaviour. It relates to the size of the high-lights.

**Specular Color Reflectance** Another parameter describing the specular reflectance behaviour.

**Face Alpha** The opaqueness of the surface. A value of 1 means it is opaque, whereas a value of 0 means it is transparent.

### 19.2.2 Lights

There should be at least one light specified so that the objects can be clearly seen.

#### Light

Specification of a light source in terms of position and colour.

**Position** The position of the light in 3D.

**Color** Specify the colour using a mixture of red, green and blue. For example, white is specified by 1,1,1, black is by 0,0,0 and purple by 1,0,1.

**Red** The intensity of the red colouring (0 to 1).

**Green** The intensity of the green colouring (0 to 1).

**Blue** The intensity of the blue colouring (0 to 1).



# Chapter 20

## Image Calculator

### Contents

---

<b>20.1 Input Images</b>	<b>155</b>
<b>20.2 Output Filename</b>	<b>155</b>
<b>20.3 Output Directory</b>	<b>155</b>
<b>20.4 Expression</b>	<b>156</b>
<b>20.5 Additional Variables</b>	<b>156</b>
20.5.1 Variable	156
<b>20.6 Options</b>	<b>156</b>
20.6.1 Data Matrix	156
20.6.2 Masking	156
20.6.3 Interpolation	156
20.6.4 Data Type	157

---

The image calculator is for performing user-specified algebraic manipulations on a set of images, with the result being written out as an image. The user is prompted to supply images to work on, a filename for the output image, and the expression to evaluate. The expression should be a standard MATLAB expression, within which the images should be referred to as `i1`, `i2`, `i3`,... etc.

### 20.1 Input Images

These are the images that are used by the calculator. They are referred to as `i1`, `i2`, `i3`, etc in the order that they are specified.

### 20.2 Output Filename

The output image is written to current working directory unless a valid full pathname is given. If a path name is given here, the output directory setting will be ignored.

If the field is left empty, i.e. set to "", then the name of the 1st input image, prepended with 'i', is used (change this letter in the `spm_defaults` if necessary).

### 20.3 Output Directory

Files produced by this function will be written into this output directory. If no directory is given, images will be written to current working directory. If both output filename and output directory contain a directory, then output filename takes precedence.

## 20.4 Expression

Example expressions (f):

```
* Mean of six images (select six images)
f = '(i1+i2+i3+i4+i5+i6)/6'
* Make a binary mask image at threshold of 100
f = 'i1>100'
* Make a mask from one image and apply to another
f = 'i2.*(i1>100)'
- here the first image is used to make the mask, which is applied to the second image
* Sum of n images
f = 'i1 + i2 + i3 + i4 + i5 + ...'
* Sum of n images (when reading data into a data-matrix - use dmtx arg)
f = 'sum(X)'
```

## 20.5 Additional Variables

Additional variables which can be used in expression.

### 20.5.1 Variable

Additional variable which can be used in expression.

#### Name

Variable name used in expression.

#### Value

Value of the variable.

## 20.6 Options

Options for image calculator

### 20.6.1 Data Matrix

If the dmtx flag is set, then images are read into a data matrix X (rather than into separate variables i1, i2, i3,...). The data matrix should be referred to as X, and contains images in rows. Computation is plane by plane, so in data-matrix mode, X is a NxK matrix, where N is the number of input images [prod(size(Vi))], and K is the number of voxels per plane [prod(Vi(1).dim(1:2))].

### 20.6.2 Masking

For data types without a representation of NaN, implicit zero masking assumes that all zero voxels are to be treated as missing, and treats them as NaN. NaN's are written as zero (by spm\_write\_plane), for data types without a representation of NaN.

### 20.6.3 Interpolation

With images of different sizes and orientations, the size and orientation of the first is used for the output image. A warning is given in this situation. Images are sampled into this orientation using the interpolation specified by the hold parameter.

The method by which the images are sampled when being written in a different space.

Nearest Neighbour

- Fastest, but not normally recommended.

Trilinear Interpolation



- OK for PET, or realigned fMRI.

Sinc Interpolation

- Better quality (but slower) interpolation, especially with higher degrees.

#### **20.6.4 Data Type**

Data-type of output image



# Chapter 21

## Import

### Contents

---

<b>21.1 DICOM Import</b> .....	<b>159</b>
21.1.1 DICOM files .....	159
21.1.2 Directory structure for converted files .....	159
21.1.3 Output directory .....	160
21.1.4 Protocol name filter .....	160
21.1.5 Conversion options .....	160
<b>21.2 MINC Import</b> .....	<b>160</b>
21.2.1 MINC files .....	160
21.2.2 Options .....	160
<b>21.3 ECAT Import</b> .....	<b>161</b>
21.3.1 ECAT files .....	161
21.3.2 Options .....	161
<b>21.4 PAR/REC Import</b> .....	<b>161</b>
21.4.1 PAR files .....	161
21.4.2 Options .....	161

---

Import.

### 21.1 DICOM Import

DICOM Conversion. Most scanners produce data in DICOM format. This routine attempts to convert DICOM files into SPM compatible image volumes, which are written into the current directory by default. Note that not all flavours of DICOM can be handled, as DICOM is a very complicated format, and some scanner manufacturers use their own fields, which are not in the official documentation at <http://medical.nema.org/>

#### 21.1.1 DICOM files

Select the DICOM files to convert.

#### 21.1.2 Directory structure for converted files

Choose root directory of converted file tree. The options are:

\* Output directory: ./<StudyDate-StudyTime>: Automatically determine the project name and try to convert into the output directory, starting with a StudyDate-StudyTime subdirectory.

This option is useful if automatic project recognition fails and one wants to convert data into a project directory.

\* Output directory: ./<PatientID>: Convert into the output directory, starting with a PatientID subdirectory.

\* Output directory: ./<PatientName>: Convert into the output directory, starting with a PatientName subdirectory.

\* No directory hierarchy: Convert all files into the output directory, without sequence/series subdirectories

### 21.1.3 Output directory

Select a directory where files are written.

### 21.1.4 Protocol name filter

A regular expression to filter protocol names. DICOM images whose protocol names do not match this filter will not be converted.

### 21.1.5 Conversion options

#### Output image format

DICOM conversion can create separate img and hdr files or combine them in one file. The single file option will help you save space on your hard disk, but may be incompatible with programs that are not NIFTI-aware.

In any case, only 3D image files will be produced.

#### Use ICEDims in filename

If image sorting fails, one can try using the additional SIEMENS ICEDims information to create unique filenames. Use this only if there would be multiple volumes with exactly the same file names.

## 21.2 MINC Import

MINC Conversion. MINC is the image data format used for exchanging data within the ICBM community, and the format used by the MNI software tools. It is based on NetCDF. MINC is no longer supported for reading images into SPM, so MINC files need to be converted to NIFTI format in order to use them. See <http://www.bic.mni.mcgill.ca/software/> for more information.

### 21.2.1 MINC files

Select the MINC files to convert.

### 21.2.2 Options

Conversion options

#### Data Type

Data-type of output images. Note that the number of bits used determines the accuracy, and the amount of disk space needed.

#### Output image format

Output files can be written as .img + .hdr, or the two can be combined into a .nii file.

## 21.3 ECAT Import

ECAT 7 Conversion. ECAT 7 is the image data format used by the more recent CTI PET scanners.

### 21.3.1 ECAT files

Select the ECAT files to convert.

### 21.3.2 Options

Conversion options

#### Output image format

Output files can be written as .img + .hdr, or the two can be combined into a .nii file.

## 21.4 PAR/REC Import

Philips PAR/REC Import.

### 21.4.1 PAR files

Select the PAR files to convert.

### 21.4.2 Options

Conversion options

#### Output directory

Select a directory where files are written.

#### Output image format

Output files can be written as .img + .hdr, or the two can be combined into a .nii file.



## Chapter 22

# De-face Images

### Contents

---

<b>22.1 Images to de-face</b> .....	<b>163</b>
-------------------------------------	------------

---

This is a little routine for attempting to strip the face from images, so individuals are more difficult to identify from surface renderings.

De-faced images are prefixed by 'anon\_'.

### 22.1 Images to de-face

Specify the NIfTI images to strip the face from.





# Chapter 23

## Deformations

### Contents

---

<b>23.1 Composition</b> . . . . .	<b>165</b>
23.1.1 Dartel flow . . . . .	165
23.1.2 Deformation Field . . . . .	166
23.1.3 Identity (Reference Image) . . . . .	166
23.1.4 Identity (Bounding Box and Voxel Size) . . . . .	166
23.1.5 Imported <code>_sn.mat</code> . . . . .	167
23.1.6 Inverse . . . . .	167
<b>23.2 Output</b> . . . . .	<b>167</b>
23.2.1 Save Deformation . . . . .	167
23.2.2 Pullback . . . . .	168
23.2.3 Pushforward . . . . .	168
23.2.4 Surface . . . . .	170
23.2.5 Save Jacobian Determinants . . . . .	170

---

This is a utility for working with deformation fields. They can be loaded, inverted, combined etc, and the results either saved to disk, or applied to some image or surface file. This utility was intended for imaging experts and may therefore be a bit difficult for naive users. It provides a great deal of flexibility, which may be confusing to some.

### 23.1 Composition

Deformation fields can be thought of as mappings. These can be combined by the operation of "composition", which is usually denoted by a circle "o". Suppose  $x:A \rightarrow B$  and  $y:B \rightarrow C$  are two mappings, where A, B and C refer to domains in 3 dimensions. Each element a in A points to element  $x(a)$  in B. This in turn points to element  $y(x(a))$  in C, so we have a mapping from A to C. The composition of these mappings is denoted by  $yox:A \rightarrow C$ . Compositions can be combined in an associative way, such that  $zo(yox) = (zoy)ox$ .

In this utility, the right-to-left order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first).

#### 23.1.1 Dartel flow

Imported Dartel flow field.

**Flow field**

The flow field stores the deformation information. The same field can be used for both forward or backward deformations (or even, in principle, half way or exaggerated deformations).

**Forward/Backwards**

The direction of the Dartel flow. Note that a backward transform will warp an individual subject's to match the template (ie maps from template to individual). A forward transform will warp the template image to the individual.

**Time Steps**

The number of time points used for solving the partial differential equations. A single time point would be equivalent to a small deformation model. Smaller values allow faster computations, but are less accurate in terms of inverse consistency and may result in the one-to-one mapping breaking down.

**Dartel Template**

Select the final Template file generated by Dartel. This will be affine registered with a TPM file, such that the resulting spatially normalised images are closer aligned to MNI space. Leave empty if you do not wish to incorporate a transform to MNI space (ie just click "done" on the file selector, without selecting any images).

**23.1.2 Deformation Field**

Deformations can be thought of as vector fields, and represented by three-volume images. In SPM, deformation fields are saved in NIfTI format, with dimensions xdim x ydim x zdim x 1 x 3. Each voxel contains the x, y and z mm coordinates of where the deformation points.

**23.1.3 Identity (Reference Image)**

This option generates an identity transform, but this can be useful for changing the dimensions of the resulting deformation (and any images that are generated from it). Dimensions, orientation etc are derived from an image.

**Image to base Id on**

Specify the image file on which to base the dimensions, orientation etc.

**23.1.4 Identity (Bounding Box and Voxel Size)**

This option generates an identity transform, but this can be useful for changing the dimensions of the resulting deformation (and any images that are generated from it). Dimensions, orientation etc are derived from a specified bounding box and voxel dimensions.

**Voxel sizes**

Specify the voxel sizes of the deformation field to be produced. Non-finite values will default to the voxel sizes of the template image that was originally used to estimate the deformation.

**Bounding box**

Specify the bounding box of the deformation field to be produced. Non-finite values will default to the bounding box of the template image that was originally used to estimate the deformation.

### 23.1.5 Imported `_sn.mat`

Spatial normalisation, and the unified segmentation model of SPM5 save a parameterisation of deformation fields. These consist of a combination of an affine transform, and nonlinear warps that are parameterised by a linear combination of cosine transform basis functions. These are saved in `*_sn.mat` files, which can be converted to deformation fields.

#### Parameter File

Specify the `_sn.mat` to be used.

#### Voxel sizes

Specify the voxel sizes of the deformation field to be produced. Non-finite values will default to the voxel sizes of the template image that was originally used to estimate the deformation.

#### Bounding box

Specify the bounding box of the deformation field to be produced. Non-finite values will default to the bounding box of the template image that was originally used to estimate the deformation.

### 23.1.6 Inverse

Creates the inverse of a deformation field. Deformations are assumed to be one-to-one, in which case they have a unique inverse. If  $y':A \rightarrow B$  is the inverse of  $y:B \rightarrow A$ , then  $y' \circ y = y \circ y' = \text{Id}$ , where  $\text{Id}$  is the identity transform.

Deformations are inverted using the method described in the appendix of:

\* Ashburner J, Andersson JLR & Friston KJ (2000) "Image Registration using a Symmetric Prior - in Three-Dimensions." *Human Brain Mapping* 9(4):212-225

#### Image to base inverse on

Specify the image file on which to base the dimensions, orientation etc.

## 23.2 Output

Various output options are available. The deformation may be saved to disk as a `"y_*.nii"` file. Images may be warped using the resulting deformation, either using a "pullback" procedure, or a "pushforward". The old style of spatial normalisation involved the pullback, whereas the pushforward requires the inverse of the deformation used by the pullback. Finally, the deformation may be used to warp a GIFTI surface file.

### 23.2.1 Save Deformation

The deformation may be saved to disk as a `"y_*.nii"` file.

#### Save as

Save the result as a three-volume image. `"y_"` will be prepended to the filename.

#### Output destination

**Current directory** All created files (deformation fields and warped images) are written to the current directory.

**Output directory** The combined deformation field and the warped images are written into the specified directory.

### 23.2.2 Pullback

This is the old way of warping images, which involves resampling images based on a mapping from the new (warped) image space back to the original image. The deformation should be the inverse of the deformation that would be used for the pushforward procedure.

#### Apply to

Apply the resulting deformation field to some images. The filenames will be prepended by "w".

#### Output destination

**Current directory** All created files (deformation fields and warped images) are written to the current directory.

**Source directories** The combined deformation field is written into the directory of the first deformation field, warped images are written to the same directories as the source images.

**Output directory** The combined deformation field and the warped images are written into the specified directory.

#### Interpolation

The method by which the images are sampled when being written in a different space. (Note that Inf or NaN values are treated as zero, rather than as missing data)

Nearest Neighbour:

- Fastest, but not normally recommended.

Trilinear Interpolation:

- OK for PET, realigned fMRI, or segmentations

B-spline Interpolation:

- Better quality (but slower) interpolation [93], especially with higher degree splines. Can produce values outside the original range (e.g. small negative values from an originally all positive image).

#### Masking

Because of subject motion, different images are likely to have different patterns of zeros from where it was not possible to sample data. With masking enabled, the program searches through the whole time series looking for voxels which need to be sampled from outside the original images. Where this occurs, that voxel is set to zero for the whole set of images (unless the image format can represent NaN, in which case NaNs are used where possible).

#### Gaussian FWHM

Specify the full-width at half maximum (FWHM) of the Gaussian blurring kernel in mm. Three values should be entered, denoting the FWHM in the x, y and z directions.

#### Filename Prefix

The name of the output file(s) will be the name of the input file(s) prefixed with this prefix. Leave empty to use SPM default prefixes.

### 23.2.3 Pushforward

This is a newer way of warping images (for SPM at least), and involves the forward pushing of voxel values from the original image into the appropriate place in the warped image. The deformation field should be the inverse of the one used for the pullback procedure.

“Smoothed” (blurred) spatially normalised images are generated in such a way that the original signal is preserved. Normalised images are generated by a “pushing” rather than a “pulling” (the

usual) procedure. Note that a procedure related to trilinear interpolation is used, and no masking is done. It is therefore recommended that the images are realigned and resliced before they are spatially normalised, in order to benefit from motion correction using higher order interpolation. Alternatively, contrast images generated from unsmoothed native-space fMRI/PET data can be spatially normalised for a 2nd level analysis.

Two “preserve” options are provided. One of them should do the equivalent of generating smoothed “modulated” spatially normalised images. The other does the equivalent of smoothing the modulated normalised fMRI/PET, and dividing by the smoothed Jacobian determinants.

### Apply to

Apply the resulting deformation field to some images. The filenames will be prepended by "w".

### Weight Image

Select an image file to weight the warped data with. This is optional, but the idea is the same as was used by JE Lee et al (2009) in their “A study of diffusion tensor imaging by tissue-specific, smoothing-compensated voxel-based analysis” paper. In principle, a mask of (eg) white matter could be supplied, such that the warped images contain average signal intensities in WM.

### Output destination

**Current directory** All created files (deformation fields and warped images) are written to the current directory.

**Source directories** The combined deformation field is written into the directory of the first deformation field, warped images are written to the same directories as the source images.

**Output directory** The combined deformation field and the warped images are written into the specified directory.

### Field of View

The dimensions and voxel size of the resulting deformation may be defined from some image, or by specifying voxel sizes and a bounding box.

**Image Defined** Use the dimensions, orientation etc of some pre-existing image.

**User Defined** The part of the deformation to use is specified by defining the bounding box and voxel sizes that you would like to have. This is probably stating the obvious to many but smaller voxels and a broader bounding box will take up more disk space, but may give a little more accuracy.

**Bounding box** Specify the bounding box of the deformation field to be produced. Non-finite values will default to the bounding box of the template image that was originally used to estimate the deformation.

**Voxel sizes** Specify the voxel sizes of the deformation field to be produced. Non-finite values will default to the voxel sizes of the template image that was originally used to estimate the deformation.

**Preserve**

**Preserve Concentrations:** Smoothed spatially normalised images (sw\*) represent weighted averages of the signal under the smoothing kernel, approximately preserving the intensities of the original images. This option is currently suggested for eg fMRI.

**Preserve Total:** Smoothed and spatially normalised images preserve the total amount of signal from each region in the images (smw\*). Areas that are expanded during warping are correspondingly reduced in intensity. This option is suggested for VBM.

**Gaussian FWHM**

Specify the full-width at half maximum (FWHM) of the Gaussian blurring kernel in mm. Three values should be entered, denoting the FWHM in the x, y and z directions. Note that you can specify [0 0 0], but any "modulated" data will show aliasing, which occurs because of the way the warped images are generated.

**Filename Prefix**

The name of the output file(s) will be the name of the input file(s) prefixed with this prefix. Leave empty to use SPM default prefixes.

**23.2.4 Surface**

Surfaces may be warped using the resulting deformation. Note that a procedure similar to the pushforward is used, so the deformation should be the inverse of the one that would be used for spatially normalising images via the pullback procedure.

**Surface**

Select a GIFTI file to warp.

**Output destination**

**Current directory** All created files (deformation fields and warped images) are written to the current directory.

**Source directories** The combined deformation field is written into the directory of the first deformation field, warped images are written to the same directories as the source images.

**Output directory** The combined deformation field and the warped images are written into the specified directory.

**23.2.5 Save Jacobian Determinants**

The Jacobian determinants may be saved to disk as a "j\_\*.nii" file.

**Save as**

Save the Jacobian determinants as an image. "j\_" will be prepended to the filename.

**Output destination**

**Current directory** All created files (deformation fields and warped images) are written to the current directory.

**Output directory** The combined deformation field and the warped images are written into the specified directory.

## Part VI

# Tools





# Chapter 24

## FieldMap Toolbox

### 24.1 Introduction

This chapter describes how to use the FieldMap toolbox version 2.1<sup>1</sup> for creating unwrapped field maps that can be used to do geometric distortion correction of EPI images [56, 57, 2]. The methods are based on earlier work by Jezzard et al., [59] and a phase-unwrapping algorithm by Jenkinson [58]. The toolbox can be used via the SPM batch editor or in an interactive mode so that the user can see the effect of applying different field maps and unwarping parameters to EPI images. A voxel displacement map (VDM) is created that can be used with Realign & Unwarp for doing a combined static and dynamic distortion correction or with an Apply VDM function for doing a static distortion correction on a set of realigned images. Realign & Unwarp is designed to work only with images acquired with the phase-encode direction aligned with the anterior-posterior axis. Images acquired with phase-encode directions aligned with other axes can be distortion corrected using the FieldMap toolbox and Apply VDM utility.

### 24.2 Presubtracted Phase and Magnitude Data

Calculate a voxel displacement map (VDM) from presubtracted phase and magnitude field map data (Figure 24.1). This option expects a single magnitude image and a single phase image resulting from the subtraction of two phase images (where the subtraction is usually done automatically by the scanner software). The phase image will be scaled between +/- PI.

#### 24.2.1 Data

Subjects or sessions for which individual field map data has been acquired.

##### Subject

Data for this subject or field map session.

**Phase Image** Select a single phase image. This should be the result from the subtraction of two phase images (where the subtraction is usually done automatically by the scanner software). The phase image will be scaled between +/- PI.

**Magnitude Image** Select a single magnitude image. This is used for masking the phase information and coregistration with the EPI data. If two magnitude images are available, select the one acquired at the shorter echo time because it will have greater signal

**FieldMap defaults** FieldMap default values can be entered as a file or set of values.

---

<sup>1</sup> FieldMap Version 2.0 can be downloaded as part of SPM: <http://www.fil.ion.ucl.ac.uk/spm/software/>  
FieldMap Version 1.1 for SPM2 can be downloaded from <http://www.fil.ion.ucl.ac.uk/spm/toolbox/fieldmap/>

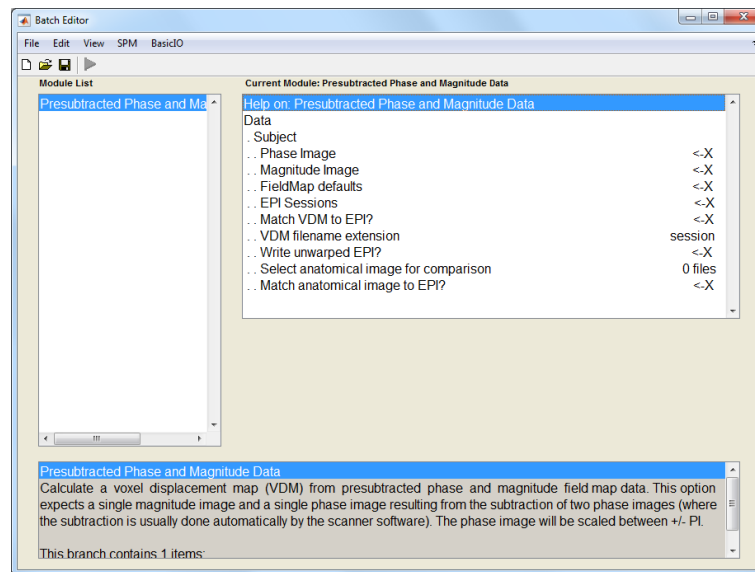


Figure 24.1: FieldMap using the SPM User Interface.

#### Defaults values Defaults values

**Echo times [short TE long TE]** Enter the short and long echo times (in ms) of the data used to acquire the field map.

**Mask brain** Select masking or no masking of the brain. If masking is selected, the magnitude image is used to generate a mask of the brain.

**Blip direction** Enter the blip direction. This is the polarity of the phase-encode blips describing the direction in which k-space is traversed along the y-axis during EPI acquisition with respect to the coordinate system used in SPM. In this coordinate system, the phase encode direction corresponds with the y-direction and is defined as positive from the posterior to the anterior of the head.

The convention used to describe the direction of the k-space traversal is based on the coordinate system used by SPM. In this coordinate system, the phase encode direction corresponds with the y-direction and is defined as positive from the posterior to the anterior of the head. The x-direction is defined as positive from left to right and the z-direction is defined as positive from foot to head. The polarity of the phase-encode blips describes in which direction k-space is traversed along the y-axis with respect to the coordinate system described here.

**Total EPI readout time** Enter the total EPI readout time (in ms). This is the time taken to acquire all of the phase encode steps required to cover k-space (ie one image slice). For example, if the EPI sequence has 64 phase encode steps, the total readout time is the time taken to acquire 64 echoes, e.g. total readout time = number of echos x echo spacing. This time does not include i) the duration of the excitation, ii) the delay between, the excitation and the start of the acquisition or iii) time for fat saturation etc.

**EPI-based field map?** Select non-EPI or EPI based field map. The field map data may be acquired using a non-EPI sequence (typically a gradient echo sequence) or an EPI sequence. The processing will be slightly different for the two cases. If using an EPI-based field map, the resulting Voxel Displacement Map will be inverted since the field map was acquired in distorted space.

**Jacobian modulation?** Select whether or not to use Jacobian modulation. This will adjust the intensities of voxels that have been stretched or compressed but in general is not recommended for EPI distortion correction

**uflags** Different options for phase unwrapping and field map processing

UNWRAPPING METHOD Select method for phase unwrapping

FWHM FWHM of Gaussian filter used to implement weighted smoothing of unwrapped maps.

PAD Size of padding kernel if required.

**WEIGHTED SMOOTHING** Select normal or weighted smoothing.

**mflags** Different options used for the segmentation and creation of the brain mask.

**TEMPLATE IMAGE FOR BRAIN MASKING** Select template file for segmentation to create brain mask

**FWHM** FWHM of Gaussian filter for smoothing brain mask.

**NUMBER OF EROSIONS** Number of erosions used to create brain mask.

**NUMBER OF DILATIONS** Number of dilations used to create brain mask.

**THRESHOLD** Threshold used to create brain mask from segmented data.

**REGULARIZATION** Regularization value used in the segmentation. A larger value helps the segmentation to converge.

**Defaults File** Select the 'pm\_defaults\*.m' file containing the parameters for the field map data. Please make sure that the parameters defined in the defaults file are correct for your field map and EPI sequence. To create your own customised defaults file, either edit the distributed version and/or save it with the name 'pm\_defaults\_yourname.m'.

**EPI Sessions** If a single set of field map data will be used for multiple EPI runs/sessions, select the first EPI in each run/session. A VDM file will be created for each run/session, matched to the first EPI in each run/session and saved with a unique name extension.

**Session** Data for this session.

**Select EPI to Unwarp** Select a single image to distortion correct. The corrected image will be saved with the prefix u. Note that this option is mainly for quality control of correction so that the original and distortion corrected images can be displayed for comparison. To unwarp multiple images please use either Realign & Unwarp or Apply VDM.

**Match VDM to EPI?** Match VDM file to EPI image. This will coregister the field map data to the selected EPI for each run/session.

In general, the field map data should be acquired so that it is as closely registered with the EPI data as possible but matching can be selected if required. If a precalculated field map was loaded then the user is prompted to select a magnitude image in the same space as the field map. If real and imaginary images were selected, the toolbox automatically creates a magnitude image from these images and saves it with the name mag\_NAME-OF-FIRST-INPUT-IMAGE.img.

**Name extension for run/session specific VDM file** This will be the name extension followed by an incremented integer for run/session specific VDM files.

**Write unwarp EPI?** Write out distortion corrected EPI image. The image is saved with the prefix u. Note that this option is mainly for quality control of correction so that the original and distortion corrected images can be displayed for comparison. To unwarp multiple images please use either Realign & Unwarp or Apply VDM.

**Select anatomical image for comparison** Select an anatomical image for comparison with the distortion corrected EPI or leave empty. Note that this option is mainly for quality control of correction.

**Match anatomical image to EPI?** Match the anatomical image to the distortion corrected EPI. Note that this option is mainly for quality control of correction allowing for visual inspection and comparison of the distortion corrected EPI.

## 24.3 Real and Imaginary Data

Calculate a voxel displacement map (VDM) from real and imaginary field map data. This option expects two real and imaginary pairs of data of two different echo times. The phase images will be scaled between +/- PI.

### 24.3.1 Data

Subjects or sessions for which individual field map data has been acquired.

#### Subject

Data for this subject or field map session.

**Short Echo Real Image** Select short echo real image

**Short Echo Imaginary Image** Select short echo imaginary image

**Long Echo Real Image** Select long echo real image

**Long Echo Imaginary Image** Select long echo imaginary image

**Other inputs** As for Presubtracted Phase and Magnitude Data.

## 24.4 Phase and Magnitude Data

Calculate a voxel displacement map (VDM) from double phase and magnitude field map data. This option expects two phase and magnitude pairs of data of two different echo times.

### 24.4.1 Data

Subjects or sessions for which individual field map data has been acquired.

#### Subject

Data for this subject or field map session.

**Short Echo Phase Image** Select short echo phase image

**Short Echo Magnitude Image** Select short echo magnitude image

**Long Echo Phase Image** Select long echo phase image

**Long Echo Magnitude Image** Select long echo magnitude image

**Other inputs** As for Presubtracted Phase and Magnitude Data.

## 24.5 Precalculated FieldMap (in Hz)

Calculate a voxel displacement map (VDM) from a precalculated field map. This option expects a processed field map (ie phase unwrapped, masked if necessary and scaled to Hz). Precalculated field maps can be generated by the FieldMap toolbox and stored as fpm\_\*.files.

### 24.5.1 Data

Subjects or sessions for which individual field map data has been acquired.

#### Subject

Data for this subject or field map session.

**Precalculated field map** Select a precalculated field map. This should be a processed field map (ie phase unwrapped, masked if necessary and scaled to Hz) , for example as generated by the FieldMap toolbox and are stored with `fpm_*` prefix.

**Select magnitude image in same space as fieldmap** Select magnitude image which is in the same space as the field map to do matching to EPI.

**Other inputs** As for Presubtracted Phase and Magnitude Data.

## 24.6 Apply VDM

Apply VDM (voxel displacement map) to resample voxel values in selected image(s). This allows a VDM to be applied to any images which are assumed to be already realigned (e.g. including EPI fMRI time series and DTI data).

The VDM can be been created from a field map acquisition using the FieldMap toolbox and comprises voxel shift values which describe geometric distortions occurring as a result of magnetic susceptibility artefacts. Distortions along any single dimension can be corrected therefore input data may have been acquired with phase encode directions in X, Y (most typical) and Z.

The selected images are assumed to be realigned to the first in the time series (e.g. using Realign: Estimate) but do not need to be resliced. The VDM is assumed to be in alignment with the images selected for resampling (note this can be achieved via the FieldMap toolbox). The resampled images are written to the input subdirectory with the same (prefixed) filename.

e.g. The typical processing steps for fMRI time series would be 1) Realign: Estimate, 2) FieldMap to create VDM, 3) Apply VDM.

Note that this routine is a general alternative to using the VDM in combination with Realign & Unwarp which estimates and corrects for the combined effects of static and movement-related susceptibility induced distortions. Apply VDM can be used when dynamic distortions are not (well) modelled by Realign & Unwarp (e.g. for fMRI data acquired with R->L phase-encoding direction, high field fMRI data or DTI data).

### 24.6.1 Data

Subjects or sessions for which VDM file is being applied to images.

#### Session

Data for this session.

**Images** Select scans for this session. These are assumed to be realigned to the first in the time series (e.g. using Realign: Estimate) but do not need to be resliced

**Fieldmap (vdm\* file)** Select VDM (voxel displacement map) for this session (e.g. created via FieldMap toolbox). This is assumed to be in alignment with the images selected for resampling (note this can be achieved via the FieldMap toolbox).

### 24.6.2 Reslice Options

Apply VDM reslice options

#### Distortion direction

In which direction are the distortions? Any single dimension can be corrected therefore input data may have been acquired with phase encode directions in Y (most typical), X or Z

**Reslice which images ?**

All Images (1..n)

This applies the VDM and reslices all the images.

All Images + Mean Image

This applies the VDM reslices all the images and creates a mean of the resliced images.

**Interpolation**

The method by which the images are sampled when being written in a different space. Nearest Neighbour is fastest, but not recommended for image realignment. Trilinear interpolation is probably OK for PET, but not so suitable for fMRI because higher degree interpolation generally gives better results [93, 94, 95]. Although higher degree methods provide better interpolation, but they are slower because they use more neighbouring voxels.

**Wrapping**

This indicates which directions in the volumes the values should wrap around in. For example, in MRI scans, the images wrap around in the phase encode direction, so (e.g.) the subject's nose may poke into the back of the subject's head. These are typically:

No wrapping - for PET or images that have already been spatially transformed. Also the recommended option if you are not really sure.

Wrap in Y - for (un-resliced) MRI where phase encoding is in the Y direction (voxel space) etc.

**Masking**

Because of subject motion, different images are likely to have different patterns of zeros from where it was not possible to sample data. With masking enabled, the program searches through the whole time series looking for voxels which need to be sampled from outside the original images. Where this occurs, that voxel is set to zero for the whole set of images (unless the image format can represent NaN, in which case NaNs are used where possible).

**Filename Prefix**

Specify the string to be prepended to the filenames of the distortion corrected image file(s). Default prefix is 'u'.

## 24.7 Creating Field Maps Using the FieldMap GUI

The FieldMap Toolbox GUI is shown on the left Figure 24.2. It is divided into two parts. The top part deals with creating the field map in Hz and the bottom part deals with creating the voxel displacement map (VDM) and unwarping the EPI. The toolbox can be used by working through the different inputs in the following order:

### 24.7.1 Create field map in Hz

**Load defaults file**

Select the defaults file from which to load default parameters. If necessary, the parameters used to create the field map can be temporarily modified using the GUI. To change the default parameters, edit `pm_defaults.m` or create a new file called `pm_defaults_NAME.m` (as described in Section 24.2.1).

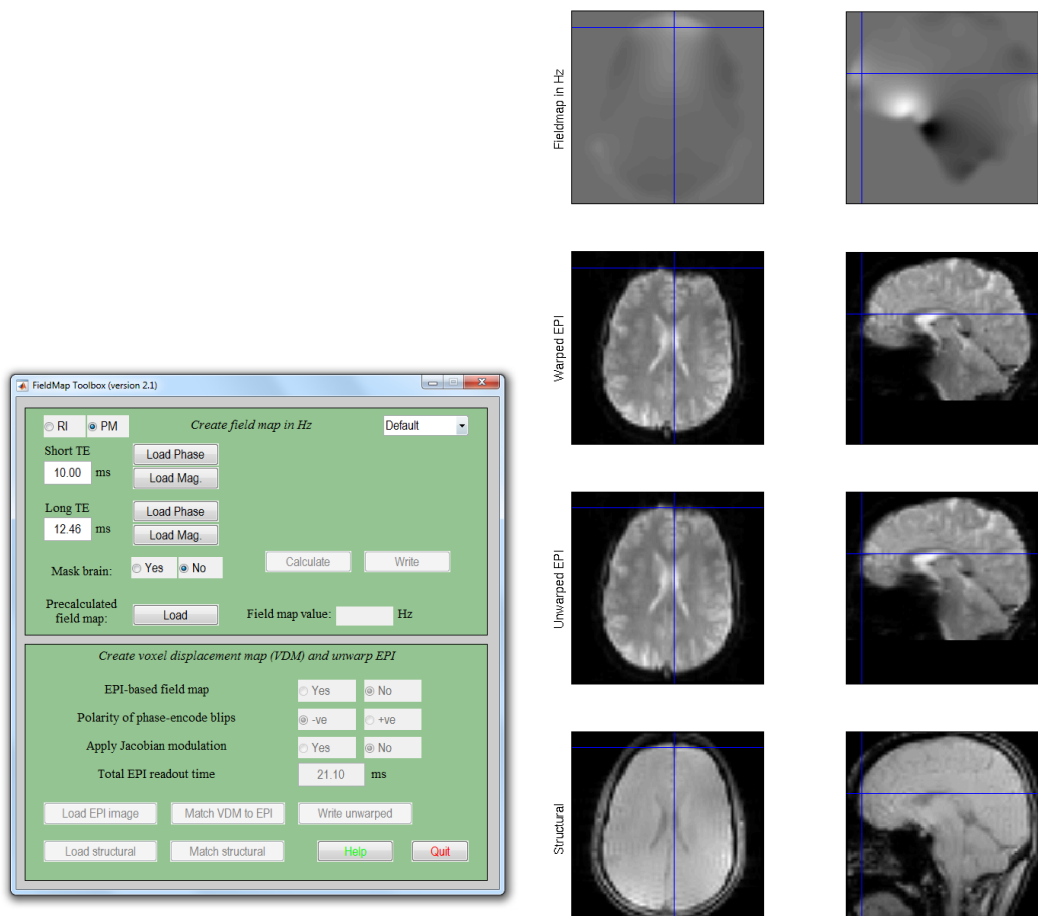


Figure 24.2: FieldMap GUI and Results.

### Data Input Format

**PM** The acquired field map images are in phase and magnitude format. There may be a single pair of phase and magnitude images (i.e. 2 images) in which case the phase image has been created by the vendor sequence from two echo times acquisitions. Alternatively there may be two pairs of phase and magnitude images, one for each echo time (ie 4 images). The units for the phase images MUST BE RADIANS BETWEEN  $+\pi$  and  $-\pi$ . The user will be asked if this is required when the images are selected.

**RI** The acquired field map images are in real and imaginary format. Two pairs of real and imaginary image volumes, one for a shorter and one for a longer echo time (ie 4 images)<sup>2</sup>.

### File Selection

Select NIfTI format images. Generally, the acquired scanner files will be in dicom format which can be correctly converted using the DICOM converter in the corresponding version of SPM. DICOM and other image formats can also be converted to using MRICro<sup>3</sup>.

If the data input format is PM, load Phase and Magnitude images:

1. Single phase image OR phase of short echo-time image.
2. Single magnitude image OR magnitude of short echo-time image.
3. LEAVE EMPTY if input consists of a single phase and magnitude pair OR phase of long echo-time image.
4. LEAVE EMPTY if input consists of a single phase and magnitude pair OR magnitude of long echo-time image.

OR If the data input format is RI, load Real and Magnitude images:

1. Real part of short echo-time image.
2. Imaginary part of short echo-time image.
3. Real part of long echo-time image.
4. Imaginary part of long echo-time image.

### Short TE/Long TE (ms)

Specify the short and long echo times in ms associated with the field map acquisition. Both of these values are required even if a single phase and magnitude image is used as input.

### Mask brain

Specify yes to generate a brain mask using the magnitude data which will be used to exclude regions of the field map outside of the brain.

### Calculate

Calculate an unwrapped field map in Hz which is stored in memory. This represents the map of phase changes associated with the measured field map data. The processing is described in more detail in Section 24.10 and involves some or all of the following steps (as specified in `spm_defaults.m`):

1. Calculation of a Hz fieldmap from input data
2. Segmentation to exclude regions outside of the brain
3. Phase unwrapping

<sup>2</sup> NB If using SPM2, the data input format can only be changed by editing the `spm_defaults.m` file. This is described in Section 24.2.1.

<sup>3</sup>MRICro is freely available from <http://www.cla.sc.edu/psyc/faculty/rorden/mricro.html>.



#### 4. Smoothing and dilation of the processed fieldmap

The processed field map (in Hz) is displayed in the graphics window (top row, right Figure 24.1) and the field at different points can be explored. The field map in Hz is converted to a VDM (voxel displacement map) using the parameters shown in the FieldMap GUI and saved with the filename `vdm5_NAME-OF-FIRST-INPUT-IMAGE.img` in the same directory as the acquired field map images. The VDM file is overwritten whenever the field map is recalculated or when any parameters are changed. The resulting VDM file can be used for unwarping the EPI using `Realign & Unwarp` in SPM (see Section 24.9).

#### Write

Write out the processed field map (in Hz) as a Nifti format image. The image will be saved with the filename `fpm_NAME-OF-FIRST-INPUT-IMAGE.img` in the same directory as the acquired field map images.

#### Load Pre-calculated

Load a precalculated unwrapped field map (`fpm_.img`). This should be a single image volume with units of Hz in Nifti format. The precalculated field map may have been created previously using the FieldMap toolbox or by other means. Once loaded, the field map is displayed in the graphics window (top row, right, Figure 24.1) and the field at different points can be explored.

#### Field map value (Hz)

Interrogate the value of the field map in Hz at the location specified by the mouse pointer in the graphics window.

### 24.7.2 Create voxel displacement map (VDM) and unwarp EPI

When any of the parameters below are changed, a new VDM is created and written out as `vdm5_NAME-OF-FIRST-INPUT-IMAGE.img`. The `vdm5_NAME-OF-FIRST-INPUT-IMAGE.mat` file is not updated unless 'Match VDM to EPI' is selected as described in Section 24.7.2.

#### EPI-based field map - Yes/No

Select Yes if the field map is based on EPI data or No otherwise. Most scanner vendor field map sequences are non-EPI.

#### Polarity of phase-encode blips - +ve/-ve

Select +ve or -ve blip direction. When images are acquired K-space can be traversed using positive or negative phase-encode blips. This direction will influence the geometric distortions in terms of whether the affected regions of the image are stretched or compressed.

The convention used to describe the direction of the k-space traversal is based on the coordinate system used by SPM. In this coordinate system, the phase encode direction corresponds with the y-direction and is defined as positive from the posterior to the anterior of the head. The x-direction is defined as positive from left to right and the z-direction is defined as positive from foot to head. The polarity of the phase-encode blips describes in which direction k-space is traversed along the y-axis with respect to the coordinate system described here.

#### Apply Jacobian modulation - Yes/No

Select Yes to do Jacobian Modulation to adjust the intensities of voxels that have been stretched or compressed. In general this is not recommended for unwarping EPI data at this stage.

**Total EPI readout time (ms)**

Enter the total time in ms for the readout of the EPI echo train which is typically 10s of ms. This is the time taken to acquire all of the phase encode steps required to cover k-space (ie one image slice). For example, if the EPI sequence has 64 phase encode steps, the total readout time is the time taken to acquire 64 echoes: total readout time = number of echoes  $\times$  echo spacing. This time does not include i) the duration of the excitation, ii) the delay between the excitation and the start of the acquisition or iii) time for fat saturation.

**Load EPI image**

Select a sample EPI image in NIFTI format. This image is automatically unwrapped using the VDM calculated with the current parameters. The warped and the unwrapped image are displayed in the graphics window underneath the field map (middle rows, right, Figure 24.1).

**Match VDM to EPI**

Select this option to match the field map magnitude data to the EPI image before it is used to unwarp the EPI. In general, the field map data should be acquired so that it is as closely registered with the EPI data as possible but matching can be selected if required. If a precalculated field map was loaded then the user is prompted to select a magnitude image in the same space as the field map. If real and imaginary images were selected, the toolbox automatically creates a magnitude image from these images and saves it with the name mag\_NAME-OF-FIRST-INPUT-IMAGE.img.

**Write unwrapped**

Write unwrapped EPI image with the filename `uNAME_OF_EPI.img`.

**Load structural**

Load a structural image for comparison with unwrapped EPI. This is displayed in the graphics window below the other images (bottom row, right fig 1).

**MatchStructural**

Coregister the structural image to the unwrapped EPI and write the resulting transformation matrix to the header of the selected structural image.

**Help**

Call `spm_help` to display `FieldMap.man`.

**Quit**

Quit the toolbox and closes all windows associated with it.

## 24.8 Using the FieldMap in Batch scripts

`FieldMap_preprocess.m` which calls `FieldMap_create.m` gives an example of how to run the FieldMap toolbox without using the GUI. To run the script, make sure your MATLAB path includes the directory where the FieldMap toolbox is installed. This can be done using the Set Path option under File in the MATLAB windows manager or using the command:

```
addpath /whatever/spm/toolbox/FieldMap
```

To run the FieldMap batch script, in MATLAB enter the following command:

```
VDM = FieldMap_preprocess(fm_dir,epi_dir, [te1, te2, epifm, tert, kdir, mask, match] );
```

where

*fm\_dir* - name of directory containing fieldmap images. (e.g. *fm\_dir* = '/path/study1/subj1/fieldmap')

*epi\_dir* - name of directory containing epi images. (e.g. *epi\_dir* = '/path/study1/subj1/images')

*te1* - short echo time (in ms)

*te2* - long echo time (in ms)

*epifm* - epi-based fieldmap - yes or no (1/0)

*tert* - total echo readout time (in ms)

*kdir* - blip direction (1/-1)

*mask* do brain segmentation to mask field map (1/0)

*match* match vdm file to first EPI in run (1/0).

NB: FieldMap will match the field map to the first epi image in the time series (after removing the dummy scans). Therefore, *epi\_dir* must be the directory that contains the epi run that all other images will be realigned to.

The script will create an *fpm\** file, a *vdm5\_\** file and an unwrapped version of the EPI saved with the prescript "u".

## 24.9 Using the VDM file with Unwarp

In SPM, select the Realign & Unwarp option. For the input data called Phase map (*vdm\** file), select the *vdm5\_\** or *vdm5\_\** file for the subject and/or session. If you acquired more than one session (or run) of EPI images, you need to select a different *vdm5\_\** file for each one. For more information about Unwarp see <http://www.fil.ion.ucl.ac.uk/spm/toolbox/unwarp>.

## 24.10 Appendices

### 24.10.1 Processing Hz field maps

Processing field maps involves a series of steps for which certain parameters in the *spm\_defaults* file must be set.

1. If the acquired field map data comprises two complex images, the phase difference between them is calculated.
2. The phase map is unwrapped using the method specified by *spm\_def.UNWRAPPING\_METHOD* = 'Mark3D' or 'Mark2D' or 'Huttonish'. For a description of these different methods see *spm\_unwrap.m* or *FieldMap\_principles.man*. The default option is 'Mark3D'.
3. A mask is created so that unwrapping only occurs in regions where there is signal. If necessary, this mask can be expanded so that any voxel that hasn't been unwrapped and is less than *spm\_def.PAD/2* voxels away from an unwrapped one will be replaced by an average of the surrounding unwrapped voxels. This can be done by setting the parameter *spm\_def.PAD* to a value greater than 0. The default value is 0 but a value > 0 (eg 10) may be necessary if normal smoothing is chosen instead of weighted smoothing (as explained in the next step).
4. If required a mask can be generated to exclude regions of the fieldmap outside of the brain (in addition to the unwrapping mask described above). This step uses SPM segmentation for which the parameters in *spm\_def.MFLAGS* can be set. For example, if the segmentation fails, (maybe because the fieldmap magnitude image doesn't have enough contrast), *spm\_def.MFLAGS.REG* can be increased to say 0.05). The other parameters control morphological operations to generate a smooth brain mask and have been set empirically.
5. The unwrapped phase map is scaled by  $1/(2*PI*difference\ in\ echo\ time)$  to convert it to Hz.
6. A weighted gaussian smoothing (weighted by the inverse of the noise) is performed on the unwrapped phase-map if the parameter *spm\_def.WS* = 1. If *spm\_def.WS* = 0, a normal smoothing is done. The weighted smoothing is particularly slow on large data sets ie high

resolution. If field maps are acquired at high resolution then it is recommended to use `spm_def.WS = 0` and do some padding of the intensity mask eg `spm_def.PAD = 10`. The size of the Gaussian filter used to implement either weighted or normal smoothing of the unwrapped maps is usually set to `spm_def.FWHM = 10`.

### 24.10.2 Converting Hz field map to VDM

1. The field map in Hz is multiplied by the total EPI readout time (in ms, ) of the EPI image to be unwrapped, resulting in a VDM. The readout time is specified by `spm_def.TOTAL_EPI_READOUT_T` (eg typically 10s of ms).The total EPI readout time is the time taken to acquire all of the phase encode steps required to cover k-space (ie one image slice). For example, if the EPI sequence has 64 phase encode steps, the total readout time is the time taken to acquire 64 echoes, e.g. total readout time = number of echoes  $\times$  echo spacing. This time does not include i) the duration of of the excitation, ii) the delay between the excitation and the start of the acquisition or iii) time for fat saturation etc.
2. The VDM is multiplied by  $\pm 1$  to indicate whether the K-space traversal for the data acquisition has a +ve or -ve blip direction. This will ensure that the unwarping is performed in the correct direction and is specified by `spm_def.K_SPACE_TRAVERSAL_BLIPI_DIR =  $\pm 1$` .
3. The toolbox must know if the field map is based on an EPI or non-EPI acquisition. If using an EPI-based field map, the VDM must be inverted since the field map was acquired in warped space. This is specified by `spm_def.EPI_BASED_FIELDMAPS = 1` or 0.
4. Jacobian Modulation can be applied to the unwrapped EPI image. This modulates the intensity of the unwrapped image so that in regions where voxels were compressed, the intensity is decreased and where voxels were stretched, the intensities are increased slightly. The modulation involves multiplying the unwrapped EPI by  $1 +$  the 1-d derivative of the VDM in the phase direction. An intensity adjustment of this nature may improve the coregistration results between an unwrapped EPI and an undistorted image. This is specified by `spm_def.DO_JACOBIAN_MODULATION = 0` or 1.
5. When any of the above conversion parameters are changed or a new EPI is selected, a new VDM is created and saved with the filename `vdm5_NAME-OF-FIRST-INPUT-IMAGE.img`. Any previous copy of the .img file is overwritten, but the corresponding .mat file is retained. It is done this way because the VDM may have already been coregistered to the EPI (as described below). Then, for an EPI-based VDM, the match between the VDM and the EPI will still be valid even if any of the above parameters have been changed. If the VDM is non-EPI-based and any of the above parameters are changed, the match between the VDM and the EPI may no longer be valid. In this case a warning is given to the user that it may be necessary to perform the coregistration again.

### 24.10.3 Matching field map data to EPI data

1. If required, the fieldmap can be matched to the EPI. This is done slightly differently depending on whether the field map is based on EPI or non-EPI data. If using an EPI field map, the magnitude image is coregistered to the EPI. The resulting transformation matrix is used to sample the VDM file in the space of the EPI before unwarping.
2. If using a non-EPI field map, the VDM is used to forward warp the magnitude image which is then coregistered to the EPI. The forward warped image is saved with the filename `wfmag_NAME-OF-FIRST-INPUT-IMAGE.img`.

# Chapter 25

## Dartel Tools

### Contents

---

<b>25.1 Initial Import</b> . . . . .	<b>186</b>
25.1.1 Parameter Files . . . . .	186
25.1.2 Output Directory . . . . .	187
25.1.3 Bounding box . . . . .	187
25.1.4 Voxel size . . . . .	187
25.1.5 Image option . . . . .	187
25.1.6 Grey Matter . . . . .	187
25.1.7 White Matter . . . . .	187
25.1.8 CSF . . . . .	187
<b>25.2 Run Dartel (create Templates)</b> . . . . .	<b>187</b>
25.2.1 Images . . . . .	187
25.2.2 Settings . . . . .	187
<b>25.3 Run Dartel (existing Templates)</b> . . . . .	<b>189</b>
25.3.1 Images . . . . .	189
25.3.2 Settings . . . . .	189
<b>25.4 Normalise to MNI Space</b> . . . . .	<b>190</b>
25.4.1 Dartel Template . . . . .	190
25.4.2 Select according to . . . . .	190
25.4.3 Voxel sizes . . . . .	191
25.4.4 Bounding box . . . . .	191
25.4.5 Preserve . . . . .	191
25.4.6 Gaussian FWHM . . . . .	191
<b>25.5 Create Warped</b> . . . . .	<b>191</b>
25.5.1 Flow fields . . . . .	192
25.5.2 Images . . . . .	192
25.5.3 Modulation . . . . .	192
25.5.4 Time Steps . . . . .	192
25.5.5 Interpolation . . . . .	192
<b>25.6 Jacobian determinants</b> . . . . .	<b>192</b>
25.6.1 Flow fields . . . . .	192
25.6.2 Time Steps . . . . .	192
<b>25.7 Create Inverse Warped</b> . . . . .	<b>193</b>
25.7.1 Flow fields . . . . .	193
25.7.2 Images . . . . .	193
25.7.3 Time Steps . . . . .	193
25.7.4 Interpolation . . . . .	193
<b>25.8 Population to ICBM Registration</b> . . . . .	<b>193</b>

25.8.1 Dartel Template . . . . .	193
<b>25.9 Kernel Utilities . . . . .</b>	<b>193</b>
25.9.1 Kernel from Images . . . . .	194
25.9.2 Kernel from Flows . . . . .	194

---

This toolbox is based around the “A Fast Diffeomorphic Registration Algorithm” paper [3]. The idea is to register images by computing a “flow field”, which can then be “exponentiated” to generate both forward and backward deformations. Currently, the software only works with images that have isotropic voxels, identical dimensions and which are in approximate alignment with each other. One of the reasons for this is that the approach assumes circulant boundary conditions, which makes modelling global rotations impossible. Another reason why the images should be approximately aligned is because there are interactions among the transformations that are minimised by beginning with images that are already almost in register. This problem could be alleviated by a time varying flow field, but this is currently computationally impractical.

Because of these limitations, images should first be imported. This involves taking the “\*\_seg\_sn.mat” files produced by the segmentation code of SPM5, and writing out rigidly transformed versions of the tissue class images, such that they are in as close alignment as possible with the tissue probability maps. Rigidly transformed original images can also be generated, with the option to have skull-stripped versions.

The next step is the registration itself. This can involve matching single images together, or it can involve the simultaneous registration of e.g. GM with GM, WM with WM and 1-(GM+WM) with 1-(GM+WM) (when needed, the 1-(GM+WM) class is generated implicitly, so there is no need to include this class yourself). This procedure begins by creating a mean of all the images, which is used as an initial template. Deformations from this template to each of the individual images are computed, and the template is then re-generated by applying the inverses of the deformations to the images and averaging. This procedure is repeated a number of times.

Finally, warped versions of the images (or other images that are in alignment with them) can be generated.

This toolbox is not yet seamlessly integrated into the SPM package. Eventually, the plan is to use many of the ideas here as the default strategy for spatial normalisation. The toolbox may change with future updates. There will also be a number of other (as yet unspecified) extensions, which may include a variable velocity version (related to LDDMM). Note that the Fast Diffeomorphism paper only describes a sum of squares objective function. The multinomial objective function is an extension, based on a more appropriate model for aligning binary data to a template.

## 25.1 Initial Import

Images first need to be imported into a form that Dartel can work with. If the default segmentation is used (ie the Segment button), then this involves taking the results of the segmentation (\*\_seg\_sn.mat) [8], in order to have rigidly aligned tissue class images. Typically, there would be imported grey matter and white matter images, but CSF images can also be included. The subsequent Dartel alignment will then attempt to nonlinearly register these tissue class images together. If the new segmentation routine is used (from the toolbox), then this includes the option to generate “imported” tissue class images. This means that a separate importing step is not needed for it.

### 25.1.1 Parameter Files

Select ‘\*\_sn.mat’ files containing the spatial transformation and segmentation parameters. Rigidly aligned versions of the image that was segmented will be generated. The image files used by the segmentation may have moved. If they have, then (so the import can find them) ensure that they are either in the output directory, or the current working directory.

### 25.1.2 Output Directory

Select the directory where the resliced files should be written.

### 25.1.3 Bounding box

The bounding box (in mm) of the volume that is to be written (relative to the anterior commissure). Non-finite values will be replaced by the bounding box of the tissue probability maps used in the segmentation.

### 25.1.4 Voxel size

The (isotropic) voxel sizes of the written images. A non-finite value will be replaced by the average voxel size of the tissue probability maps used by the segmentation.

### 25.1.5 Image option

A resliced version of the original image can be produced, which may have various procedures applied to it. All options will rescale the images so that the mean of the white matter intensity is set to one. The “skull stripped” versions are the images simply scaled by the sum of the grey and white matter probabilities.

### 25.1.6 Grey Matter

Produce a resliced version of this tissue class?

### 25.1.7 White Matter

Produce a resliced version of this tissue class?

### 25.1.8 CSF

Produce a resliced version of this tissue class?

## 25.2 Run Dartel (create Templates)

Run the Dartel nonlinear image registration procedure. This involves iteratively matching all the selected images to a template generated from their own mean. A series of Template\*.nii files are generated, which become increasingly crisp as the registration proceeds.

### 25.2.1 Images

Select the images to be warped together. Multiple sets of images can be simultaneously registered. For example, the first set may be a bunch of grey matter images, and the second set may be the white matter images of the same subjects.

#### Images

Select a set of imported images of the same type to be registered by minimising a measure of difference from the template.

### 25.2.2 Settings

Various settings for the optimisation. The default values should work reasonably well for aligning tissue class images together.

**Template basename**

Enter the base for the template name. Templates generated at each outer iteration of the procedure will be `basename_1.nii`, `basename_2.nii` etc. If empty, then no template will be saved. Similarly, the estimated flow-fields will have the `basename` appended to them.

**Regularisation Form**

The registration is penalised by some “energy” term. Here, the form of this energy term is specified. Three different forms of regularisation can currently be used.

**Outer Iterations**

The images are averaged, and each individual image is warped to match this average. This is repeated a number of times.

**Outer Iteration** Different parameters can be specified for each outer iteration. Each of them warps the images to the template, and then regenerates the template from the average of the warped images. Multiple outer iterations should be used for more accurate results, beginning with a more coarse registration (more regularisation) then ending with the more detailed registration (less regularisation).

**Inner Iterations** The number of Gauss-Newton iterations to be done within this outer iteration. After this, new average(s) are created, which the individual images are warped to match.

**Reg params** For linear elasticity, the parameters are `mu`, `lambda` and `id`. For membrane energy, the parameters are `lambda`, `unused` and `id`. `id` is a term for penalising absolute displacements, and should therefore be small. For bending energy, the parameters are `lambda`, `id1` and `id2`, and the regularisation is by  $(-\text{lambda} * \text{Laplacian} + \text{id1})^2 + \text{id2}$ .

Use more regularisation for the early iterations so that the deformations are smooth, and then use less for the later ones so that the details can be better matched.

**Time Steps** The number of time points used for solving the partial differential equations. A single time point would be equivalent to a small deformation model. Smaller values allow faster computations, but are less accurate in terms of inverse consistency and may result in the one-to-one mapping breaking down. Earlier iteration could use fewer time points, but later ones should use about 64 (or fewer if the deformations are very smooth).

**Smoothing Parameter** A LogOdds parameterisation of the template is smoothed using a multi-grid scheme. The amount of smoothing is determined by this parameter.

**Optimisation Settings**

Settings for the optimisation. If you are unsure about them, then leave them at the default values. Optimisation is by repeating a number of Levenberg-Marquardt iterations, in which the equations are solved using a full multi-grid (FMG) scheme. FMG and Levenberg-Marquardt are both described in Numerical Recipes (2nd edition).

**LM Regularisation** Levenberg-Marquardt regularisation. Larger values increase the stability of the optimisation, but slow it down. A value of zero results in a Gauss-Newton strategy, but this is not recommended as it may result in instabilities in the FMG.

**Cycles** Number of cycles used by the full multi-grid matrix solver. More cycles result in higher accuracy, but slow down the algorithm. See Numerical Recipes for more information on multi-grid methods.



**Iterations** Number of relaxation iterations performed in each multi-grid cycle. More iterations are needed if using “bending energy” regularisation, because the relaxation scheme only runs very slowly. See the chapter on solving partial differential equations in Numerical Recipes for more information about relaxation methods.

## 25.3 Run Dartel (existing Templates)

Run the Dartel nonlinear image registration procedure to match individual images to pre-existing template data. Start out with smooth templates, and select crisp templates for the later iterations.

### 25.3.1 Images

Select the images to be warped together. Multiple sets of images can be simultaneously registered. For example, the first set may be a bunch of grey matter images, and the second set may be the white matter images of the same subjects.

#### Images

Select a set of imported images of the same type to be registered by minimising a measure of difference from the template.

### 25.3.2 Settings

Various settings for the optimisation. The default values should work reasonably well for aligning tissue class images together.

#### Regularisation Form

The registration is penalised by some “energy” term. Here, the form of this energy term is specified. Three different forms of regularisation can currently be used.

#### Outer Iterations

The images are warped to match a sequence of templates. Early iterations should ideally use smoother templates and more regularisation than later iterations.

**Outer Iteration** Different parameters and templates can be specified for each outer iteration.

**Inner Iterations** The number of Gauss-Newton iterations to be done within this outer iteration.

**Reg params** For linear elasticity, the parameters are mu, lambda and id. For membrane energy, the parameters are lambda, unused and id.id is a term for penalising absolute displacements, and should therefore be small. For bending energy, the parameters are lambda, id1 and id2, and the regularisation is by  $(-\lambda * \text{Laplacian} + \text{id1})^2 + \text{id2}$ .

Use more regularisation for the early iterations so that the deformations are smooth, and then use less for the later ones so that the details can be better matched.

**Time Steps** The number of time points used for solving the partial differential equations. A single time point would be equivalent to a small deformation model. Smaller values allow faster computations, but are less accurate in terms of inverse consistency and may result in the one-to-one mapping breaking down. Earlier iteration could use fewer time points, but later ones should use about 64 (or fewer if the deformations are very smooth).

**Template** Select template. Smoother templates should be used for the early iterations. Note that the template should be a 4D file, with the 4th dimension equal to the number of sets of images.

### Optimisation Settings

Settings for the optimisation. If you are unsure about them, then leave them at the default values. Optimisation is by repeating a number of Levenberg-Marquardt iterations, in which the equations are solved using a full multi-grid (FMG) scheme. FMG and Levenberg-Marquardt are both described in Numerical Recipes (2nd edition).

**LM Regularisation** Levenberg-Marquardt regularisation. Larger values increase the the stability of the optimisation, but slow it down. A value of zero results in a Gauss-Newton strategy, but this is not recommended as it may result in instabilities in the FMG.

**Cycles** Number of cycles used by the full multi-grid matrix solver. More cycles result in higher accuracy, but slow down the algorithm. See Numerical Recipes for more information on multi-grid methods.

**Iterations** Number of relaxation iterations performed in each multi-grid cycle. More iterations are needed if using “bending energy” regularisation, because the relaxation scheme only runs very slowly. See the chapter on solving partial differential equations in Numerical Recipes for more information about relaxation methods.

## 25.4 Normalise to MNI Space

Normally, Dartel generates warped images that align with the average-shaped template. This routine includes an initial affine registration of the template (the final one generated by Dartel), with the TPM data released with SPM.

“Smoothed” (blurred) spatially normalised images are generated in such a way that the original signal is preserved. Normalised images are generated by a “pushing” rather than a “pulling” (the usual) procedure. Note that a procedure related to trilinear interpolation is used, and no masking is done. It is therefore recommended that the images are realigned and resliced before they are spatially normalised, in order to benefit from motion correction using higher order interpolation. Alternatively, contrast images generated from unsmoothed native-space fMRI/PET data can be spatially normalised for a 2nd level analysis.

Two “preserve” options are provided. One of them should do the equivalent of generating smoothed “modulated” spatially normalised images. The other does the equivalent of smoothing the modulated normalised fMRI/PET, and dividing by the smoothed Jacobian determinants.

### 25.4.1 Dartel Template

Select the final Template file generated by Dartel. This will be affine registered with a TPM file, such that the resulting spatially normalised images are closer aligned to MNI space. Leave empty if you do not wish to incorporate a transform to MNI space (ie just click “done’ on the file selector, without selecting any images).

### 25.4.2 Select according to

You may wish to spatially normalise only a few subjects, but have many scans per subject (eg for fMRI), or you may have lots of subjects, but with a small and fixed number of scans for each of them (eg for VBM). The idea is to chose the way of selecting files that is easier.

#### Few Subjects

Select this option if there are only a few subjects, each with many or a variable number of scans each. You will then need to specify a series of subjects, and the flow field and images of each of them.

**Subject** Subject to be spatially normalized.

**Flow Field** Dartel flow field for this subject.

**Images** Images for this subject to spatially normalise.

### Many Subjects

Select this option if you have many subjects to spatially normalise, but there are a small and fixed number of scans for each subject.

**Flow fields** The flow fields store the deformation information. The same fields can be used for both forward or backward deformations (or even, in principle, half way or exaggerated deformations).

**Images** The flow field deformations can be applied to multiple images. At this point, you are choosing how many images each flow field should be applied to.

**Images** Select images to be warped. Note that there should be the same number of images as there are flow fields, such that each flow field warps one image.

#### 25.4.3 Voxel sizes

Specify the voxel sizes of the deformation field to be produced. Non-finite values will default to the voxel sizes of the template image that was originally used to estimate the deformation.

#### 25.4.4 Bounding box

Specify the bounding box of the deformation field to be produced. Non-finite values will default to the bounding box of the template image that was originally used to estimate the deformation.

#### 25.4.5 Preserve

Preserve Concentrations (no "modulation"): Smoothed spatially normalised images (sw\*) represent weighted averages of the signal under the smoothing kernel, approximately preserving the intensities of the original images. This option is currently suggested for eg fMRI.

Preserve Amount ("modulation"): Smoothed and spatially normalised images preserve the total amount of signal from each region in the images (smw\*). Areas that are expanded during warping are correspondingly reduced in intensity. This option is suggested for VBM.

#### 25.4.6 Gaussian FWHM

Specify the full-width at half maximum (FWHM) of the Gaussian blurring kernel in mm. Three values should be entered, denoting the FWHM in the x, y and z directions. Note that you can also specify [0 0 0], but any "modulated" data will show aliasing (see eg Wikipedia), which occurs because of the way the warped images are generated.

## 25.5 Create Warped

This allows spatially normalised images to be generated. Note that voxel sizes and bounding boxes can not be adjusted, and that there may be strange effects due to the boundary conditions used by the warping. Also note that the warped images are not in Talairach or MNI space. The coordinate system is that of the average shape and size of the subjects to which Dartel was applied. In order to have MNI-space normalised images, then the Deformations Utility can be used to compose the individual Dartel warps, with a deformation field that matches (e.g.) the Template grey matter generated by Dartel, with one of the grey matter volumes released with SPM.

### 25.5.1 Flow fields

The flow fields store the deformation information. The same fields can be used for both forward or backward deformations (or even, in principle, half way or exaggerated deformations).

### 25.5.2 Images

The flow field deformations can be applied to multiple images. At this point, you are choosing how many images each flow field should be applied to.

#### Images

Select images to be warped. Note that there should be the same number of images as there are flow fields, such that each flow field warps one image.

### 25.5.3 Modulation

This allows the spatially normalised images to be rescaled by the Jacobian determinants of the deformations. Note that the rescaling is only approximate for deformations generated using smaller numbers of time steps.

### 25.5.4 Time Steps

The number of time points used for solving the partial differential equations. Note that Jacobian determinants are not very accurate for very small numbers of time steps (less than about 16).

### 25.5.5 Interpolation

The method by which the images are sampled when being written in a different space. (Note that Inf or NaN values are treated as zero, rather than as missing data)

Nearest Neighbour:

- Fastest, but not normally recommended.

Trilinear Interpolation:

- OK for PET, realigned fMRI, or segmentations

B-spline Interpolation:

- Better quality (but slower) interpolation [93], especially with higher degree splines. Can produce values outside the original range (e.g. small negative values from an originally all positive image).

## 25.6 Jacobian determinants

Create Jacobian determinant fields from flowfields.

### 25.6.1 Flow fields

The flow fields store the deformation information. The same fields can be used for both forward or backward deformations (or even, in principle, half way or exaggerated deformations).

### 25.6.2 Time Steps

The number of time points used for solving the partial differential equations. Note that Jacobian determinants are not very accurate for very small numbers of time steps (less than about 16).

## 25.7 Create Inverse Warped

Create inverse normalised versions of some image(s). The image that is inverse-normalised should be in alignment with the template (generated during the warping procedure). Note that the results have the same dimensions as the “flow fields”, but are mapped to the original images via the affine transformations in their headers.

### 25.7.1 Flow fields

The flow fields store the deformation information. The same fields can be used for both forward or backward deformations (or even, in principle, half way or exaggerated deformations).

### 25.7.2 Images

Select the image(s) to be inverse normalised. These should be in alignment with the template image of the warping procedure (Run Dartel).

### 25.7.3 Time Steps

The number of time points used for solving the partial differential equations. Note that Jacobian determinants are not very accurate for very small numbers of time steps (less than about 16).

### 25.7.4 Interpolation

The method by which the images are sampled when being written in a different space. (Note that Inf or NaN values are treated as zero, rather than as missing data)

Nearest Neighbour:

- Fastest, but not normally recommended.

Trilinear Interpolation:

- OK for PET, realigned fMRI, or segmentations

B-spline Interpolation:

- Better quality (but slower) interpolation [93], especially with higher degree splines. Can produce values outside the original range (e.g. small negative values from an originally all positive image).

## 25.8 Population to ICBM Registration

Estimate the mapping from the population average to ICBM space. This is achieved by aligning the population average with grey and white matter tissue probability maps generated by Vladimir Fonov of the McConnell Brain Imaging Centre, Montreal Neurological Institute, McGill University [26].

### 25.8.1 Dartel Template

Select the final Template file generated by Dartel.

## 25.9 Kernel Utilities

Dartel can be used for generating matrices of dot-products for various kernel pattern-recognition procedures.

The idea of applying pattern-recognition procedures is to obtain a multi-variate characterisation of the anatomical differences among groups of subjects. These characterisations can then be used to separate (eg) healthy individuals from particular patient populations. There is still a great deal of methodological work to be done, so the types of kernel that can be generated here are unlikely to be the definitive ways of proceeding. They are only just a few ideas that may be worth trying out. The idea is simply to attempt a vaguely principled way to combine generative

models with discriminative models (see the “Pattern Recognition and Machine Learning” book by Chris Bishop for more ideas). Better ways (higher predictive accuracy) will eventually emerge.

Various pattern recognition algorithms are available freely over the Internet. Possible approaches include Support-Vector Machines, Relevance-Vector machines and Gaussian Process Models. Gaussian Process Models probably give the most accurate probabilistic predictions, and allow kernels generated from different pieces of data to be most easily combined.

### 25.9.1 Kernel from Images

Generate a kernel matrix from images. In principle, this same function could be used for generating kernels from any image data (e.g. “modulated” grey matter). If there is prior knowledge about some region providing more predictive information (e.g. the hippocampi for AD), then it is possible to weight the generation of the kernel accordingly. The matrix of dot-products is saved in a variable “Phi”, which can be loaded from the `dp_*.mat` file. The “kernel trick” can be used to convert these dot-products into distance measures for e.g. radial basis-function approaches.

#### Data

Select images to generate dot-products from.

#### Weighting image

The kernel can be generated so that some voxels contribute to the similarity measures more than others. This is achieved by supplying a weighting image, which each of the component images are multiplied before the dot-products are computed. This image needs to have the same dimensions as the component images, but orientation information (encoded by matrices in the headers) is ignored. If left empty, then all voxels are weighted equally.

#### Dot-product Filename

Enter a filename for results (it will be prefixed by “dp\_” and saved in the current directory).

### 25.9.2 Kernel from Flows

Generate a kernel from flow fields. The dot-products are saved in a variable “Phi” in the resulting `dp_*.mat` file.

#### Flow fields

Select the flow fields for each subject.

#### Regularisation Form

The registration is penalised by some “energy” term. Here, the form of this energy term is specified. Three different forms of regularisation can currently be used.

#### Reg params

For linear elasticity, the parameters are ‘mu’, ‘lambda’ and ‘id’. For membrane and bending energy, the parameters are ‘lambda’, unused and ‘id’. The term ‘id’ is for penalising absolute displacements, and should therefore be small.

#### Dot-product Filename

Enter a filename for results (it will be prefixed by “dp\_” and saved in the current directory).

# Chapter 26

## Shoot Tools

### Contents

---

<b>26.1 Run Shooting (create Templates)</b> . . . . .	<b>195</b>
26.1.1 Images . . . . .	196
<b>26.2 Run Shoot (existing Templates)</b> . . . . .	<b>196</b>
26.2.1 Images . . . . .	196
26.2.2 Templates . . . . .	196
<b>26.3 Kernel Utilities</b> . . . . .	<b>196</b>
26.3.1 Kernel from velocities . . . . .	196
26.3.2 Generate Scalar Momenta . . . . .	197
26.3.3 Kernel from Images . . . . .	197

---

This toolbox is based around the “Diffeomorphic Registration using Geodesic Shooting and Gauss-Newton Optimisation” paper, which has been submitted to NeuroImage. The idea is to register images by estimating an initial velocity field, which can then be integrated to generate both forward and backward deformations. Currently, the software only works with images that have isotropic voxels, identical dimensions and which are in approximate alignment with each other. One of the reasons for this is that the approach assumes circulant boundary conditions, which makes modelling global rotations impossible. Because of these limitations, the registration should be based on images that have first been “imported” via the New Segment toolbox.

The next step is the registration itself, which involves the simultaneous registration of e.g. GM with GM, WM with WM and 1-(GM+WM) with 1-(GM+WM) (when needed, the 1-(GM+WM) class is generated implicitly, so there is no need to include this class yourself). This procedure begins by creating a mean of all the images, which is used as an initial template. Deformations from this template to each of the individual images are computed, and the template is then re-generated by applying the inverses of the deformations to the images and averaging. This procedure is repeated a number of times.

This toolbox should be considered as only a beta (trial) version, and will include a number of (as yet unspecified) extensions in future updates. Please report any bugs or problems to the SPM mailing list.

### 26.1 Run Shooting (create Templates)

Run the geodesic shooting nonlinear image registration procedure. This involves iteratively matching all the selected images to a template generated from their own mean. A series of Template\*.nii files are generated, which become increasingly crisp as the registration proceeds.

### 26.1.1 Images

Select the images to be warped together. Multiple sets of images can be simultaneously registered. For example, the first set may be a bunch of grey matter images, and the second set may be the white matter images of the same subjects.

#### Images

Select a set of imported images of the same type to be registered by minimising a measure of difference from the template.

## 26.2 Run Shoot (existing Templates)

Run the Shoot nonlinear image registration procedure to match individual images to pre-existing template data. Start out with smooth templates, and select crisp templates for the later iterations.

### 26.2.1 Images

Select the images to be warped together. Multiple sets of images can be simultaneously registered. For example, the first set may be a bunch of grey matter images, and the second set may be the white matter images of the same subjects.

#### Images

Select a set of imported images of the same type to be registered by minimising a measure of difference from the template.

### 26.2.2 Templates

Select templates. Smoother templates should be used for the early iterations. Note that the template should be a 4D file, with the 4th dimension equal to the number of sets of images.

## 26.3 Kernel Utilities

Shoot can be used for generating matrices of dot-products for various kernel pattern-recognition procedures.

The idea of applying pattern-recognition procedures is to obtain a multi-variate characterisation of the anatomical differences among groups of subjects. These characterisations can then be used to separate (eg) healthy individuals from particular patient populations. There is still a great deal of methodological work to be done, so the types of kernel that can be generated here are unlikely to be the definitive ways of proceeding. They are only just a few ideas that may be worth trying out. The idea is simply to attempt a vaguely principled way to combine generative models with discriminative models (see the “Pattern Recognition and Machine Learning” book by Chris Bishop for more ideas). Better ways (higher predictive accuracy) will eventually emerge.

Various pattern recognition algorithms are available freely over the Internet. Possible approaches include Support-Vector Machines and Gaussian Process Models. Gaussian Process Models probably give the most accurate probabilistic predictions, and allow kernels generated from different pieces of data to be most easily combined.

### 26.3.1 Kernel from velocities

Generate a kernel from velocity fields. The dot-products are saved in a variable “K” in the resulting `dp_*.mat` file.

#### Velocity fields

Select the velocity fields for each subject.



**Dot-product Filename**

Enter a filename for results (it will be prefixed by “dp\_” and saved in the current directory).

**26.3.2 Generate Scalar Momenta**

Generate spatially smoothed “scalar momenta” [citesingh2010multivariate,singh2012genetic](#) in a form suitable for using with pattern recognition. In principle, a Gaussian Process model can be used to determine the optimal (positive) linear combination of kernel matrices. The idea would be to combine a kernel matrix derived from these, with a kernel derived from the velocity-fields. Such a combined kernel should then encode more relevant information than the individual kernels alone. The scalar momentum fields that are generated contain a number of volumes equal to the number of sets of “rc\*” images used (equal to the number of volumes in the template - 1). See Figures 10 and 11 of [\[9\]](#) for examples of scalar momenta (Jacobian scaled residuals) for simulated data.

**Template**

Residual differences are computed between the warped images and template. These are then scaled by the Jacobian determinants at each point, and spatially smoothed.

**Images**

Multiple sets of images are used here. For example, the first set may be a bunch of grey matter images, and the second set may be the white matter images of the same subjects. The number of sets of images must be the same as was used to generate the template.

**Images** Select tissue class images (one per subject).

**Deformation fields**

Select the deformation fields for each subject.

**Jacobian determinant fields**

Select the Jacobian determinant fields for each subject. Residual differences are computed between the warped images and template. These are then scaled by the Jacobian determinants at each point, and spatially smoothed.

**Smoothing**

The scalar momenta can be smoothed with a Gaussian to reduce dimensionality. More smoothing is recommended if there are fewer training images or if more channels of data were used for driving the registration. From preliminary experiments, a value of about 10mm seems to work reasonably well.

**26.3.3 Kernel from Images**

Generate a kernel matrix from images. In principle, this same function could be used for generating kernels from any image data (e.g. “modulated” grey matter). If there is prior knowledge about some region providing more predictive information (e.g. the hippocampi for AD), then it is possible to weight the generation of the kernel accordingly. The matrix of dot-products is saved in a variable “K”, which can be loaded from the dp\_\*.mat file. The “kernel trick” can be used to convert these dot-products into distance measures for e.g. radial basis-function approaches.

**Data**

Select images to generate dot-products from.

**Weighting image**

The kernel can be generated so that some voxels contribute to the similarity measures more than others. This is achieved by supplying a weighting image, which each of the component images are multiplied before the dot-products are computed. This image needs to have the same dimensions as the component images, but orientation information (encoded by matrices in the headers) is ignored. If left empty, then all voxels are weighted equally.

**Dot-product Filename**

Enter a filename for results (it will be prefixed by “dp\_” and saved in the current directory).

# Chapter 27

## Longitudinal Registration

### Contents

---

<b>27.1 Pairwise Longitudinal Registration</b> . . . . .	<b>199</b>
27.1.1 Time 1 Volumes . . . . .	200
27.1.2 Time 2 Volumes . . . . .	200
27.1.3 Time Difference . . . . .	200
27.1.4 Noise Estimate . . . . .	200
27.1.5 Warping Regularisation . . . . .	200
27.1.6 Bias Regularisation . . . . .	200
27.1.7 Save Mid-point average . . . . .	201
27.1.8 Save Jacobian Rate . . . . .	201
27.1.9 Save Divergence Rate . . . . .	201
27.1.10 Deformation Fields . . . . .	201
<b>27.2 Serial Longitudinal Registration</b> . . . . .	<b>201</b>
27.2.1 Volumes . . . . .	201
27.2.2 Times . . . . .	202
27.2.3 Noise Estimate . . . . .	202
27.2.4 Warping Regularisation . . . . .	202
27.2.5 Bias Regularisation . . . . .	202
27.2.6 Save Mid-point average . . . . .	203
27.2.7 Save Jacobians . . . . .	203
27.2.8 Save Divergence . . . . .	203
27.2.9 Deformation Fields . . . . .	203

---

### 27.1 Pairwise Longitudinal Registration

Longitudinal registration of pairs of anatomical MRI scans. It is based on pairwise inverse-consistent alignment between the first and second scan of each subject, and incorporates a bias field correction. Prior to running the registration, the scans should already be in very rough alignment, although because the model incorporates a rigid-body transform, this need not be extremely precise. Note that there are a bunch of hyper-parameters to be specified. If you are unsure what values to take, then the defaults should be a reasonable guess of what works. Note that changes to these hyper-parameters will impact the results obtained.

The alignment assumes that all scans have similar resolutions and dimensions, and were collected on the same (or very similar) MR scanner using the same pulse sequence. If these assumption are not correct, then the approach will not work as well.

### 27.1.1 Time 1 Volumes

Select first time point scans of each subject.

### 27.1.2 Time 2 Volumes

Select second time point scans of each subject. Note that the order that the first and second time points are specified should be the same. The algorithm does not incorporate any magical way of figuring out which scans go together.

### 27.1.3 Time Difference

Specify the time difference between the scans in years. This can be a single value (if it is the same for all subjects) or a vector of values (if it differs among subjects).

### 27.1.4 Noise Estimate

Specify the standard deviation of the noise in the images. If a scalar is entered, all images will be assumed to have the same level of noise. For any non-finite values, the algorithm will try to estimate reasonable noise estimates based on fitting a mixture of two Rician distributions to the intensity histogram of each of the images. This works reasonably well for simple MRI scans, but less well for derived images (such as averages). The assumption is that the residuals, after fitting the registration model, are i.i.d. Gaussian.

### 27.1.5 Warping Regularisation

Registration involves simultaneously minimising two terms. One of these is a measure of similarity between the images (mean-squared difference in the current situation), whereas the other is a measure of the roughness of the deformations. This measure of roughness involves the sum of the following terms:

- \* Absolute displacements need to be penalised by a tiny amount. The first element encodes the amount of penalty on these. Ideally, absolute displacements should not be penalised, but it is necessary for technical reasons.

- \* The ‘membrane energy’ of the deformation is penalised (2nd element), usually by a relatively small amount. This penalises the sum of squares of the derivatives of the velocity field (ie the sum of squares of the elements of the Jacobian tensors).

- \* The ‘bending energy’ is penalised (3rd element). This penalises the sum of squares of the 2nd derivatives of the velocity.

- \* Linear elasticity regularisation is also included (4th and 5th elements). The first parameter ( $\mu$ ) is similar to that for linear elasticity, except it penalises the sum of squares of the Jacobian tensors after they have been made symmetric (by averaging with the transpose). This term essentially penalises length changes, without penalising rotations.

- \* The final term also relates to linear elasticity, and is the weight that denotes how much to penalise changes to the divergence of the velocities ( $\lambda$ ). This divergence is a measure of the rate of volumetric expansion or contraction.

Note that regularisation is specified based on what is believed to be appropriate for a year of growth. The specified values are divided by the number of years time difference.

### 27.1.6 Bias Regularisation

MR images are usually corrupted by a smooth, spatially varying artifact that modulates the intensity of the image (bias). These artifacts, although not usually a problem for visual inspection, can impede automated processing of the images.

An important issue relates to the distinction between variations in the difference between the images that arise because of the differential bias artifact due to the physics of MR scanning, and those that arise due to shape differences. The objective is to model the latter by deformations, while modelling the former with a bias field. We know a priori that intensity variations due to MR physics tend to be spatially smooth. A more accurate estimate of a bias field can be obtained

by including prior knowledge about the distribution of the fields likely to be encountered by the correction algorithm. For example, if it is known that there is little or no intensity non-uniformity, then it would be wise to penalise large estimates of the intensity non-uniformity.

Knowing what works best should be a matter of empirical exploration, as it depends on the scans themselves. For example, if your data has very little of the artifact, then the bias regularisation should be increased. This effectively tells the algorithm that there is very little bias in your data, so it does not try to model it.

### 27.1.7 Save Mid-point average

Do you want to save the mid-point average template image? This is likely to be useful for groupwise alignment, and is prefixed by “avg\_” and written out in the same directory of the first time point data.

### 27.1.8 Save Jacobian Rate

Do you want to save a map of the differences between the Jacobian determinants, divided by the time interval? Some consider these useful for morphometrics (although the divergences of the initial velocities may be preferable). The difference between two Jacobian determinants is computed and this is divided by the time interval. One original Jacobian map is for the deformation from the mid point to the first scan, and the other is for the deformation from the mid point to the second scan. Each of these encodes the relative volume (at each spatial location) between the scan and the mid-point average. Values less than 0 indicate contraction (over time), whereas values greater than zero indicate expansion. These files are prefixed by “jd\_” and written out in the same directory of the first time point data.

### 27.1.9 Save Divergence Rate

Do you want to save a map of divergence of the velocity field? This is useful for morphometrics, and may be considered as the rate of volumetric expansion. Negative values indicate contraction. These files are prefixed by “dv\_” and written out in the same directory of the first time point data. Note that the divergences written out have been divided by the time interval between scans

### 27.1.10 Deformation Fields

Deformation fields can be saved to disk, and used by the Deformations Utility. Deformations are saved as `y_*.nii` files, which contain three volumes to encode the x, y and z coordinates. They are written in the same directory as the corresponding image.

## 27.2 Serial Longitudinal Registration

Longitudinal registration of series of anatomical MRI scans for a single subject. It is based on groupwise alignment among each of the subject’s scans, and incorporates a bias field correction. Prior to running the registration, the scans should already be in very rough alignment, although because the model incorporates a rigid-body transform, this need not be extremely precise. Note that there are a bunch of hyper-parameters to be specified. If you are unsure what values to take, then the defaults should be a reasonable guess of what works. Note that changes to these hyper-parameters will impact the results obtained.

The alignment assumes that all scans have similar resolutions and dimensions, and were collected on the same (or very similar) MR scanner using the same pulse sequence. If these assumption are not correct, then the approach will not work as well. There are a number of settings (noise estimate, regularisation etc). Default settings often work well, but it can be very helpful to try some different values, as these can have a large effect on the results.

### 27.2.1 Volumes

Select scans for this subject.

### 27.2.2 Times

Specify the times of the scans in years.

### 27.2.3 Noise Estimate

Specify the standard deviation of the noise in the images. If a scalar is entered, all images will be assumed to have the same level of noise. For any non-finite values, the algorithm will try to estimate reasonable noise estimates based on fitting a mixture of two Rician distributions to the intensity histogram of each of the images. This works reasonably well for simple MRI scans, but less well for derived images (such as averages). The assumption is that the residuals, after fitting the registration model, are i.i.d. Gaussian.

### 27.2.4 Warping Regularisation

Registration involves simultaneously minimising two terms. One of these is a measure of similarity between the images (mean-squared difference in the current situation), whereas the other is a measure of the roughness of the deformations. This measure of roughness involves the sum of the following terms:

- \* Absolute displacements need to be penalised by a tiny amount. The first element encodes the amount of penalty on these. Ideally, absolute displacements should not be penalised, but it is necessary for technical reasons.

- \* The ‘membrane energy’ of the deformation is penalised (2nd element), usually by a relatively small amount. This penalises the sum of squares of the derivatives of the velocity field (ie the sum of squares of the elements of the Jacobian tensors).

- \* The ‘bending energy’ is penalised (3rd element). This penalises the sum of squares of the 2nd derivatives of the velocity.

- \* Linear elasticity regularisation is also included (4th and 5th elements). The first parameter ( $\mu$ ) is similar to that for linear elasticity, except it penalises the sum of squares of the Jacobian tensors after they have been made symmetric (by averaging with the transpose). This term essentially penalises length changes, without penalising rotations.

- \* The final term also relates to linear elasticity, and is the weight that denotes how much to penalise changes to the divergence of the velocities ( $\lambda$ ). This divergence is a measure of the rate of volumetric expansion or contraction.

Note that regularisation is specified based on what is believed to be appropriate for a year of growth. The specified values are divided by the number of years time difference.

### 27.2.5 Bias Regularisation

MR images are usually corrupted by a smooth, spatially varying artifact that modulates the intensity of the image (bias). These artifacts, although not usually a problem for visual inspection, can impede automated processing of the images.

An important issue relates to the distinction between variations in the difference between the images that arise because of the differential bias artifact due to the physics of MR scanning, and those that arise due to shape differences. The objective is to model the latter by deformations, while modelling the former with a bias field. We know a priori that intensity variations due to MR physics tend to be spatially smooth. A more accurate estimate of a bias field can be obtained by including prior knowledge about the distribution of the fields likely to be encountered by the correction algorithm. For example, if it is known that there is little or no intensity non-uniformity, then it would be wise to penalise large estimates of the intensity non-uniformity.

Knowing what works best should be a matter of empirical exploration, as it depends on the scans themselves. For example, if your data has very little of the artifact, then the bias regularisation should be increased. This effectively tells the algorithm that there is very little bias in your data, so it does not try to model it.

### 27.2.6 Save Mid-point average

Do you want to save the mid-point average template image? This is likely to be useful for groupwise alignment, and is prefixed by “avg\_” and written out in the same directory of the first time point data.

### 27.2.7 Save Jacobians

Do you want to save a map of the Jacobian determinants? Some consider these useful for morphometrics (although the divergences of the initial velocities may be preferable). Each map of Jacobians encodes the relative volume (at each spatial location) between the scan and the median time-point average. Values less than one indicate contraction (over time), whereas values greater than one indicate expansion. These files are prefixed by “j\_” and written out in the same directory of the first time point data.

### 27.2.8 Save Divergence

Do you want to save a map of divergence of the velocity field? This is useful for morphometrics, and may be considered as the rate of volumetric expansion. Negative values indicate contraction. These files are prefixed by “dv\_” and written out in the same directory of the first time point data.

### 27.2.9 Deformation Fields

Deformation fields can be saved to disk, and used by the Deformations Utility. Deformations are saved as `y_*.nii` files, which contain three volumes to encode the x, y and z coordinates. They are written in the same directory as the corresponding image.





# Chapter 28

## Old Normalise

### Contents

---

<b>28.1 Old Normalise: Estimate</b> . . . . .	<b>206</b>
28.1.1 Data . . . . .	206
28.1.2 Estimation Options . . . . .	206
<b>28.2 Old Normalise: Write</b> . . . . .	<b>207</b>
28.2.1 Data . . . . .	207
28.2.2 Writing Options . . . . .	208
<b>28.3 Old Normalise: Estimate &amp; Write</b> . . . . .	<b>208</b>
28.3.1 Data . . . . .	208
28.3.2 Estimation Options . . . . .	209
28.3.3 Writing Options . . . . .	210

---

This very ancient module spatially (stereotactically) normalises MRI, PET or SPECT images into a standard space defined by some ideal model or template image[s]. The template images supplied with SPM conform to the space defined by the ICBM, NIH P-20 project, and approximate that of the the space described in the atlas of Talairach and Tournoux (1988). The transformation can also be applied to any other image that has been coregistered with these scans. A few researchers may wish to continue using this strategy, but (when good quality anatomical MRI scans are available) the DARTEL approach is now generally recommended instead.

Generally, the algorithms work by minimising the sum of squares difference between the image which is to be normalised, and a linear combination of one or more template images. For the least squares registration to produce an unbiased estimate of the spatial transformation, the image contrast in the templates (or linear combination of templates) should be similar to that of the image from which the spatial normalisation is derived. The registration simply searches for an optimum solution. If the starting estimates are not good, then the optimum it finds may not find the global optimum.

The first step of the normalisation is to determine the optimum 12-parameter affine transformation. Initially, the registration is performed by matching the whole of the head (including the scalp) to the template. Following this, the registration proceeded by only matching the brains together, by appropriate weighting of the template voxels. This is a completely automated procedure (that does not require “scalp editing”) that discounts the confounding effects of skull and scalp differences. A Bayesian framework is used, such that the registration searches for the solution that maximises the a posteriori probability of it being correct [10] . i.e., it maximises the product of the likelihood function (derived from the residual squared difference) and the prior function (which is based on the probability of obtaining a particular set of zooms and shears).

The affine registration is followed by estimating nonlinear deformations, whereby the deformations are defined by a linear combination of three dimensional discrete cosine transform (DCT)

basis functions [5]. The default options result in each of the deformation fields being described by 1176 parameters, where these represent the coefficients of the deformations in three orthogonal directions. The matching involved simultaneously minimising the membrane energies of the deformation fields and the residual squared difference between the images and template(s).

The primary use is for stereotactic normalisation to facilitate inter-subject averaging and precise characterisation of functional anatomy [4]. It is not necessary to spatially normalise the data (this is only a pre-requisite for inter-subject averaging or reporting in the Talairach space). If you wish to circumnavigate this step (e.g. if you have single slice data or do not have an appropriate high resolution MRI scan) simply specify where you think the anterior commissure is with the ORIGIN in the header of the first scan (using the 'Display' facility) and proceed directly to 'Smoothing' or 'Statistics'.

All normalised images are written to the same subdirectory as the original images, prefixed with a 'w'. The details of the transformations are displayed in the results window, and the parameters are saved in the "\*\_sn.mat" file.

## 28.1 Old Normalise: Estimate

Computes the warp that best registers a source image (or series of source images) to match a template, saving it to a file `imagename'_sn.mat'`.

### 28.1.1 Data

List of subjects. Images of each subject should be warped differently.

#### Subject

Data for this subject. The same parameters are used within subject.

**Source Image** The image that is warped to match the template(s). The result is a set of warps, which can be applied to this image, or any other image that is in register with it.

**Source Weighting Image** Optional weighting images (consisting of pixel values between the range of zero to one) to be used for registering abnormal or lesioned brains. These images should match the dimensions of the image from which the parameters are estimated, and should contain zeros corresponding to regions of abnormal tissue.

### 28.1.2 Estimation Options

Various settings for estimating warps.

#### Template Image

Specify a template image to match the source image with. The contrast in the template must be similar to that of the source image in order to achieve a good registration. It is also possible to select more than one template, in which case the registration algorithm will try to find the best linear combination of these images in order to best model the intensities in the source image.

#### Template Weighting Image

Applies a weighting mask to the template(s) during the parameter estimation. With the default brain mask, weights in and around the brain have values of one whereas those clearly outside the brain are zero. This is an attempt to base the normalisation purely upon the shape of the brain, rather than the shape of the head (since low frequency basis functions can not really cope with variations in skull thickness).

The option is now available for a user specified weighting image. This should have the same dimensions and mat file as the template images, with values in the range of zero to one.

**Source Image Smoothing**

Smoothing to apply to a copy of the source image. The template and source images should have approximately the same smoothness. Remember that the templates supplied with SPM have been smoothed by 8mm, and that smoothnesses combine by Pythagoras' rule.

**Template Image Smoothing**

Smoothing to apply to a copy of the template image. The template and source images should have approximately the same smoothness. Remember that the templates supplied with SPM have been smoothed by 8mm, and that smoothnesses combine by Pythagoras' rule.

**Affine Regularisation**

Affine registration into a standard space can be made more robust by regularisation (penalising excessive stretching or shrinking). The best solutions can be obtained by knowing the approximate amount of stretching that is needed (e.g. ICBM templates are slightly bigger than typical brains, so greater zooms are likely to be needed). If registering to an image in ICBM/MNI space, then choose the first option. If registering to a template that is close in size, then select the second option. If you do not want to regularise, then choose the third.

**Nonlinear Frequency Cutoff**

Cutoff of DCT bases. Only DCT bases of periods longer than the cutoff are used to describe the warps. The number used will depend on the cutoff and the field of view of the template image(s).

**Nonlinear Iterations**

Number of iterations of nonlinear warping performed.

**Nonlinear Regularisation**

The amount of regularisation for the nonlinear part of the spatial normalisation. Pick a value around one. However, if your normalised images appear distorted, then it may be an idea to increase the amount of regularisation (by an order of magnitude) - or even just use an affine normalisation. The regularisation influences the smoothness of the deformation fields.

## 28.2 Old Normalise: Write

Allows previously estimated warps (stored in `image_name'_sn.mat'` files) to be applied to series of images.

### 28.2.1 Data

List of subjects. Images of each subject should be warped differently.

**Subject**

Data for this subject. The same parameters are used within subject.

**Parameter File** Select the `'_sn.mat'` file containing the spatial normalisation parameters for that subject.

**Images to Write** These are the images for warping according to the estimated parameters. They can be any images that are in register with the "source" image used to generate the parameters.

## 28.2.2 Writing Options

Various options for writing normalised images.

### Preserve

Preserve Concentrations: Spatially normalised images are not "modulated". The warped images preserve the intensities of the original images.

Preserve Total: Spatially normalised images are "modulated" in order to preserve the total amount of signal in the images. Areas that are expanded during warping are correspondingly reduced in intensity.

### Bounding box

The bounding box (in mm) of the volume which is to be written (relative to the anterior commissure).

### Voxel sizes

The voxel sizes (x, y & z, in mm) of the written normalised images.

### Interpolation

The method by which the images are sampled when being written in a different space. (Note that Inf or NaN values are treated as zero, rather than as missing data)

Nearest Neighbour:

- Fastest, but not normally recommended.

Trilinear Interpolation:

- OK for PET, realigned fMRI, or segmentations

B-spline Interpolation:

- Better quality (but slower) interpolation [93], especially with higher degree splines. Can produce values outside the original range (e.g. small negative values from an originally all positive image).

### Wrapping

These are typically:

No wrapping: for PET or images that have already been spatially transformed.

Wrap in Y: for (un-resliced) MRI where phase encoding is in the Y direction (voxel space).

### Filename Prefix

Specify the string to be prepended to the filenames of the normalised image file(s). Default prefix is 'w'.

## 28.3 Old Normalise: Estimate & Write

Computes the warp that best registers a source image (or series of source images) to match a template, saving it to the file `imagename'_sn.mat'`. This option also allows the contents of the `imagename'_sn.mat'` files to be applied to a series of images.

### 28.3.1 Data

List of subjects. Images of each subject should be warped differently.

#### Subject

Data for this subject. The same parameters are used within subject.

**Source Image** The image that is warped to match the template(s). The result is a set of warps, which can be applied to this image, or any other image that is in register with it.

**Source Weighting Image** Optional weighting images (consisting of pixel values between the range of zero to one) to be used for registering abnormal or lesioned brains. These images should match the dimensions of the image from which the parameters are estimated, and should contain zeros corresponding to regions of abnormal tissue.

**Images to Write** These are the images for warping according to the estimated parameters. They can be any images that are in register with the "source" image used to generate the parameters.

### 28.3.2 Estimation Options

Various settings for estimating warps.

#### Template Image

Specify a template image to match the source image with. The contrast in the template must be similar to that of the source image in order to achieve a good registration. It is also possible to select more than one template, in which case the registration algorithm will try to find the best linear combination of these images in order to best model the intensities in the source image.

#### Template Weighting Image

Applies a weighting mask to the template(s) during the parameter estimation. With the default brain mask, weights in and around the brain have values of one whereas those clearly outside the brain are zero. This is an attempt to base the normalisation purely upon the shape of the brain, rather than the shape of the head (since low frequency basis functions can not really cope with variations in skull thickness).

The option is now available for a user specified weighting image. This should have the same dimensions and mat file as the template images, with values in the range of zero to one.

#### Source Image Smoothing

Smoothing to apply to a copy of the source image. The template and source images should have approximately the same smoothness. Remember that the templates supplied with SPM have been smoothed by 8mm, and that smoothnesses combine by Pythagoras' rule.

#### Template Image Smoothing

Smoothing to apply to a copy of the template image. The template and source images should have approximately the same smoothness. Remember that the templates supplied with SPM have been smoothed by 8mm, and that smoothnesses combine by Pythagoras' rule.

#### Affine Regularisation

Affine registration into a standard space can be made more robust by regularisation (penalising excessive stretching or shrinking). The best solutions can be obtained by knowing the approximate amount of stretching that is needed (e.g. ICBM templates are slightly bigger than typical brains, so greater zooms are likely to be needed). If registering to an image in ICBM/MNI space, then choose the first option. If registering to a template that is close in size, then select the second option. If you do not want to regularise, then choose the third.

#### Nonlinear Frequency Cutoff

Cutoff of DCT bases. Only DCT bases of periods longer than the cutoff are used to describe the warps. The number used will depend on the cutoff and the field of view of the template image(s).

**Nonlinear Iterations**

Number of iterations of nonlinear warping performed.

**Nonlinear Regularisation**

The amount of regularisation for the nonlinear part of the spatial normalisation. Pick a value around one. However, if your normalised images appear distorted, then it may be an idea to increase the amount of regularisation (by an order of magnitude) - or even just use an affine normalisation. The regularisation influences the smoothness of the deformation fields.

**28.3.3 Writing Options**

Various options for writing normalised images.

**Preserve**

Preserve Concentrations: Spatially normalised images are not "modulated". The warped images preserve the intensities of the original images.

Preserve Total: Spatially normalised images are "modulated" in order to preserve the total amount of signal in the images. Areas that are expanded during warping are correspondingly reduced in intensity.

**Bounding box**

The bounding box (in mm) of the volume which is to be written (relative to the anterior commissure).

**Voxel sizes**

The voxel sizes (x, y & z, in mm) of the written normalised images.

**Interpolation**

The method by which the images are sampled when being written in a different space. (Note that Inf or NaN values are treated as zero, rather than as missing data)

Nearest Neighbour:

- Fastest, but not normally recommended.

Trilinear Interpolation:

- OK for PET, realigned fMRI, or segmentations

B-spline Interpolation:

- Better quality (but slower) interpolation [93], especially with higher degree splines. Can produce values outside the original range (e.g. small negative values from an originally all positive image).

**Wrapping**

These are typically:

No wrapping: for PET or images that have already been spatially transformed.

Wrap in Y: for (un-resliced) MRI where phase encoding is in the Y direction (voxel space).

**Filename Prefix**

Specify the string to be prepended to the filenames of the normalised image file(s). Default prefix is 'w'.

# Chapter 29

## Old Segment

### Contents

---

<b>29.1 Data</b>	<b>212</b>
<b>29.2 Output Files</b>	<b>212</b>
29.2.1 Grey Matter	215
29.2.2 White Matter	215
29.2.3 Cerebro-Spinal Fluid	215
29.2.4 Bias Corrected	215
29.2.5 Clean up any partitions	215
<b>29.3 Custom</b>	<b>215</b>
29.3.1 Tissue probability maps	215
29.3.2 Gaussians per class	216
29.3.3 Affine Regularisation	216
29.3.4 Warping Regularisation	216
29.3.5 Warp Frequency Cutoff	217
29.3.6 Bias regularisation	217
29.3.7 Bias FWHM	217
29.3.8 Sampling distance	217
29.3.9 Masking image	217

---

Segment, bias correct and spatially normalise - all in the same model [8]. This function can be used for bias correcting, spatially normalising or segmenting your data. Note that this module needs the images to be roughly aligned with the tissue probability maps before you begin. If strange results are obtained, then this is usually because the images were poorly aligned beforehand. The Display option can be used to manually reposition the images so that the AC is close to coordinate 0,0,0 (within a couple of cm) and the orientation is within a few degrees of the tissue probability map data.

Many investigators use tools within older versions of SPM for a technique that has become known as "optimised" voxel-based morphometry (VBM). VBM performs region-wise volumetric comparisons among populations of subjects. It requires the images to be spatially normalised, segmented into different tissue classes, and smoothed, prior to performing statistical tests [98, 73, 6, 7]. The "optimised" pre-processing strategy involved spatially normalising subjects' brain images to a standard space, by matching grey matter in these images, to a grey matter reference. The historical motivation behind this approach was to reduce the confounding effects of non-brain (e.g. scalp) structural variability on the registration. Tissue classification in older versions of SPM required the images to be registered with tissue probability maps. After registration, these maps represented the prior probability of different tissue classes being found at each location in an

image. Bayes rule can then be used to combine these priors with tissue type probabilities derived from voxel intensities, to provide the posterior probability.

This procedure was inherently circular, because the registration required an initial tissue classification, and the tissue classification requires an initial registration. This circularity is resolved here by combining both components into a single generative model. This model also includes parameters that account for image intensity non-uniformity. Estimating the model parameters (for a maximum a posteriori solution) involves alternating among classification, bias correction and registration steps. This approach provides better results than simple serial applications of each component.

Note that multi-spectral segmentation (e.g. from a registered T1 and T2 image) is not yet implemented, but is planned for a future SPM version.

## 29.1 Data

Select scans for processing. This assumes that there is one scan for each subject. Note that multi-spectral (when there are two or more registered images of different contrasts) processing is not yet implemented for this method.

## 29.2 Output Files

This routine produces spatial normalisation parameters (\*\_seg\_sn.mat files) by default. These can be used for writing spatially normalised versions of your data, via the "Normalise: Write" option. This mechanism may produce superior results than the "Normalise: Estimate" option (but probably not as good as those produced using DARTEL).

In addition, it also produces files that can be used for doing inverse normalisation. If you have an image of regions defined in the standard space, then the inverse deformations can be used to warp these regions so that it approximately overlay your image. To use this facility, the bounding-box and voxel sizes should be set to non-finite values (e.g. [NaN NaN NaN] for the voxel sizes, and ones(2,3)\*NaN for the bounding box. This would be done by the spatial normalisation module, which allows you to select a set of parameters that describe the nonlinear warps, and the images that they should be applied to.

There are a number of options about what data you would like the routine to produce. The routine can be used for producing images of tissue classes, as well as bias corrected images. The native space option will produce a tissue class image (c\*) that is in alignment with the original (see Figure 29.1). You can also produce spatially normalised versions - both with (mwc\*) and without (wc\*) modulation (see Figure 29.2). The bounding box and voxel sizes of the spatially normalised versions are the same as that of the tissue probability maps with which they are registered. These can be used for doing voxel-based morphometry with (also see the "Using DARTEL" chapter of the manual). All you need to do is smooth them and do the stats (which means no more questions on the mailing list about how to do "optimized VBM").

Modulation is to compensate for the effect of spatial normalisation. When warping a series of images to match a template, it is inevitable that volumetric differences will be introduced into the warped images. For example, if one subject's temporal lobe has half the volume of that of the template, then its volume will be doubled during spatial normalisation. This will also result in a doubling of the voxels labelled grey matter. In order to remove this confound, the spatially normalised grey matter (or other tissue class) is adjusted by multiplying by its relative volume before and after warping. If warping results in a region doubling its volume, then the correction will halve the intensity of the tissue label. This whole procedure has the effect of preserving the total amount of grey matter signal in the normalised partitions.

A deformation field is a vector field, where three values are associated with each location in the field. The field maps from co-ordinates in the normalised image back to co-ordinates in the original image. The value of the field at co-ordinate [x y z] in the normalised space will be the co-ordinate [x' y' z'] in the original volume. The gradient of the deformation field at a co-ordinate is its Jacobian matrix, and it consists of a 3x3 matrix:



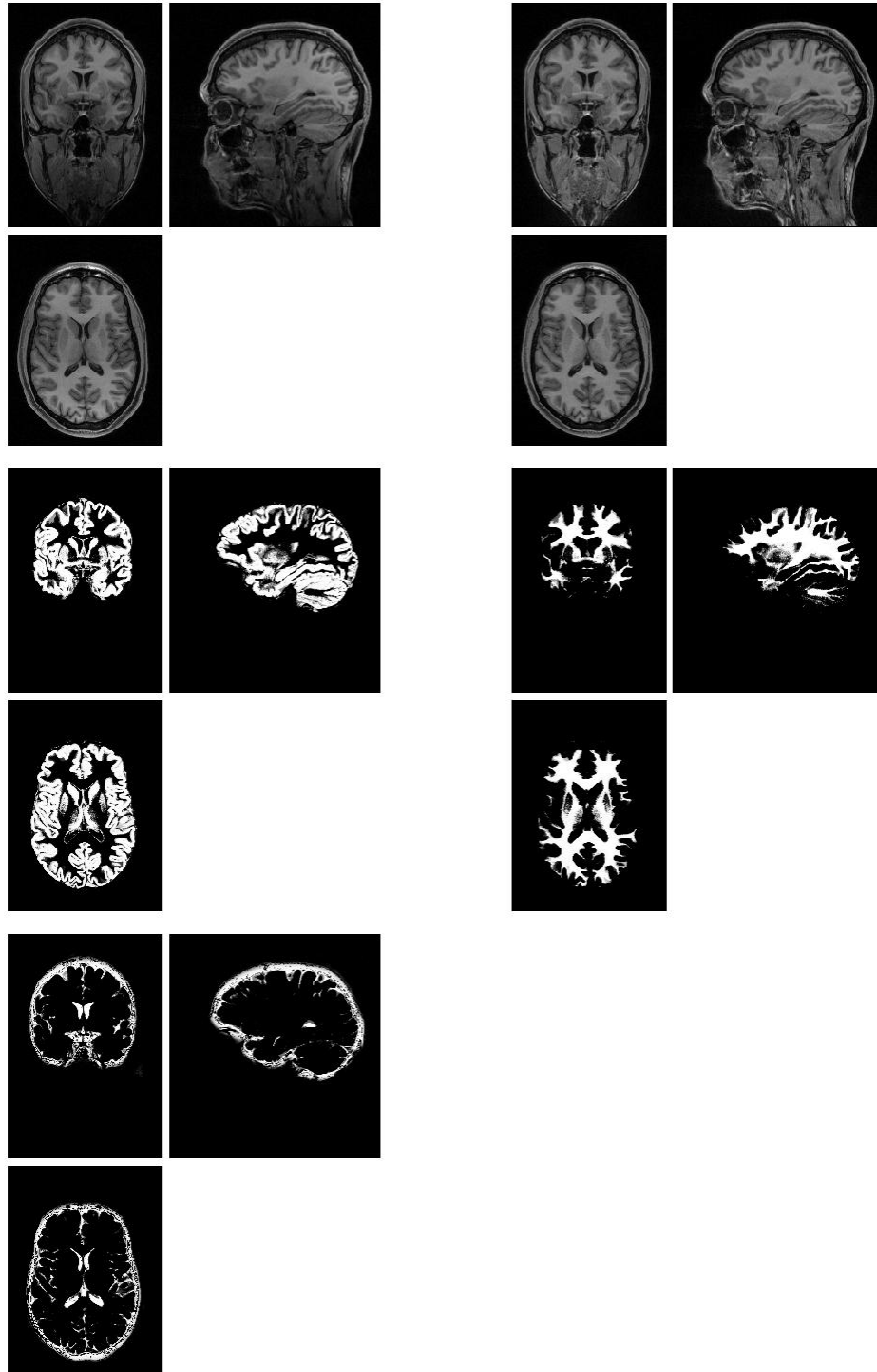


Figure 29.1: Segmentation results. These are the results that can be obtained in the original space of the image (i.e. the results that are not spatially normalised). Top left: original image (X.img). Top right: bias corrected image (mX.img). Middle and bottom rows: segmented grey matter (c1X.img), white matter (c2X.img) and CSF (c3X.img).

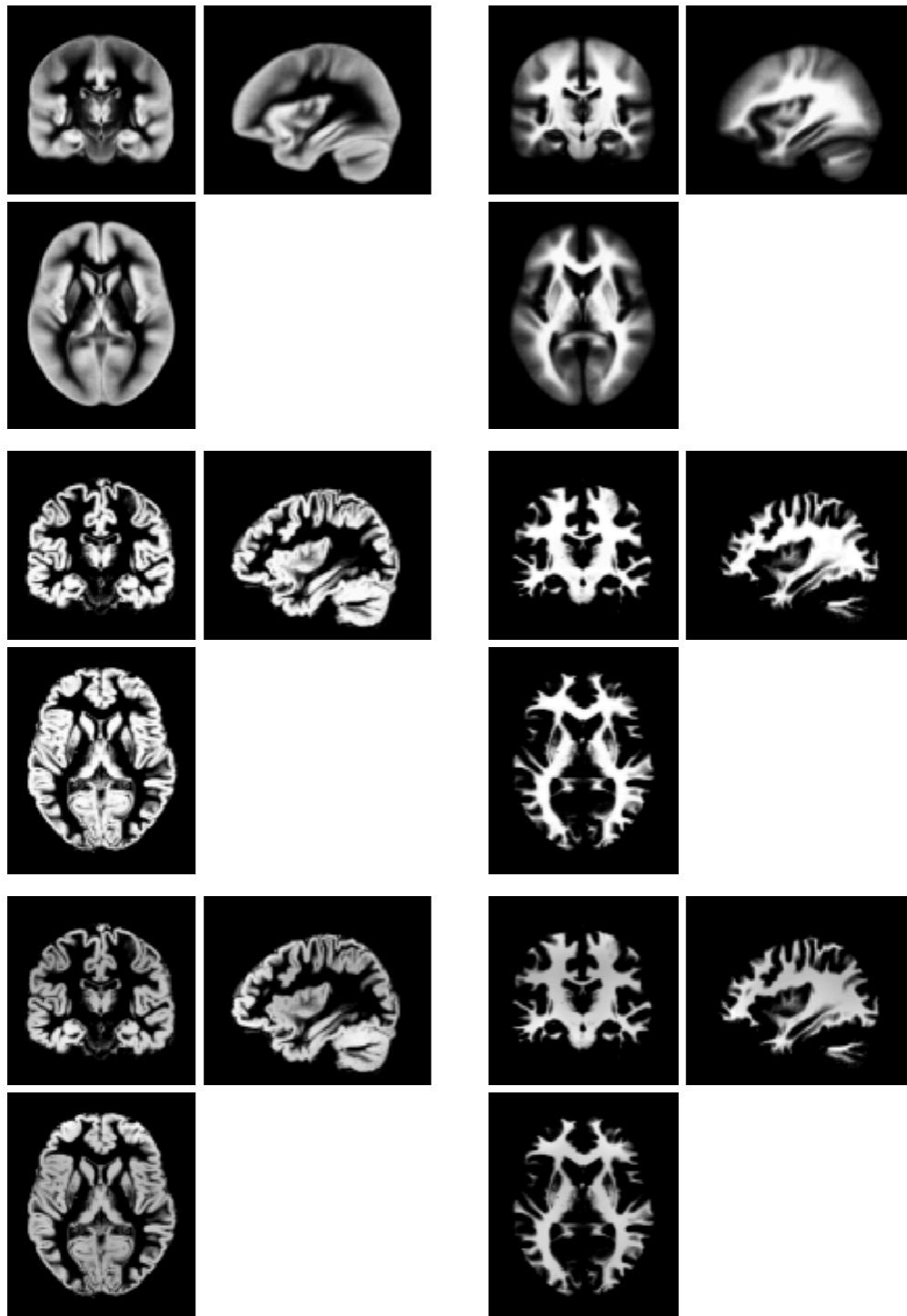


Figure 29.2: Segmentation results. These are the spatially normalised results that can be obtained (note that CSF data is not shown). Top row: The tissue probability maps used to guide the segmentation. Middle row: Spatially normalised tissue maps of grey and white matter (wc1X.img and wc2X.img). Bottom row: Modulated spatially normalised tissue maps of grey and white matter (mwc1X.img and mwc2X.img).

$$\begin{pmatrix} \frac{dx'}{dx} & \frac{dx'}{dy} & \frac{dx'}{dz} \\ \frac{dy'}{dx} & \frac{dy'}{dy} & \frac{dy'}{dz} \\ \frac{dz'}{dx} & \frac{dz'}{dy} & \frac{dz'}{dz} \end{pmatrix}$$

The value of  $dx'/dy$  is a measure of how much  $x'$  changes if  $y$  is changed by a tiny amount. The determinant of the Jacobian is the measure of relative volumes of warped and unwarped structures. The modulation step simply involves multiplying by the relative volumes (see Figure 29.2).

### 29.2.1 Grey Matter

Options to produce grey matter images: `c1*`, `wc1*` and `mwc1*`.

### 29.2.2 White Matter

Options to produce white matter images: `c2*`, `wc2*` and `mwc2*`.

### 29.2.3 Cerebro-Spinal Fluid

Options to produce CSF images: `c3*`, `wc3*` and `mwc3*`.

### 29.2.4 Bias Corrected

This is the option to produce a bias corrected version of your image. MR images are usually corrupted by a smooth, spatially varying artifact that modulates the intensity of the image (bias). These artifacts, although not usually a problem for visual inspection, can impede automated processing of the images. The bias corrected version should have more uniform intensities within the different types of tissues.

### 29.2.5 Clean up any partitions

This uses a crude routine for extracting the brain from segmented images. It begins by taking the white matter, and eroding it a couple of times to get rid of any odd voxels. The algorithm continues on to do conditional dilations for several iterations, where the condition is based upon gray or white matter being present. This identified region is then used to clean up the grey and white matter partitions, and has a slight influence on the CSF partition.

If you find pieces of brain being chopped out in your data, then you may wish to disable or tone down the cleanup procedure.

## 29.3 Custom

Various options can be adjusted in order to improve the performance of the algorithm with your data. Knowing what works best should be a matter of empirical exploration. For example, if your data has very little intensity non-uniformity artifact, then the bias regularisation should be increased. This effectively tells the algorithm that there is very little bias in your data, so it does not try to model it.

### 29.3.1 Tissue probability maps

Select the tissue probability images. These should be maps of grey matter, white matter and cerebro-spinal fluid probability. A nonlinear deformation field is estimated that best overlays the tissue probability maps on the individual subjects' image. The default tissue probability maps are modified versions of the ICBM Tissue Probabilistic Atlases. These tissue probability maps are kindly provided by the International Consortium for Brain Mapping, John C. Mazziotta and Arthur W. Toga. [http://www.loni.ucla.edu/ICBM/ICBM\\_TissueProb.html](http://www.loni.ucla.edu/ICBM/ICBM_TissueProb.html). The original data

are derived from 452 T1-weighted scans, which were aligned with an atlas space, corrected for scan inhomogeneities, and classified into grey matter, white matter and cerebrospinal fluid. These data were then affine registered to the MNI space and downsampled to 2mm resolution.

Rather than assuming stationary prior probabilities based upon mixing proportions, additional information is used, based on other subjects' brain images. Priors are usually generated by registering a large number of subjects together, assigning voxels to different tissue types and averaging tissue classes over subjects. Three tissue classes are used: grey matter, white matter and cerebro-spinal fluid. A fourth class is also used, which is simply one minus the sum of the first three. These maps give the prior probability of any voxel in a registered image being of any of the tissue classes - irrespective of its intensity.

The model is refined further by allowing the tissue probability maps to be deformed according to a set of estimated parameters. This allows spatial normalisation and segmentation to be combined into the same model. This implementation uses a low-dimensional approach, which parameterises the deformations by a linear combination of about a thousand cosine transform bases. This is not an especially precise way of encoding deformations, but it can model the variability of overall brain shape. Evaluations by Hellier et al have shown that this simple model can achieve a registration accuracy comparable to other fully automated methods with many more parameters.

### 29.3.2 Gaussians per class

The number of Gaussians used to represent the intensity distribution for each tissue class can be greater than one. In other words, a tissue probability map may be shared by several clusters. The assumption of a single Gaussian distribution for each class does not hold for a number of reasons. In particular, a voxel may not be purely of one tissue type, and instead contain signal from a number of different tissues (partial volume effects). Some partial volume voxels could fall at the interface between different classes, or they may fall in the middle of structures such as the thalamus, which may be considered as being either grey or white matter. Various other image segmentation approaches use additional clusters to model such partial volume effects. These generally assume that a pure tissue class has a Gaussian intensity distribution, whereas intensity distributions for partial volume voxels are broader, falling between the intensities of the pure classes. Unlike these partial volume segmentation approaches, the model adopted here simply assumes that the intensity distribution of each class may not be Gaussian, and assigns belonging probabilities according to these non-Gaussian distributions. Typical numbers of Gaussians could be two for grey matter, two for white matter, two for CSF, and four for everything else.

### 29.3.3 Affine Regularisation

The procedure is a local optimisation, so it needs reasonable initial starting estimates. Images should be placed in approximate alignment using the Display function of SPM before beginning. A Mutual Information affine registration with the tissue probability maps (D'Agostino et al, 2004) is used to achieve approximate alignment. Note that this step does not include any model for intensity non-uniformity. This means that if the procedure is to be initialised with the affine registration, then the data should not be too corrupted with this artifact. If there is a lot of intensity non-uniformity, then manually position your image in order to achieve closer starting estimates, and turn off the affine registration.

Affine registration into a standard space can be made more robust by regularisation (penalising excessive stretching or shrinking). The best solutions can be obtained by knowing the approximate amount of stretching that is needed (e.g. ICBM templates are slightly bigger than typical brains, so greater zooms are likely to be needed). For example, if registering to an image in ICBM/MNI space, then choose this option. If registering to a template that is close in size, then select the appropriate option for this.

### 29.3.4 Warping Regularisation

The objective function for registering the tissue probability maps to the image to process, involves minimising the sum of two terms. One term gives a function of how probable the data

is given the warping parameters. The other is a function of how probable the parameters are, and provides a penalty for unlikely deformations. Smoother deformations are deemed to be more probable. The amount of regularisation determines the tradeoff between the terms. Pick a value around one. However, if your normalised images appear distorted, then it may be an idea to increase the amount of regularisation (by an order of magnitude). More regularisation gives smoother deformations, where the smoothness measure is determined by the bending energy of the deformations.

### 29.3.5 Warp Frequency Cutoff

Cutoff of DCT bases. Only DCT bases of periods longer than the cutoff are used to describe the warps. The number actually used will depend on the cutoff and the field of view of your image. A smaller cutoff frequency will allow more detailed deformations to be modelled, but unfortunately comes at a cost of greatly increasing the amount of memory needed, and the time taken.

### 29.3.6 Bias regularisation

MR images are usually corrupted by a smooth, spatially varying artifact that modulates the intensity of the image (bias). These artifacts, although not usually a problem for visual inspection, can impede automated processing of the images.

An important issue relates to the distinction between intensity variations that arise because of bias artifact due to the physics of MR scanning, and those that arise due to different tissue properties. The objective is to model the latter by different tissue classes, while modelling the former with a bias field. We know a priori that intensity variations due to MR physics tend to be spatially smooth, whereas those due to different tissue types tend to contain more high frequency information. A more accurate estimate of a bias field can be obtained by including prior knowledge about the distribution of the fields likely to be encountered by the correction algorithm. For example, if it is known that there is little or no intensity non-uniformity, then it would be wise to penalise large values for the intensity non-uniformity parameters. This regularisation can be placed within a Bayesian context, whereby the penalty incurred is the negative logarithm of a prior probability for any particular pattern of non-uniformity.

### 29.3.7 Bias FWHM

FWHM of Gaussian smoothness of bias. If your intensity non-uniformity is very smooth, then choose a large FWHM. This will prevent the algorithm from trying to model out intensity variation due to different tissue types. The model for intensity non-uniformity is one of i.i.d. Gaussian noise that has been smoothed by some amount, before taking the exponential. Note also that smoother bias fields need fewer parameters to describe them. This means that the algorithm is faster for smoother intensity non-uniformities.

### 29.3.8 Sampling distance

The approximate distance between sampled points when estimating the model parameters. Smaller values use more of the data, but the procedure is slower.

### 29.3.9 Masking image

The segmentation can be masked by an image that conforms to the same space as the images to be segmented. If an image is selected, then it must match the image(s) voxel-for voxel, and have the same voxel-to-world mapping. Regions containing a value of zero in this image do not contribute when estimating the various parameters.



## Part VII

# Data sets and examples





## Chapter 30

# Auditory fMRI data

This experiment was conducted by Geraint Rees under the direction of Karl Friston and the FIL methods group. The purpose was to explore equipment and techniques in the early days of our fMRI experience. As such, it has not been formally written up, and is freely available for personal education and evaluation purposes.

This data set was the first ever collected and analysed in the Functional Imaging Laboratory (FIL) and is known locally as the mother of all experiments (MoAE).

This data set comprises whole brain BOLD/EPI images acquired on a modified 2T Siemens MAGNETOM Vision system. Each acquisition consisted of 64 contiguous slices ( $64 \times 64 \times 64$   $3 \times 3 \times 3$  mm<sup>3</sup> voxels). Acquisition took 6.05s, with the scan to scan repeat time (TR) set arbitrarily to 7s.

96 acquisitions were made (TR=7s) from a single subject, in blocks of 6, giving 16 42s blocks. The condition for successive blocks alternated between rest and auditory stimulation, starting with rest. Auditory stimulation was bi-syllabic words presented binaurally at a rate of 60 per minute. The functional data starts at acquisition 4, image `fm00223_004.{hdr,img}`, and are stored in folder `fm00223`. Due to T1 effects it is advisable to discard the first few scans (there were no “dummy” lead-in scans). A structural image was also acquired: `sm00223_002.{hdr,img}`, stored in folder `sm00223`. These images are stored in Analyze format (now superseded by the NIfTI format, but SPM reads natively both formats and always saves images as NIfTI) and are available from the SPM site <sup>1</sup>.

To analyse the data, first create a new directory DIR, eg. `C:\data\auditory`, in which to place the results of your analysis. Then create 3 subdirectories (i) `dummy`, (ii) `jobs` and (iii) `classical`. As the analysis proceeds these directories will be filled with dummy scans, job-specification files, design matrices and models estimated using classical inference.

Start up MATLAB enter your `jobs` directory and type `spm_fmri` at the MATLAB prompt. SPM will then open in fMRI mode with three windows (see Figure 30.1): (1) the top-left or “Menu” window, (2) the bottom-left or “Interactive” window and (3) the right-hand or “Graphics” window. Analysis then takes place in three major stages (i) spatial pre-processing, (ii) model specification, review and estimation and (iii) inference. These stages organise the buttons in SPM’s Menu window.

### 30.1 Preamble (dummy scans)

To avoid T1 effects in the initial scans of an fMRI time series we recommend discarding the first few scans. To make this example simple, we’ll discard the first complete cycle (12 scans, 04-15), leaving 84 scans, image files 16-99. This is best done by moving these files to a different directory, `dummy`, that we created earlier.

---

<sup>1</sup>Auditory fMRI dataset: <http://www.fil.ion.ucl.ac.uk/spm/data/auditory/>

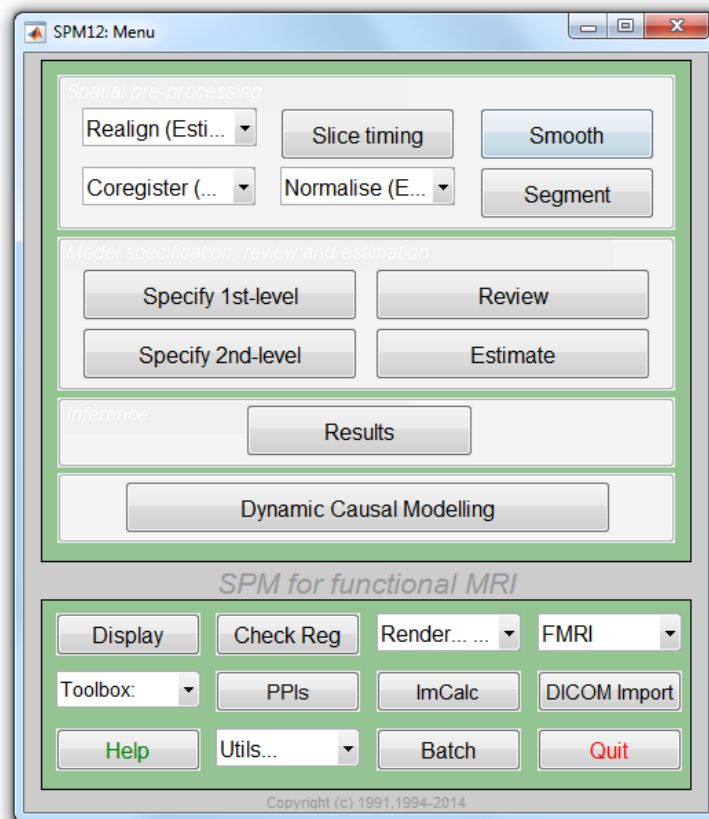


Figure 30.1: The SPM base window comprises three sections *i*) spatial pre-processing, *(ii)* model specification, review and estimation and *(iii)* inference.

## 30.2 Spatial pre-processing

### 30.2.1 Realignment

Under the spatial pre-processing section of the SPM Menu window select **REALIGN (EST & RES)** from the **REALIGN** pulldown menu. This will call up a realignment job specification in the batch editor. Then

- Highlight “Data”, select “New Session”, then highlight the newly created “Session” option.
- Press “Select Files” and use the SPM file selector to choose all of the functional images eg. (“fM000\*.img”). There should be 84 files.
- Press “Resliced images” in the “Reslice Options” and select “Mean Image Only”.
- Save the job file as eg. DIR\jobs\realign.mat.
- Press the **RUN** button in the batch editor (green arrow).

This will run the realign job which will estimate the 6 parameter (rigid body) spatial transformation that will align the times series of images and will modify the header of the input images (\*.hdr), such that they reflect the relative orientation of the data after correction for movement artefacts. SPM will then plot the estimated time series of translations and rotations shown in Figure 30.2. These data are also saved to a file eg. rp\_fm00223\_016.txt, so that these variables can be later used as regressors when fitting GLMs. This allows movements effects to be discounted when looking for brain activations.

SPM will also create a mean image eg. meanfm00223\_016.img which will be used in the next step of spatial processing - coregistration.

### 30.2.2 Coregistration

Select **COREGISTER (ESTIMATE)** from the **COREGISTER** pulldown. This will call up the specification of a coregistration job in the batch editor.

- Highlight “Reference Image” and then select the mean fMRI scan from realignment eg. meanfm00223\_016.img.
- Highlight “Source Image” and then select the structural image eg. sM00223\_002.img.
- Press the **Save** button and save the job as DIR\jobs\coregister.mat.
- Then press the **RUN** button.

SPM will then implement a coregistration between the structural and functional data that maximises the mutual information. The image in figure 30.3 should then appear in the Graphics window. SPM will have changed the header of the source file which in this case is the structural image sM00223\_002.hdr.

The **CHECK REG** facility is useful here, to check the results of coregistration. Press the **CHECK REG** button in the lower section of the Menu window and then select the “Reference” and “Source” Images specified above ie meanfm00223\_016.img and sM00223\_002.img. SPM will then produce an image like that shown in Figure 30.4 in the Graphics window. You can then use your mouse to navigate these images to confirm that there is an anatomical correspondence.

### 30.2.3 Segmentation

Press the **SEGMENT** button. This will call up the specification of a segmentation job in the batch editor. Highlight the “Volumes” field and then select the subject’s registered anatomical image eg. sM00223\_002.img. Highlight “Save Bias Corrected” and select “Save Bias Corrected”. Highlight “Deformation Fields” (the bottom of the list) and select “Forward”. Save the job file as **segment.mat** and then press **RUN**. SPM will segment the structural image using the default tissue probability maps as priors.

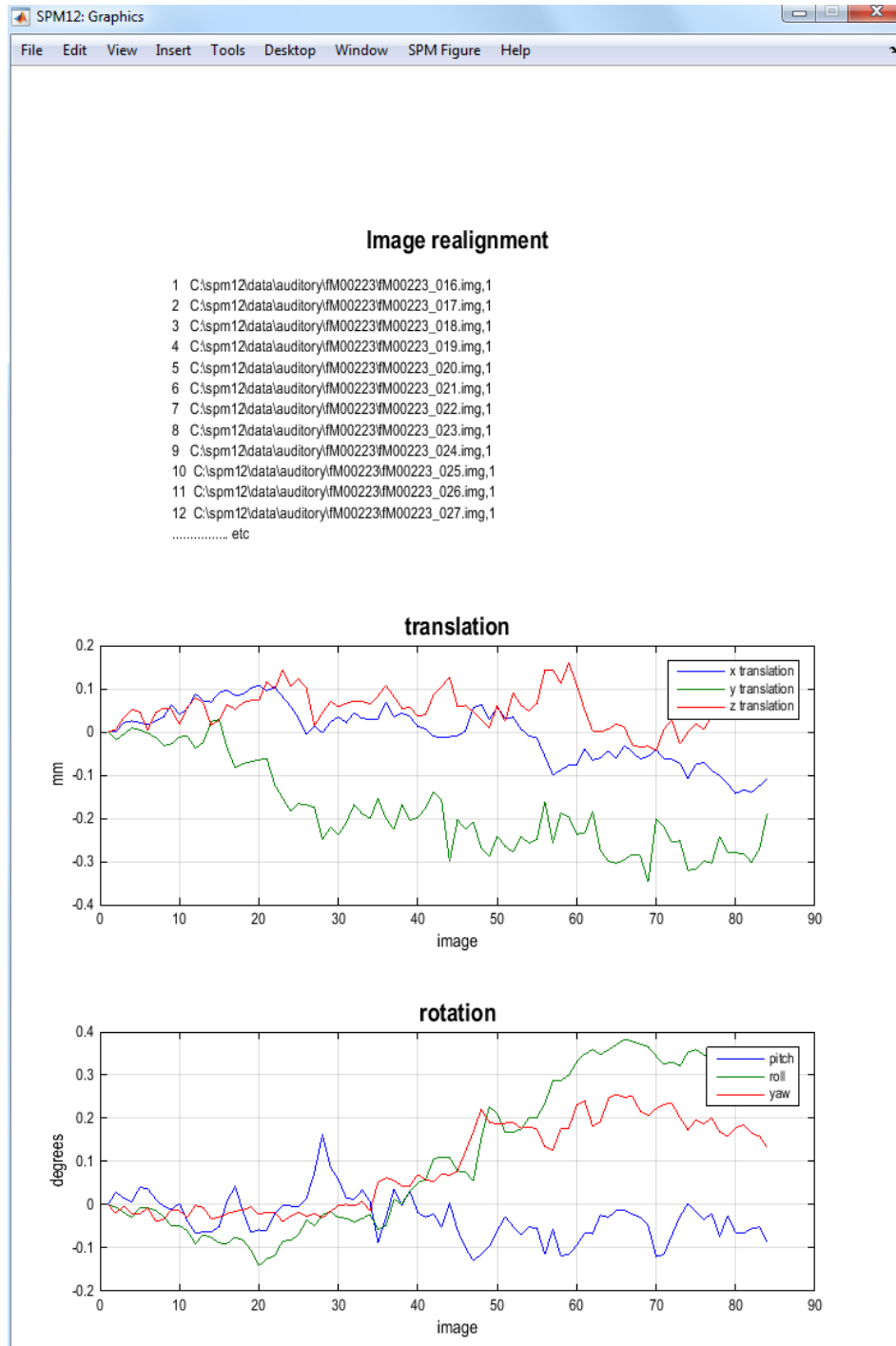
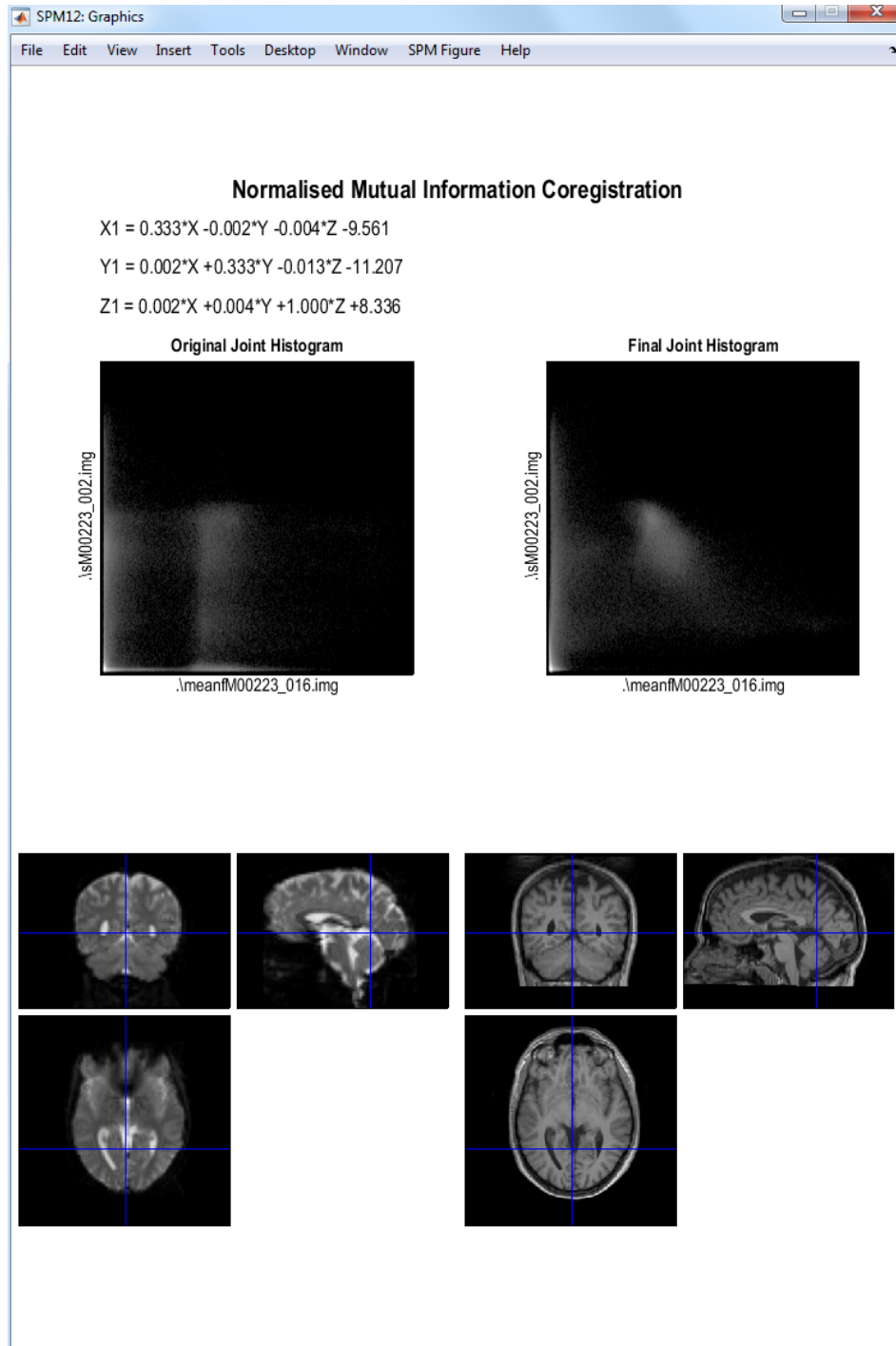


Figure 30.2: Realignment of Auditory data.

Figure 30.3: *Mutual Information Coregistration of Auditory data.*

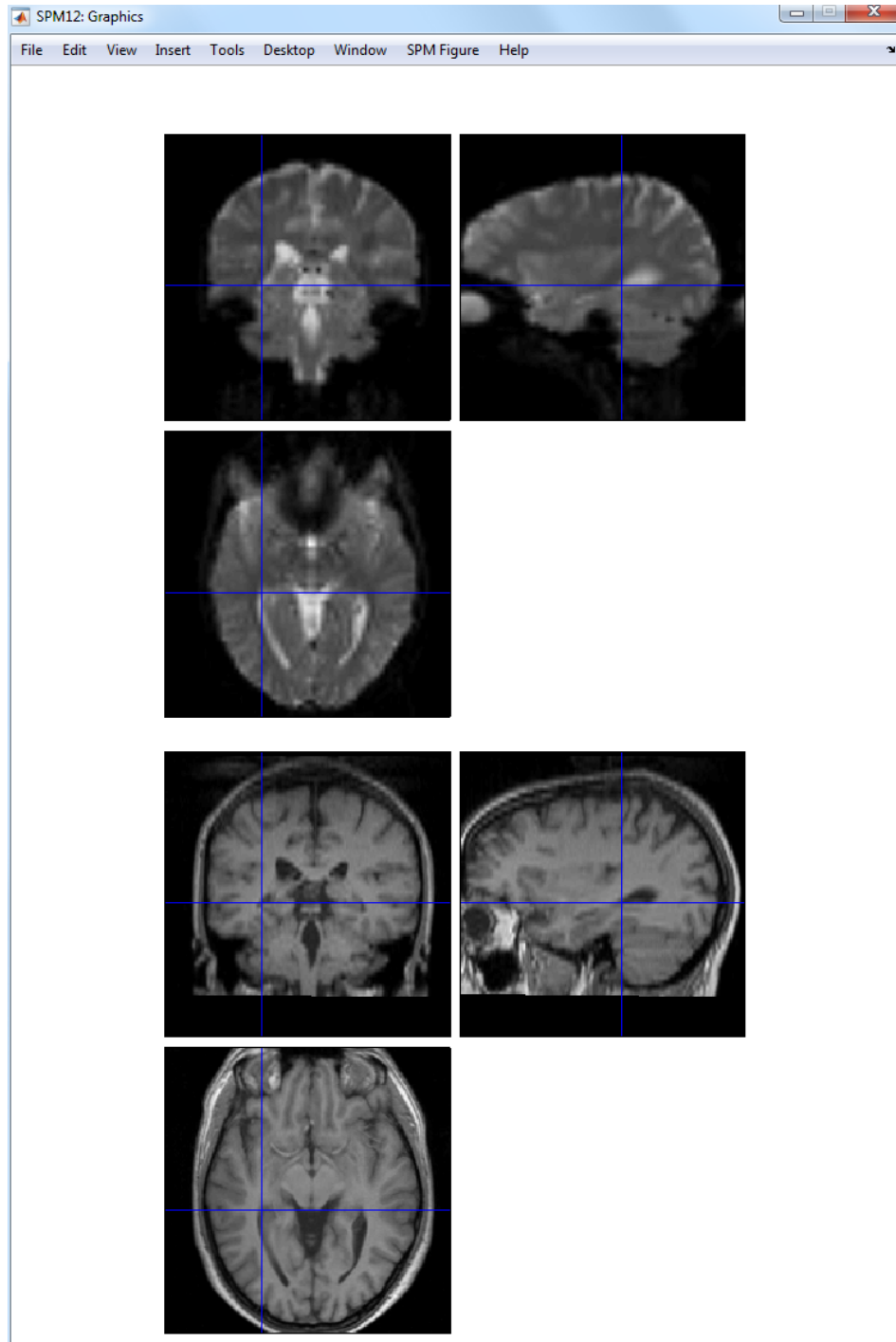


Figure 30.4: *Checking registration of functional and “registered” structural data.*

SPM will create gray and white matter images and bias-field corrected structural image. These can be viewed using the CHECKREG facility as described in the previous section. Figure 30.5 shows the gray matter image, `c1sM0023_002.nii` along with the original structural. Figure 30.6 shows the structural and bias-corrected image, `msM0023_002.nii`.

SPM will also write a deformation field, file `y_sM00223_002.nii` in the original structural directory. It contains 3 volumes to encode the x, y and z coordinates. Given that the structural and functional data are in alignment, this can be used to spatially normalise the functional data.

### 30.2.4 Normalise

Select NORMALISE (WRITE) from the NORMALISE pulldown menu. This will call up the specification of a normalise job in the batch editor.

- Highlight “Data”, select New “Subject”,
- Highlight “Deformation Field” and select the `y_sM00223_002.nii` file that you created in the previous section,
- Highlight “Images to Write” and select all of the realigned functional images `fM000*.img`. You can right click over the listed files, choose “Select all” and press “Done”.
- In the “Writing Options”, change “Voxel sizes” from [2 2 2] to [3 3 3]. This step is not strictly necessary: it will write images out at a resolution closer to that at which they were acquired.
- Press “Save”, save the job as `normalise_functional.mat` and then press the RUN button.

SPM will then write spatially normalised files to the functional data directory. These files have the prefix `w`.

If you wish to superimpose a subject’s functional activations on their own anatomy<sup>2</sup> you will also need to apply the spatial normalisation parameters to their (bias-corrected) anatomical image. To do this

- Select NORMALISE (WRITE), highlight “Data”, select “New Subject”.
- Highlight “Deformation Field”, select the `y_sM00223_002.nii` file that you created in the previous section, press “Done”.
- Highlight “Images to Write”, select the bias-corrected structural eg. `msM00223_002.nii`, press “Done”.
- Open “Writing Options”, select voxel sizes and change the default [2 2 2] to [1 1 3] which corresponds to the original resolution of the images.
- Save the job as `normalise_structural.mat` and press the RUN button.

### 30.2.5 Smoothing

Press the SMOOTH button. This will call up the specification of a smooth job in the batch editor.

- Select “Images to Smooth” and then select the spatially normalised files created in the last section eg. `wf*.img`. This can be done efficiently by changing the filter in the SPM file selector to `~wf.*`. SPM will then only list those files beginning with letters `wf` ie. those that have been spatially normalised.
- Highlight “FWHM” and change [8 8 8] to [6 6 6]. This will smooth the data by 6mm in each direction.
- Save the job as `smooth.mat` and press the Run button.

An example of functional image and its smoothed version is displayed on Figure 30.7.

<sup>2</sup>Beginners may wish to skip this step, and instead just superimpose functional activations on an “average structural image”.

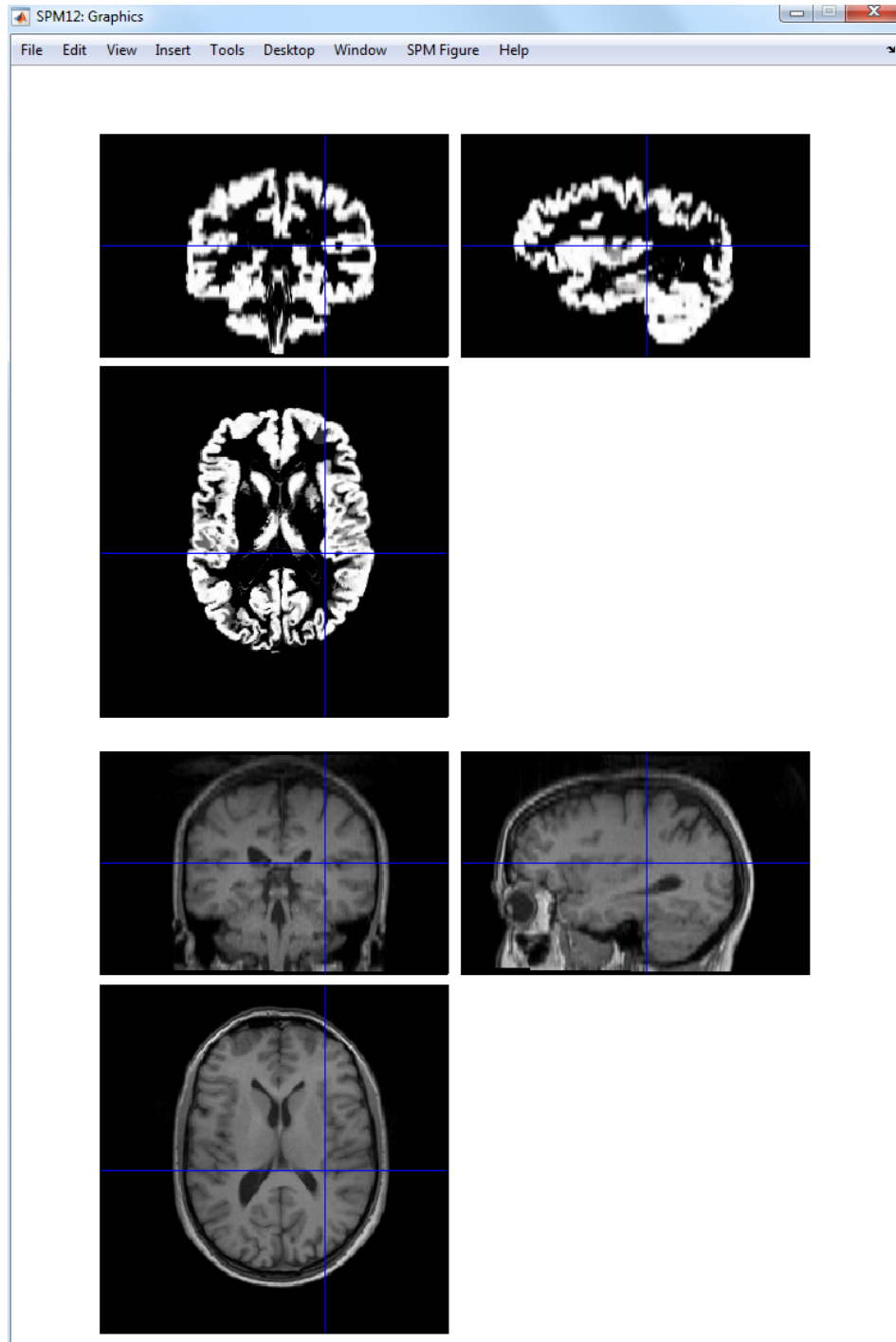


Figure 30.5: *Gray matter image and “registered” structural image.*



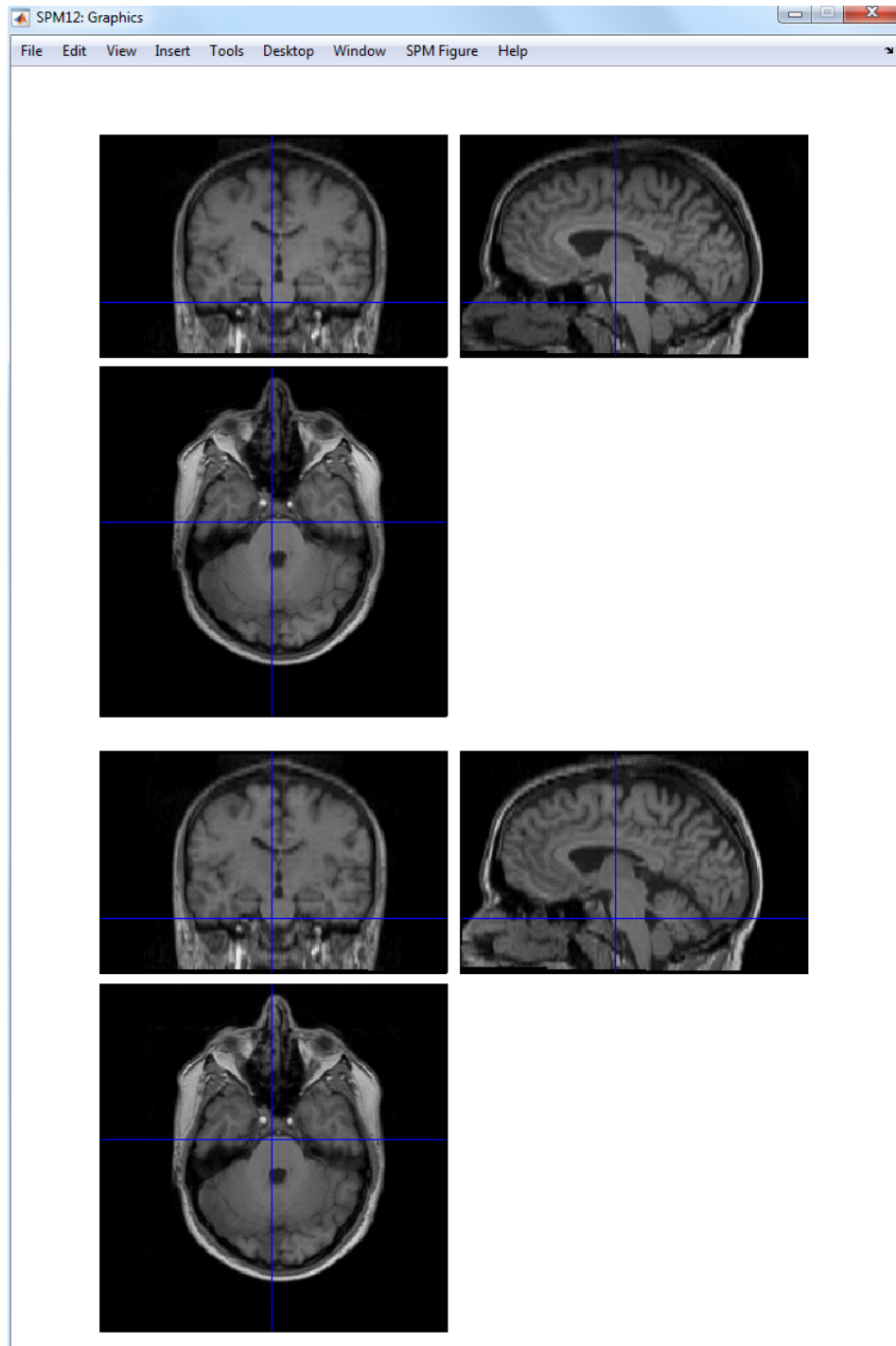


Figure 30.6: *Structural image (top) and bias-corrected structural image (bottom). Notice that the original structural is darker at the top than at the bottom. This non-uniformity has been removed in the bias-corrected image.*

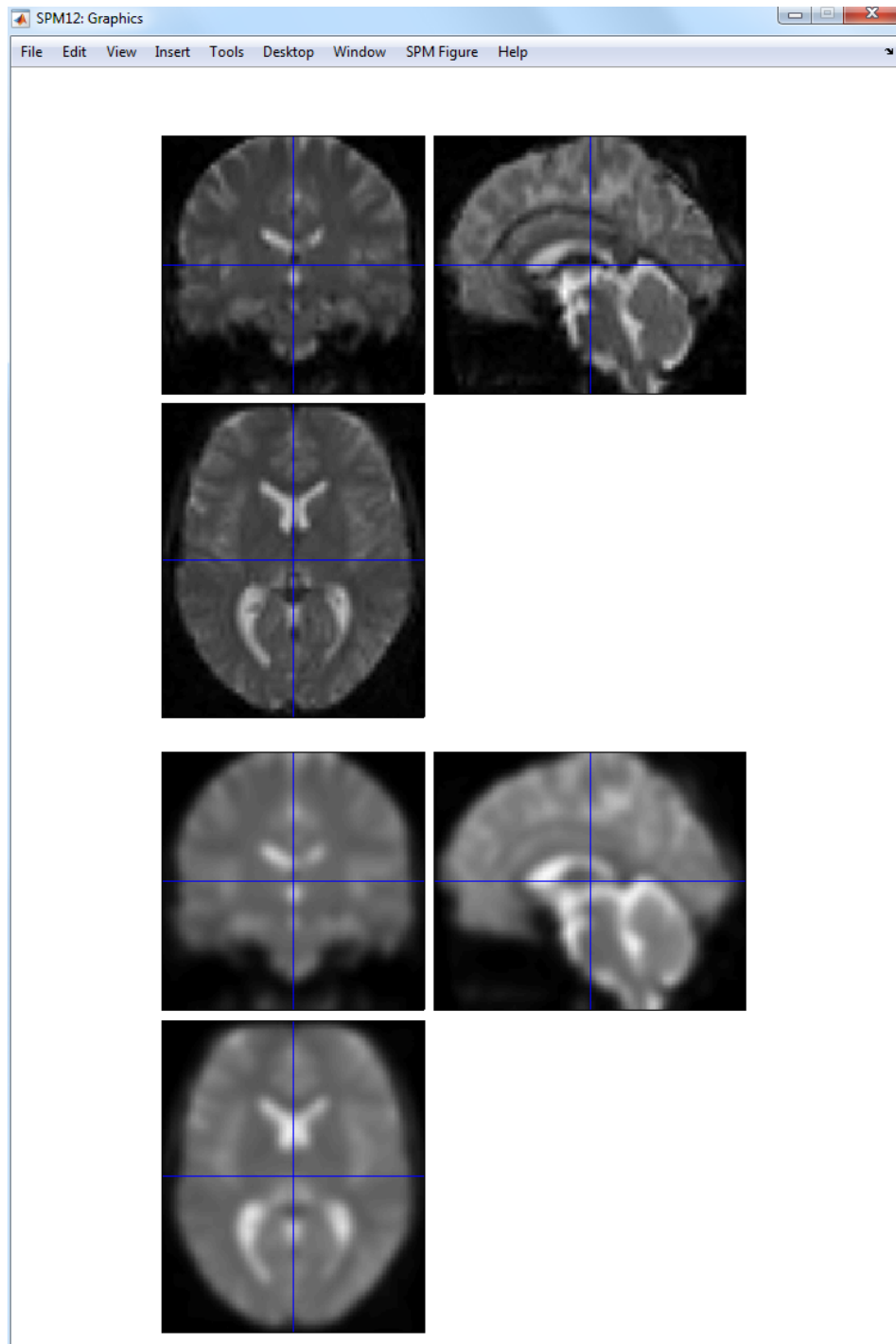


Figure 30.7: *Functional image (top) and 6mm-smoothed functional image (bottom). These images were obtained using SPM's "CheckReg" facility.*

## 30.3 Model specification, review and estimation

Press the “Specify 1st-level” button. This will call up the specification of an fMRI specification job in the batch editor. Then

- Open the “Timing parameters” option.
- Highlight “Units for design” and select “Scans”.
- Highlight “Inter-scan interval” and enter 7. That’s the TR in seconds.
- Highlight “Data and Design” and select “New Subject/Session”. Then open the newly created “Subject/Session” option.
- Highlight “Scans” and use SPM’s file selector to choose the 84 smoothed, normalised functional images ie `swfM00223_016.img` to `swfM00223_099.img`. These can be selected easily using the `^sw.*` filter, and select all. Then press “Done”.
- Highlight “Condition” and select “New condition”.
- Open the newly created “Condition” option. Highlight “Name” and enter “listening”. Highlight “Onsets” and enter “6:12:84”. Highlight “Durations” and enter “6”.
- Highlight “Directory” and select the `DIR/classical` directory you created earlier.
- Save the job as `specify.mat` and press the Run button.

SPM will then write an `SPM.mat` file to the `DIR/classical` directory. It will also plot the design matrix, as shown in Figure 30.8.

At this stage it is advisable to check your model specification using SPM’s review facility which is accessed via the “Review” button. This brings up a “design” tab on the interactive window clicking on which produces a pulldown menu. If you select the first item “Design Matrix” SPM will produce the image shown in Figure 30.8. If you select “Explore” then “Session 1” then “listening”, SPM will produce the plots shown in Figure 30.9.

If you select the second item on the “Design” tab, “Design Orthogonality”, SPM will produce the plot shown in Figure 30.10. Columns  $x_1$  and  $x_2$  are orthogonal if the inner product  $x_1^T x_2 = 0$ . The inner product can also be written  $x_1^T x_2 = |x_1||x_2|\cos\theta$  where  $|x|$  denotes the length of  $x$  and  $\theta$  is the angle between the two vectors. So, the vectors will be orthogonal if  $\cos\theta = 0$ . The upper-diagonal elements in the matrix at the bottom of figure 30.10 plot  $\cos\theta$  for each pair of columns in the design matrix. Here we have a single entry. A degree of non-orthogonality or collinearity is indicated by the gray shading.

### 30.3.1 Estimate

Press the ESTIMATE button. This will call up the specification of an fMRI estimation job in the batch editor. Then

- Highlight the “Select SPM.mat” option and then choose the `SPM.mat` file saved in the `classical` subdirectory.
- Save the job as `estimate.mat` and press the Run button.

SPM will write a number of files into the selected directory including an `SPM.mat` file.

## 30.4 Inference

After estimation:

- Press “Results”.
- Select the `SPM.mat` file created in the last section.

This will invoke the contrast manager.

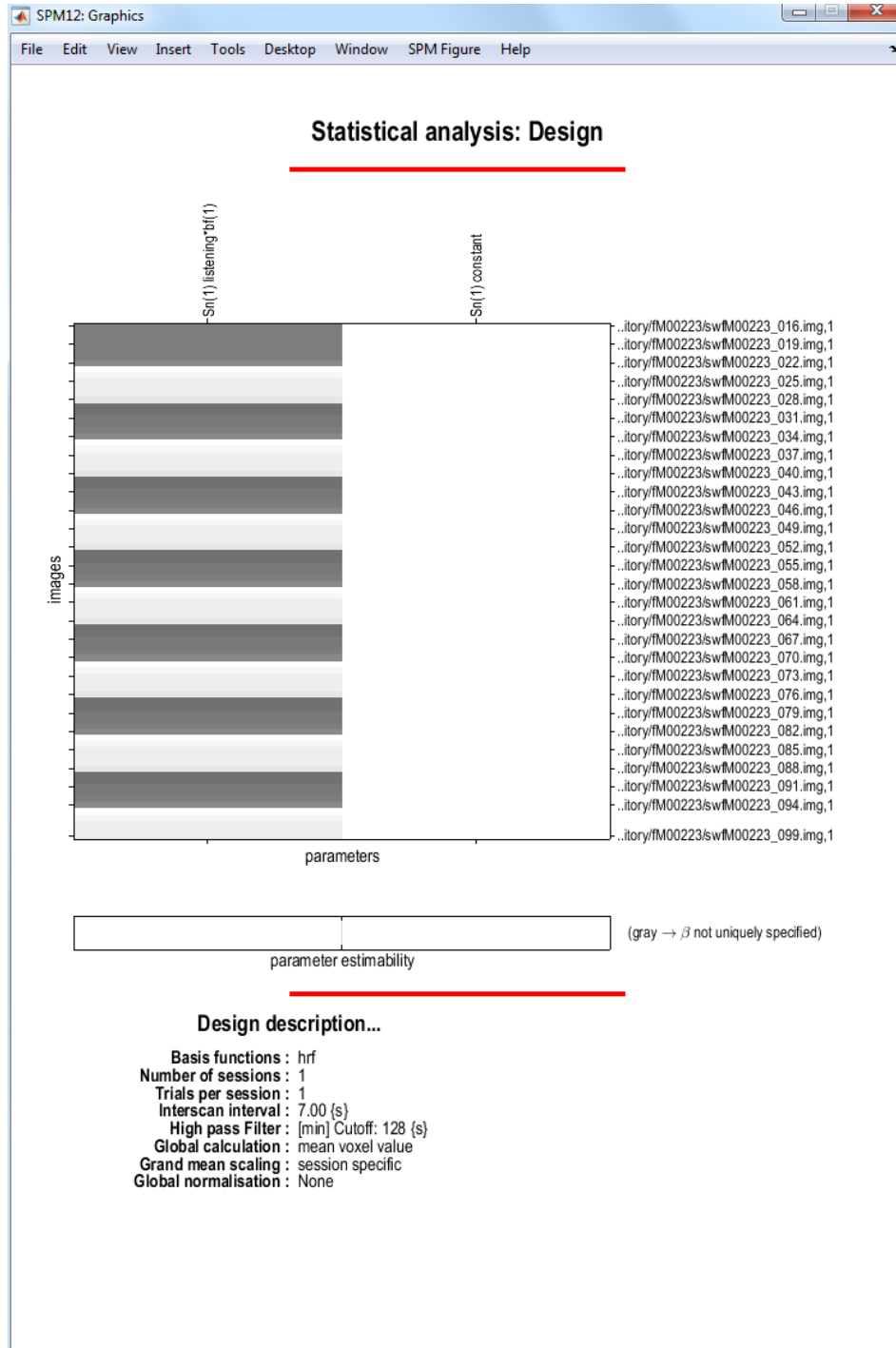


Figure 30.8: *Design matrix*: The filenames on the right-hand side of the design matrix indicate the scan associated with each row.

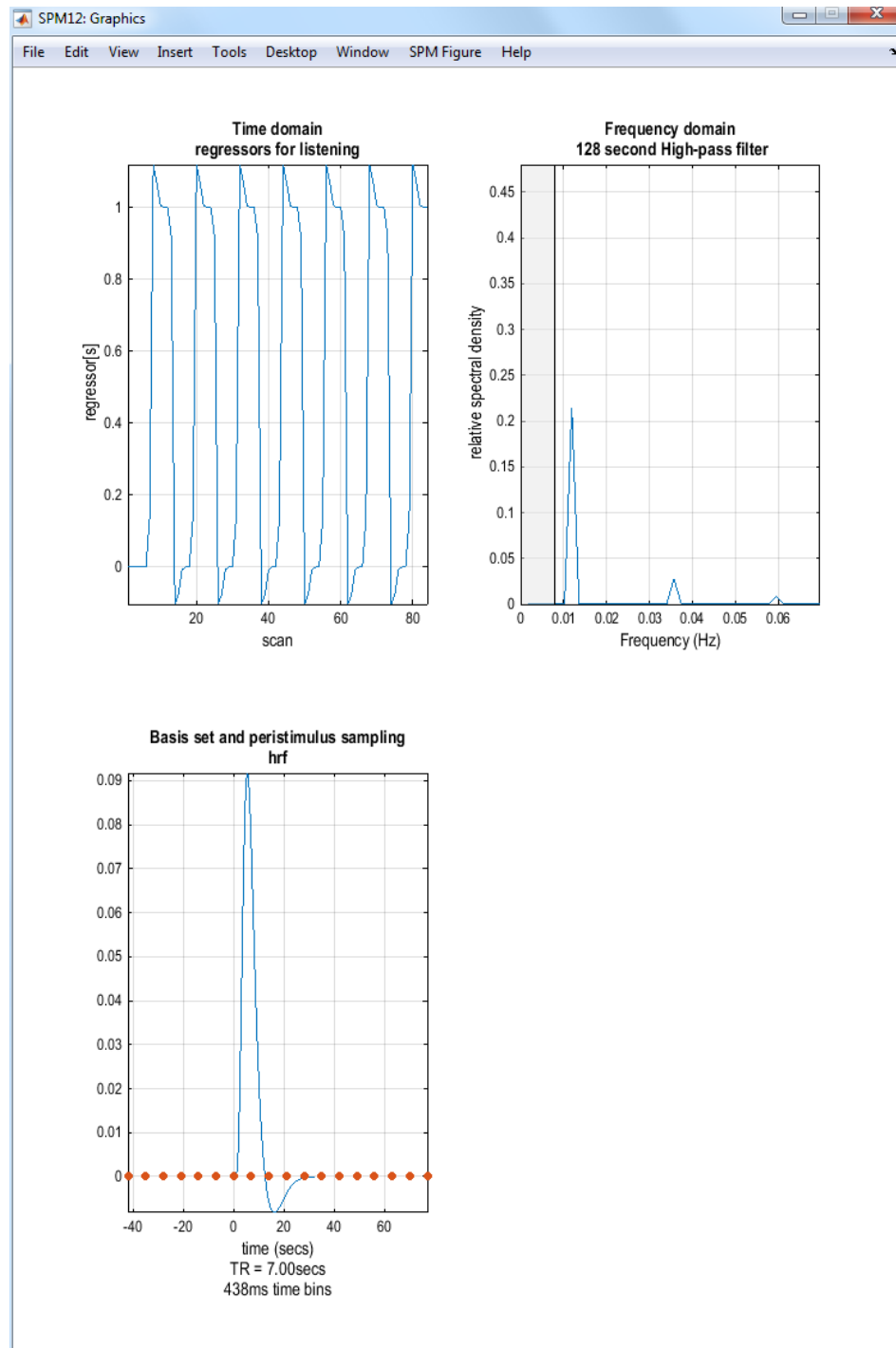


Figure 30.9: *Exploring the design matrix in Figure 30.8: This shows the time series of the “listening” regressor (top left), a frequency domain plot of the “listening” regressor (top right) and the basis function used to convert assumed neuronal activity into hemodynamic activity. In this model we used the default option - the canonical basis function. The frequency domain plot shows that the frequency content of the “listening” regressor is above the set frequencies that are removed by the High Pass Filter (HPF) (these are shown in gray - in this model we accepted the default HPF cut-off of 128s or 0.008Hz).*

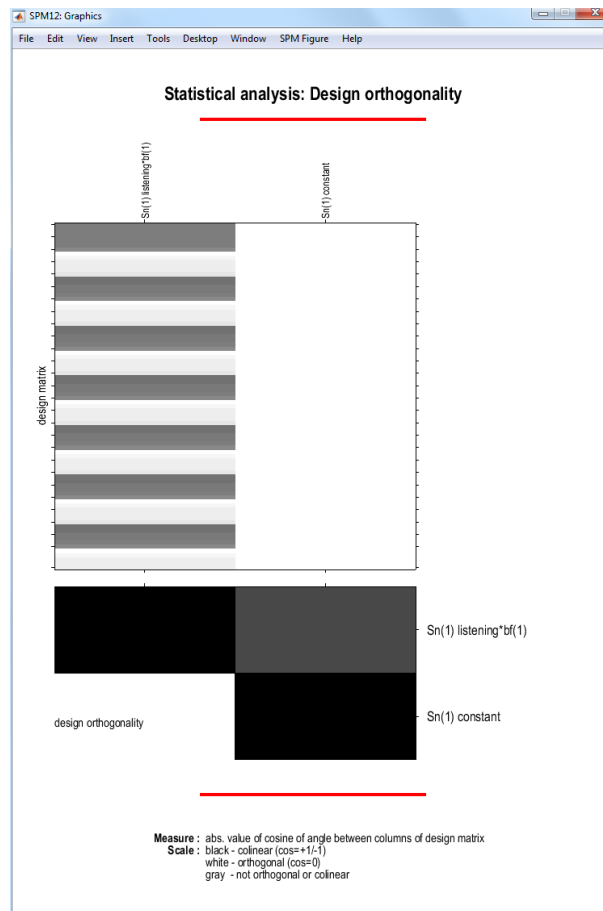


Figure 30.10: *Design Orthogonality*: The description above the first column in the design matrix  $S_n(1)\text{Listening}*\text{bf}(1)$  means that this column refers to the first session of data (in this analysis there is only 1 session), the name of this condition/trial is 'listening' and the trial information has been convolved with the first basis function (the canonical hemodynamic response). The constant regressor for session 1 is referred to as  $S_n(1)\text{Constant}$ . The orthogonality matrix at the bottom indicates a degree of collinearity between regressors.

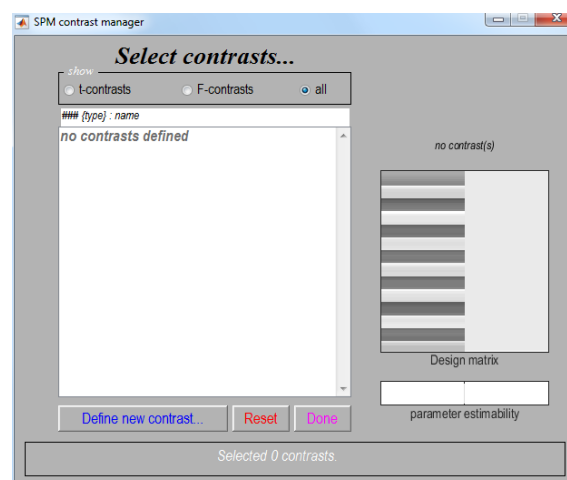


Figure 30.11: *The contrast manager*

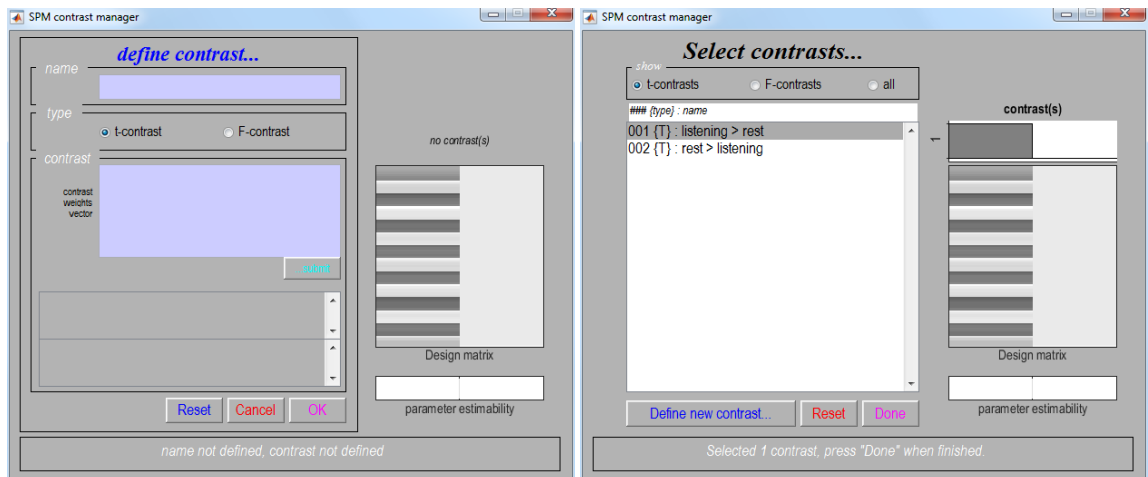


Figure 30.12: *Left: A contrast is entered by specifying the numeric values in the lower window and the name in the upper window. Right: After contrasts have been specified they can be selected.*

### 30.4.1 Contrast manager

The contrast manager displays the design matrix (surfable) in the right panel and lists specified contrasts in the left panel. Either “t-contrast” or “F-contrast” can be selected. To examine statistical results for condition effects

- Select “Define new contrast”

One sided main effects for the listening condition (i.e., a one-sided t-test) can be specified (in this example) as “1” (listening > rest) and “-1” (rest > listening). SPM will accept estimable contrasts only. Accepted contrasts are displayed at the bottom of the contrast manager window in green, incorrect ones are displayed in red. To view a contrast

- Select the contrast name e.g., “listening > rest”.
- Press “Done”.

### 30.4.2 Masking

You will then be prompted with

- *Apply masking ? [none/contrast/image].*
- “Specify none”.

Masking implies selecting voxels specified by other contrasts. If “yes”, SPM will prompt for (one or more) masking contrasts, the significance level of the mask (default  $p = 0.05$  uncorrected), and will ask whether an inclusive or exclusive mask should be used. Exclusive will remove all voxels which reach the default level of significance in the masking contrast, inclusive will remove all voxels which do not reach the default level of significance in the masking contrast. Masking does not affect  $p$ -values of the “target” contrast, it only includes or excludes voxels.

### 30.4.3 Thresholds

You will then be prompted with

- *p value adjustment to control: [FWE/none].*
  - Select “FWE”.
- *p value(family-wise error).*

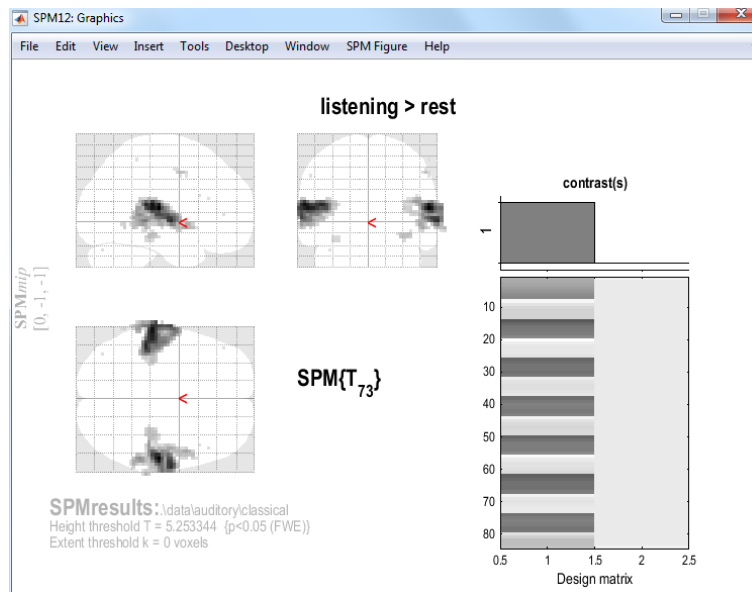


Figure 30.13: *SPM showing bilateral activation of auditory cortex.*

- Accept the default value, 0.05.

A Family Wise Error (FWE) is a false positive anywhere in the SPM. Now, imagine repeating your experiment many times and producing SPMs. The proportion of SPMs containing FWEs is the FWE rate. A value of 0.05 implies that on average 1 in 20 SPMs contains one or more false positives somewhere in the image.

If you choose the “none” option above this corresponds to making statistical inferences at the “voxel level”. These use “uncorrected”  $p$  values, whereas FWE thresholds are said to use “corrected”  $p$ -values. SPM’s default uncorrected  $p$ -value is  $p=0.001$ . This means that the probability of a false positive at each voxel is 0.001. So if, you have 50,000 voxels you can expect  $50,000 \times 0.001 = 50$  false positives in each SPM.

You will then be prompted with

- *Extent Threshold {voxels} [0].*

- Accept the default value, “0”.

Entering a value  $k$  here will produce SPMs with clusters containing at least  $k$  voxels. SPM will then produce the SPM shown in Figure 30.13.

#### 30.4.4 Files

A number of files are written to the working directory at this time. Images containing weighted parameter estimates are saved as `con_0001.nii`, `con_0002.nii`, etc. in the working directory. Images of T-statistics are saved as `spmT_0001.nii`, `spmT_0002.nii` etc., also in the working directory.

#### 30.4.5 Maximum Intensity Projections

SPM displays a Maximum Intensity Projection (MIP) of the statistical map in the Graphics window. The MIP is projected on a glass brain in three orthogonal planes. The MIP is surfable: right-clicking in the MIP will activate a pulldown menu, left-clicking on the red cursor will allow it to be dragged to a new position.



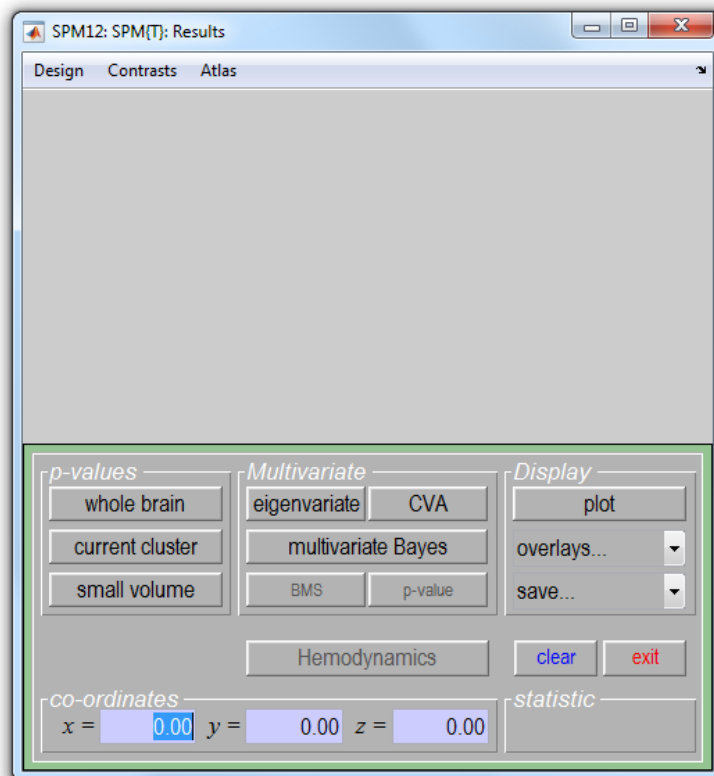


Figure 30.14: *SPM's Interactive window during results assessment. The “p-values” section is used to produce tables of statistical information. The visualisation section is used to plot responses at a voxel or to visual activations overlaid on anatomical images. The “Multivariate” section, ie. the “eigenvariate” button, is used to extract data for subsequent analyses such as assessment of PsychoPhysiological Interactions (PPIs) or Dynamic Causal Models (DCMs).*

SPM: Satellite Results Table

Statistics: *p*-values adjusted for search volume

set-level		cluster-level				peak-level				mm mm mm	
<i>p</i>	<i>c</i>	<i>P</i> <sub>FWE-corr</sub>	<i>q</i> <sub>FDR-corr</sub>	<i>k</i> <sub>E</sub>	<i>P</i> <sub>uncorr</sub>	<i>P</i> <sub>FWE-corr</sub>	<i>q</i> <sub>FDR-corr</sub>	<i>T</i>	( <i>Z</i> <sub>z</sub> )	<i>P</i> <sub>uncorr</sub>	
0.000	12	0.000	0.000	425	0.000	0.000	0.000	17.09	Inf	0.000	57 -22 11
						0.000	0.000	14.42	Inf	0.000	66 -10 -1
						0.000	0.000	10.89	Inf	0.000	69 -19 11
		0.000	0.000	568	0.000	0.000	0.000	15.94	Inf	0.000	-63 -28 14
						0.000	0.000	14.68	Inf	0.000	-48 -34 14
						0.000	0.000	13.23	Inf	0.000	-66 -10 2
		0.000	0.000	31	0.000	0.000	0.003	7.03	6.12	0.000	36 -28 -13
		0.000	0.001	12	0.000	0.000	0.004	6.90	6.03	0.000	54 -1 44
		0.003	0.096	3	0.048	0.002	0.075	6.07	5.44	0.000	-63 -55 -7
		0.000	0.003	10	0.001	0.006	0.207	5.77	5.22	0.000	-33 -31 -19
		0.015	0.232	1	0.232	0.032	0.783	5.37	4.91	0.000	-39 -76 -28
		0.015	0.232	1	0.232	0.032	0.783	5.37	4.91	0.000	57 -43 56
		0.015	0.232	1	0.232	0.042	0.923	5.30	4.86	0.000	-57 -1 41
		0.015	0.232	1	0.232	0.043	0.923	5.29	4.85	0.000	-45 41 5
		0.015	0.232	1	0.232	0.044	0.923	5.29	4.85	0.000	48 26 20
		0.015	0.232	1	0.232	0.050	0.999	5.25	4.82	0.000	45 17 23

table shows 3 local maxima more than 8.0mm apart

Height threshold: *T* = 5.25, *p* = 0.000 (0.050)      Degrees of freedom = [1.0, 73.0]  
 Extent threshold: *k* = 0 voxels      FWHM = 9.9 9.9 8.6 mm mm mm  
 Expected voxels per cluster: *k* >= 0.753      Volume: 1901397 = 70421 voxels = 1995.7 resels  
 Expected number of clusters: <= 0.07      Voxel size: 3.0 3.0 3.0 mm mm mm; (resel = 31.52 voxels)  
 FWEp: 5.253, FDRp: 6.320, FWEc: 1, FDRc: 10

Figure 30.15: Volume table for “listening > rest” effect. This table of values was created by pressing the SPM Figure > Results Table option at the top of the Graphics window and then pressing the “whole brain” button. This displays the table of results in a separate window.

### 30.4.6 Design matrix

SPM also displays the design matrix with the selected contrast. The design matrix is also surfable: right-clicking will show parameter names, left-clicking will show design matrix values for each scan.

In the SPM Interactive window (lower left panel) a button box appears with various options for displaying statistical results (*p*-values panel) and creating plots/overlays (visualisation panel). Clicking “Design” (upper left) will activate a pulldown menu as in the “Explore design” option.

### 30.4.7 Statistical tables

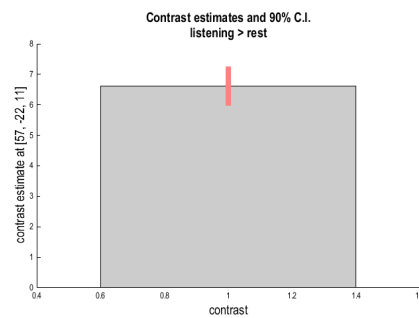
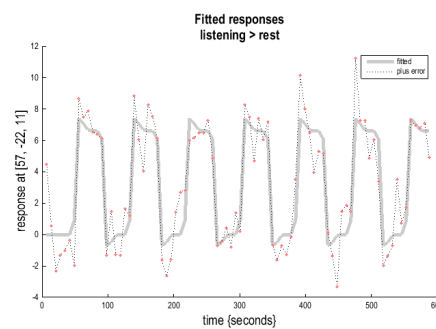
To get a summary of local maxima, press the “whole brain” button in the *p*-values section of the Interactive window. This will list all clusters above the chosen level of significance as well as separate (>8mm apart) maxima within a cluster, with details of significance thresholds and search volume underneath, as shown in Figure 30.15

The columns in volume table show, from right to left:

- **x, y, z (mm)**: coordinates in MNI space for each maximum.
- **peak-level**: the chance (*p*) of finding (under the null hypothesis) a peak with this or a greater height (*T*- or *Z*-statistic), corrected (FWE or FDR)/ uncorrected for search volume.
- **cluster-level**: the chance (*p*) of finding a cluster with this many (*k*) or a greater number of voxels, corrected (FWE or FDR)/ uncorrected for search volume.
- **set-level**: the chance (*p*) of finding this (*c*) or a greater number of clusters in the search volume.

It is also worth noting that:

- The table is surfable: clicking a row of cluster coordinates will move the pointer in the MIP to that cluster, clicking other numbers will display the exact value in the MATLAB window (e.g. 0.000 = 6.1971e-07).
- To inspect a specific cluster (e.g., in this example data set, the right auditory cortex), either move the cursor in the MIP (by left-clicking and dragging the cursor, or right-clicking the MIP background which will activate a pulldown menu).

Figure 30.16: *Estimated effect size.*Figure 30.17: *Fitted responses.*

- Alternatively, click the cluster coordinates in the volume table, or type the coordinates in the co-ordinates section of the Interactive window.

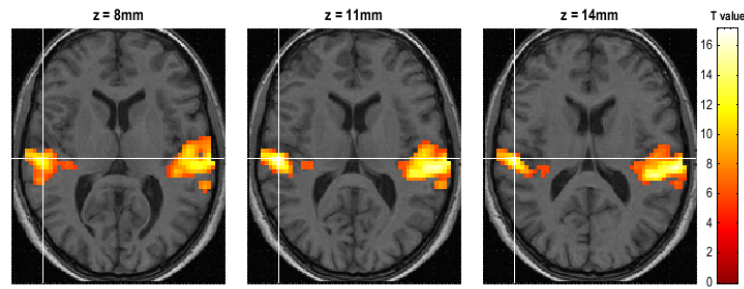
It is also possible to produce tables of statistical information for a single cluster of interest rather than for the whole volume. Firstly, select the relevant cluster in the MIP and then press the “current cluster” button in the  $p$ -values section of the Interactive window. This will show coordinates and voxel-level statistics for local maxima ( $>4\text{mm}$  apart) in the selected cluster. This table is also surfable.

### 30.4.8 Plotting responses at a voxel

A voxel can be chosen with coordinates corresponding to those in the Interactive window. The responses at this voxel can then be plotted using the “Plot” button in the visualisation section of the Interactive window. This will provide you with five further options:

1. Contrast estimates and 90% CI: SPM will prompt for a specific contrast (e.g., listening $>$ rest). The plot will show effect size and 90% confidence intervals. See eg. Figure 30.16.
2. Fitted responses: Plots adjusted data and fitted response across session/subject. SPM will prompt for a specific contrast and provides the option to choose different ordinates (“an explanatory variable”, “scan or time”, or “user specified”). If “scan or time”, the plot will show adjusted or fitted data with errors added as shown in Figure 30.17.
3. Event-related responses: Plots adjusted data and fitted response across peri-stimulus time.
4. Parametric responses.
5. Volterra kernels.

For plotting event-related responses SPM provides three options

Figure 30.18: *Slices*.

1. Fitted response and PSTH (peri-stimulus time histogram): plots mean regressor(s) (ie. averaged over session) and mean signal  $\pm$  SE for each peri-stimulus time bin.
2. Fitted response and 90% CI: plots mean regressor(s) along with a 90% confidence interval.
3. Fitted response and adjusted data: plots regressor(s) and individual data (note that in this example the data are shown in columns due to the fixed TR/ISI relationship).

Its worth noting that

- The values for the fitted response across session/subject for the selected plot can be displayed and accessed in the MATLAB window by typing “Y”. Typing “y” will display the adjusted data.
- “Adjusted” data = adjusted for confounds (e.g., global flow) and high- and low pass filtering.

### 30.4.9 Overlays

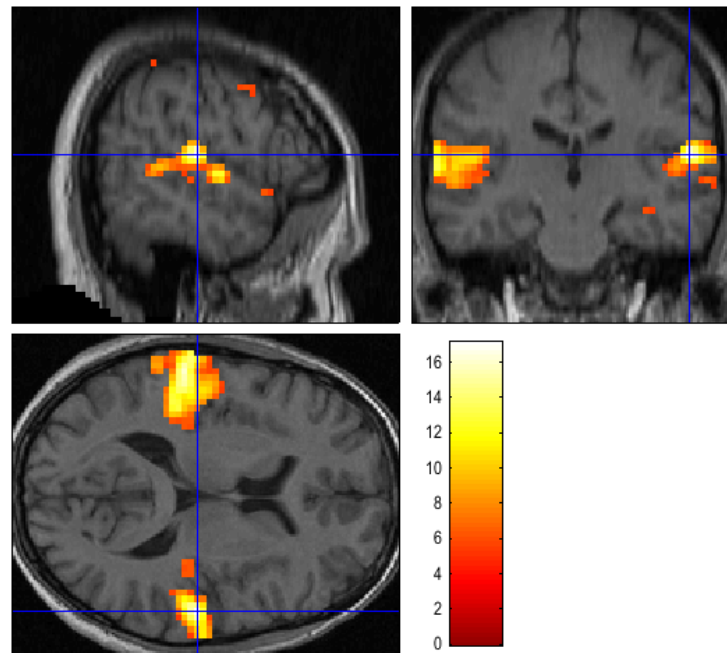
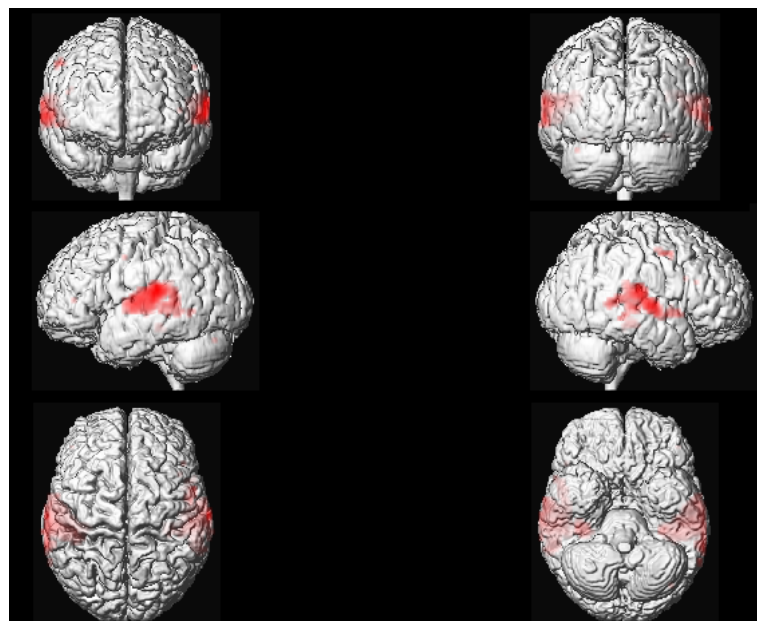
The visualisation section of the Interactive window also provides an overlay facility for anatomical visualisation of clusters of activation. Pressing “Overlays” will activate a pulldown menu with several options including:

1. **Slices:** overlay on three adjacent (2mm) transaxial slices. SPM will prompt for an image for rendering. This could be a canonical image (see `spm_templates.man`) or an individual T1/mean EPI image for single-subject analyses. Beware that the left-right convention in the display of that option will depend on how your data are actually stored on disk.
2. **Sections:** overlay on three intersecting (sagittal, coronal, axial) slices. These renderings are surfable: clicking the images will move the crosshair.
3. **Render:** overlay on a volume rendered brain.

Thresholded SPMS can be saved as NIfTI image files in the working directory by using the “Save” button in the Interactive window. In Figures 30.18, 30.19 and 30.20 the ‘listening > rest’ activation has been superimposed on the spatially normalised, bias-corrected anatomical image `wmsM00223_002.nii` created earlier.

For the “Render” option we first created a rendering for this subject. This was implemented by

- “Normalise (Write)” the two images `c1sM00223_002.nii` and `c2sM00223_002.nii` using the “Deformation Field” `y_sM00223_002.nii` and a voxel size of [1 1 1].
- Selecting “Extract Surface” from the “Render” pulldown menu.
- Selecting the gray and white matter images `wc1sM00223_002.nii` and `wc2sM00223_002.nii` created in the first step.
- Saving the results using the default options (Rendering and Surface).

Figure 30.19: *Sections.*Figure 30.20: *Render.*

Render (Acrobat Reader required)

Figure 30.21: *3D Rendering using canonical mesh.*

SPM plots the rendered anatomical image in the graphics window and saves it as `render_wc1sM00223_002.mat`. The surface image is saved as `surf_wc1sM00223_002.mat`.

It is also possible to project and display the results on a surface mesh, we are going to use here one of the canonical mesh distributed with SPM (in MNI space). Press “Overlays” and choose “Render”, then go in the `canonical` folder of your SPM installation and select file `cortex_20484.surf.gii` (this is a surface mesh stored using the GIFTI format) and you will obtain a figure similar to [30.21](#).

# Chapter 31

## Face fMRI data

As another, more sophisticated example, consider the data from a repetition priming experiment performed using event-related fMRI. Briefly, this is a  $2 \times 2$  factorial study with factors “fame” and “repetition” where famous and non-famous faces were presented twice against a checkerboard baseline (for more details, see [55]). The subject was asked to make fame judgements by making key presses. There are thus four event-types of interest; first and second presentations of famous and non-famous faces, which we denote N1, N2, F1 and F2. The experimental stimuli and timings of events are shown in Figures 31.1 and 31.2.

Images were acquired using continuous Echo-Planar Imaging (EPI) with TE=40ms, TR=2s and 24 descending slices ( $64 \times 64 \times 3 \times 3 \text{ mm}^2$ ), 3mm thick with a 1.5mm gap. The data archive is available from the SPM website<sup>1</sup>. This contains 351 Analyze format functional images `sm03953_0005_*. {hdr, im}` of dimension  $64 \times 64 \times 24$  with  $3 \times 3 \times 4.5 \text{ mm}^3$  voxels. A structural image is also provided in Analyze format (`sm03953_0007. {hdr, img}`).

To analyse the data, first create a new directory DIR eg. `C:\data\face_rep`, in which to place the results of your analysis. Then create 4 subdirectories (i) `jobs`, (ii) `categorical`, (iii) `parametric` and (iv) `bayesian`. As the analysis proceeds these directories will be filled with job-specification files, design matrices and models estimated using classical or Bayesian methods.

As well as the classical/Bayesian distinction we will show how this data can be analysed from a parametric as well as a categorical perspective. We will look at the main effects of fame and repetition and in the parameteric analysis we will look at responses as a function of “lag”, that is, the number of faces intervening between repetition of a specific face.

Start up MATLAB enter your jobs directory and type `spm_fmri` at the MATLAB prompt. SPM will then open in fMRI mode with three windows (1) the top-left or “Menu” window, (2) the bottom-left or “Interactive” window and (3) the right-hand or “Graphics” window. Analysis then takes place in three major stages (i) spatial pre-processing, (ii) model specification, review and estimation and (iii) inference. These stages organise the buttons in SPM’s base window.

### 31.1 Spatial pre-processing

#### 31.1.1 Display

Display eg. the first functional image using the “Display” button. Note orbitofrontal and inferior temporal drop-out and ghosting. This can be seen more clearly by selecting “Brighten” from the “Effects” menu in the “Colours” menu from the “SPM Figure” tab at the top of the Graphics window.

#### 31.1.2 Realignment

Under the spatial pre-processing section of the SPM base window select REALIGN (EST & RES) from the REALIGN pulldown menu. This will call up a realignment job specification in the batch editor window. Then

---

<sup>1</sup>Face Repetition dataset: [http://www.fil.ion.ucl.ac.uk/spm/data/face\\_rep/](http://www.fil.ion.ucl.ac.uk/spm/data/face_rep/)

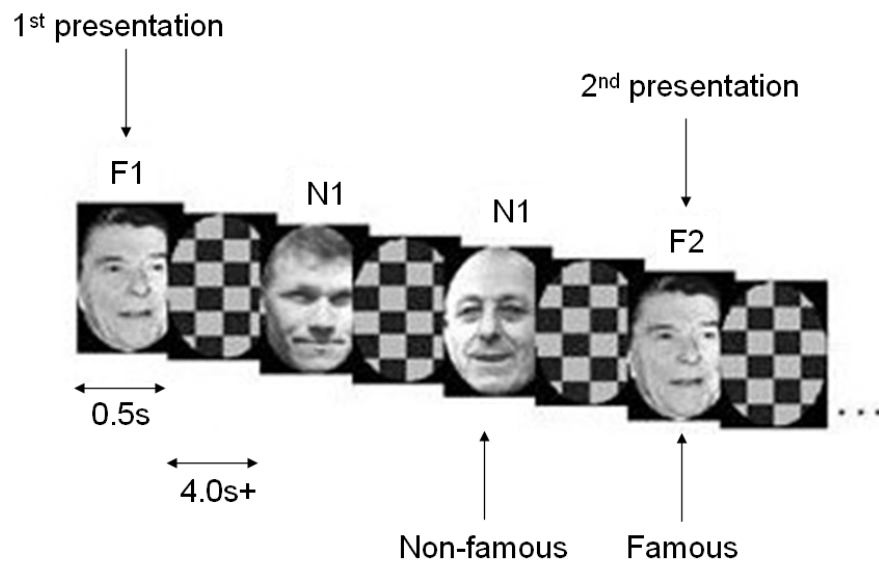


Figure 31.1: **Face repetition paradigm:** There were 2 presentations of 26 Famous and 26 Nonfamous Greyscale photographs, for 0.5s each, randomly intermixed. The minimal Stimulus Onset Asynchrony (SOA)=4.5s, with probability 2/3 (ie 1/3 null events). The subject made one of two right finger key presses denoting whether or not the subject thought the face was famous.

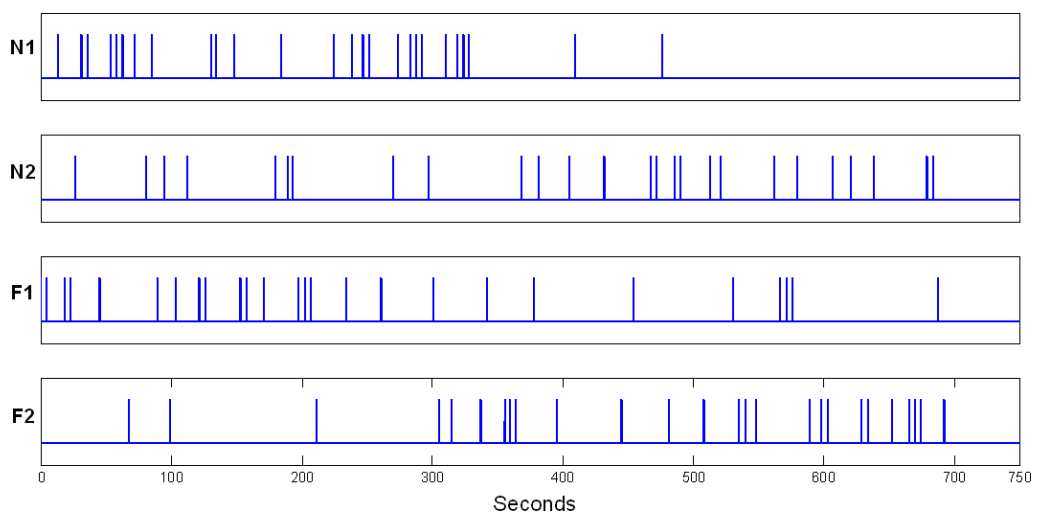


Figure 31.2: *Time series of events.*



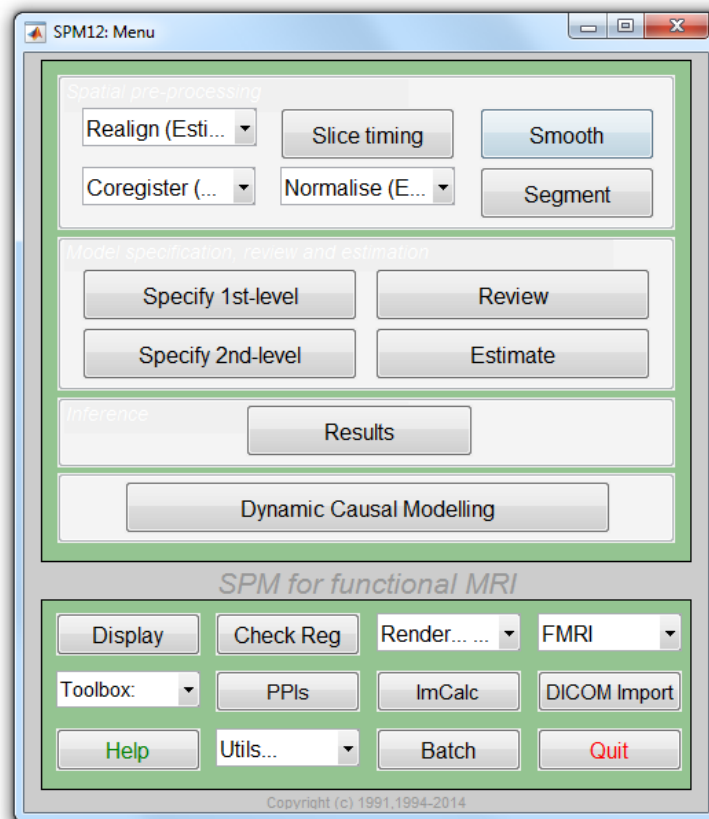


Figure 31.3: The SPM base window comprises three sections (i) spatial pre-processing, (ii) model specification, review and estimation and (iii) inference.

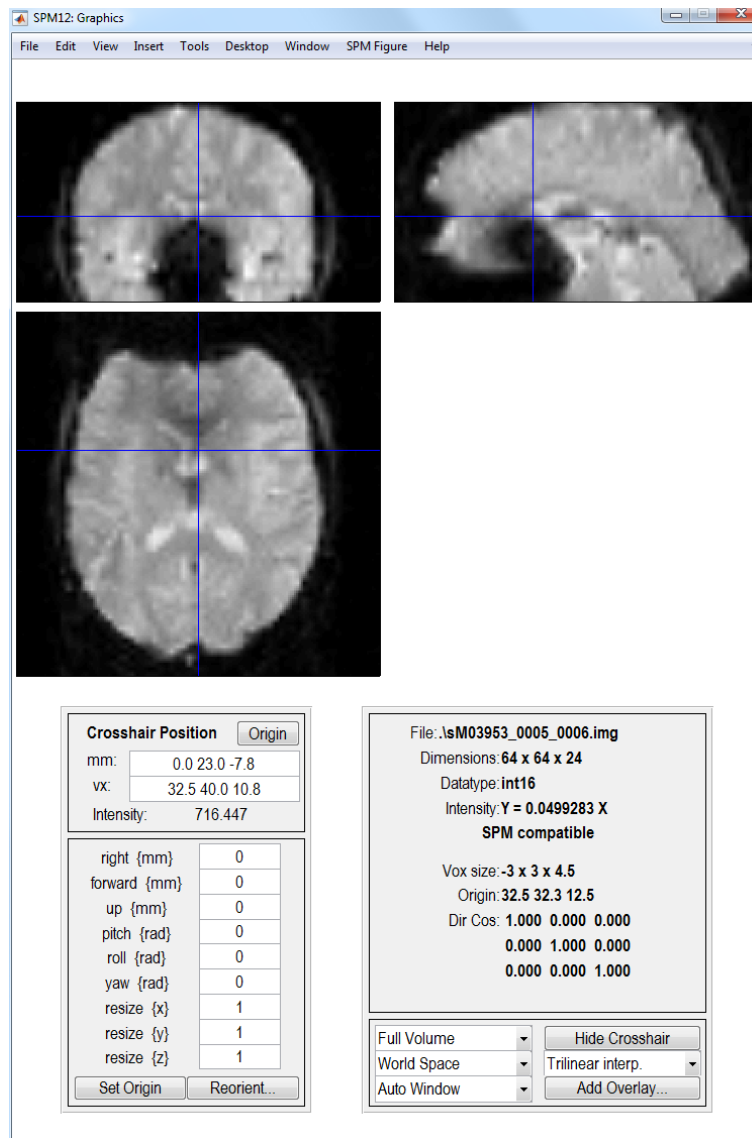


Figure 31.4: *Signal dropout in EPI images.*

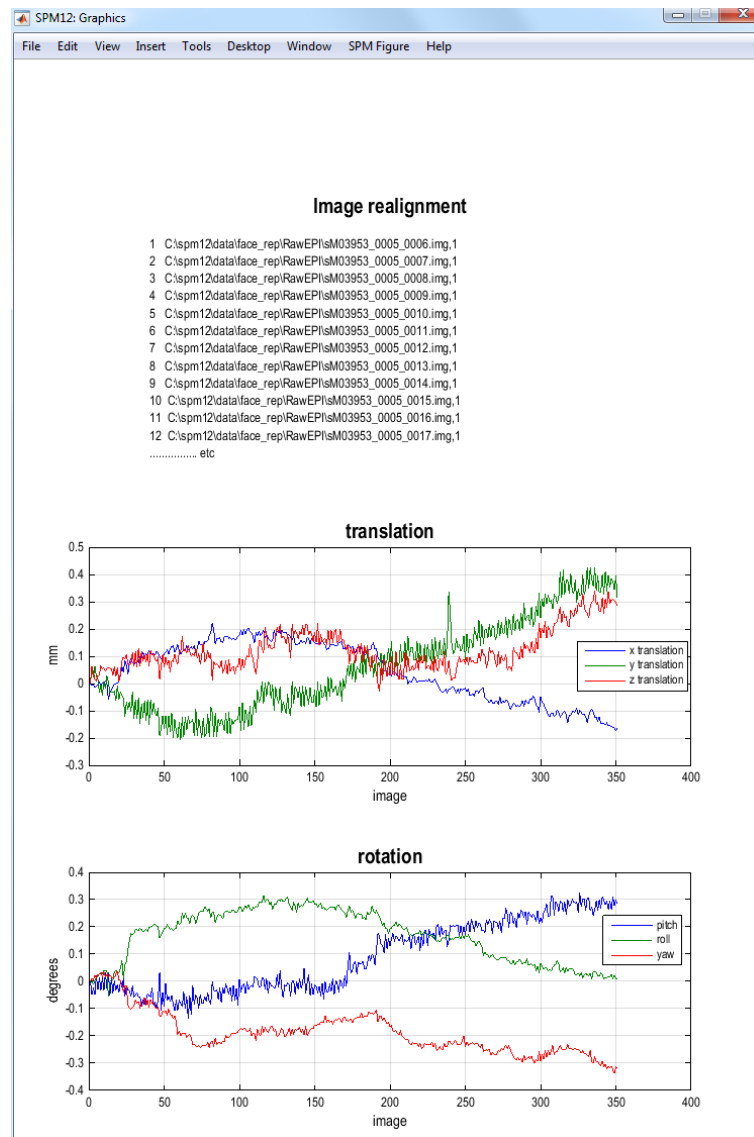


Figure 31.5: **Realignment of face data:** Movement less than the size of a voxel, which for this data set is 3mm, is not considered problematic.

- Highlight data, select “New Session”, then highlight the newly created “Session” option.
- Select “Specify Files” and use the SPM file selector to choose all of your functional images eg. `sM03953_0005_*.img`. You should select 351 files.
- Save the job file as eg. `DIR/jobs/realign.mat`.
- Press the Run button in the batch editor window (green triangle).

This will run the realign job which will write realigned images into the directory where the functional images are. These new images will be prefixed with the letter “r”. SPM will then plot the estimated time series of translations and rotations shown in Figure 31.5. These data, the realignment parameters, are also saved to a file eg. `rp_sM03953_0005_0006.txt`, so that these variables can be used as regressors when fitting GLMs. This allows movements effects to be discounted when looking for brain activations.

SPM will also create a mean image eg. `meansM03953_0005_0006.{hdr,img}` which will be used in the next step of spatial processing - coregistration.

### 31.1.3 Slice timing correction

Press the SLICE TIMING button. This will call up the specification of a slice timing job in the batch editor window. Note that these data consist of  $N=24$  axial slices acquired continuously with a  $TR=2s$  (ie  $TA = TR - TR/N$ , where  $TA$  is the time between the onset of the first and last slice of one volume, and the  $TR$  is the time between the onset of the first slice of one volume and the first slice of next volume) and in a descending order (ie, most superior slice was sampled first). The data however are ordered within the file such that the first slice (slice number 1) is the most inferior slice, making the slice acquisition order [24 23 22 ... 1].

- Highlight “Data” and select “New Sessions”
- Highlight the newly create “Sessions” option, “Specify Files” and select the 351 realigned functional images using the filter `~r.*`.
- Select “Number of Slices” and enter 24.
- Select  $TR$  and enter 2.
- Select  $TA$  and enter 1.92 (or  $2 - 2/24$ ).
- Select “Slice order” and enter 24:-1:1.
- Select “Reference Slice”, and enter 12.
- Save the job as `slice_timing.mat` and press the “Run” button.

SPM will write slice-time corrected files with the prefix “a” in the functional data directory.

### 31.1.4 Coregistration

Select COREGISTER (ESTIMATE) from the **Coregister** pulldown menu. This will call up the specification of a coregistration job in the batch editor window.

- Highlight “Reference Image” and then select the mean functional image `meansM03953_0005_0006.img`.
- Highlight “Source Image” and then select the structural image eg. `sM03953_0007.img`.
- Press the “Save” button and save the job as `coreg.job`
- Then press the “Run” button.

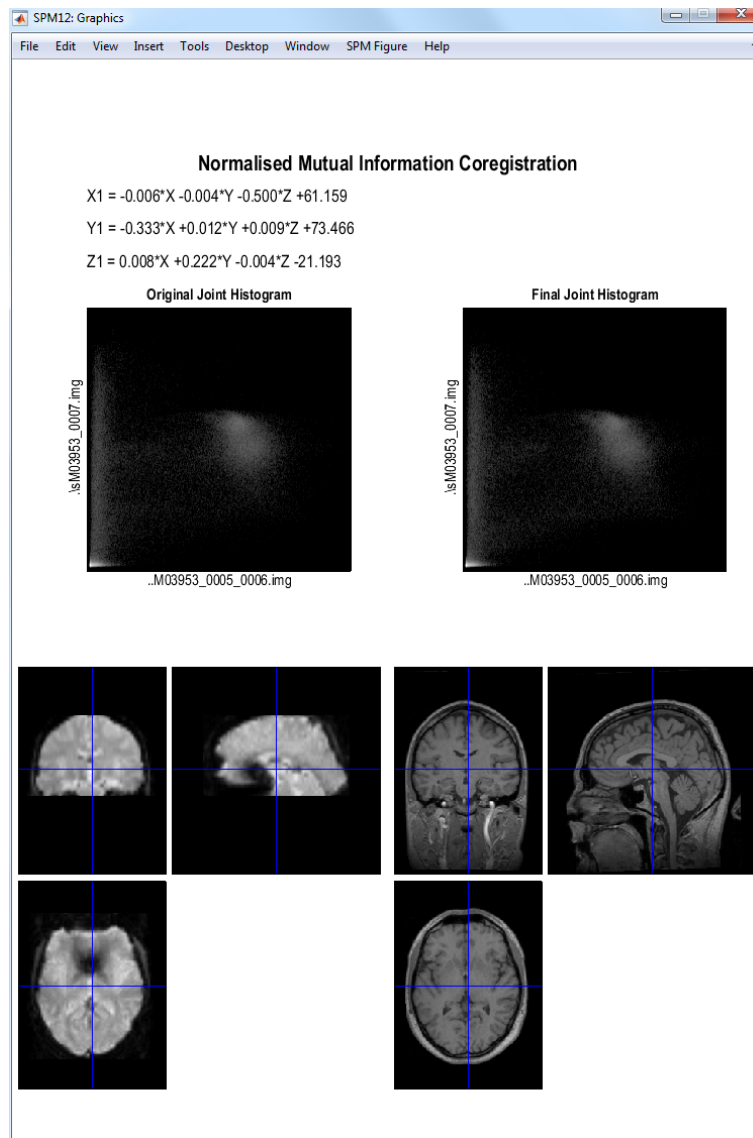
SPM will then implement a coregistration between the structural and functional data that maximises the mutual information. The image in figure 31.6 should then appear in the Graphics window. SPM will have changed the header of the source file which in this case is the structural image `sM03953_0007.hdr`.

### 31.1.5 Segmentation

Press the SEGMENT button. This will call up the specification of a segmentation job in the batch editor window. Highlight the “Volumes” field in “Data > Channels” and then select the subjects coregistered anatomical image eg. `sM03953_0007.img`. Change “Save Bias Corrected” so that it contains “Save Bias Corrected” instead of “Save Nothing”. At the bottom of the list, select “Forward” in “Deformation Fields”. Save the job file as `segment.mat` and then press the Run button. SPM will segment the structural image using the default tissue probability maps as priors. SPM will create, by default, gray and white matter images and bias-field corrected structural image. These can be viewed using the CheckReg facility as described in the previous section. Figure 31.7 shows the gray matter image, `c1sM03953_0007.nii`, along with the original structural<sup>2</sup>.

SPM will also write a spatial normalisation deformation field file eg. `y_sM03953_0007.nii` file in the original structural directory. This will be used in the next section to normalise the functional data.

<sup>2</sup>Segmentation can sometimes fail if the source (structural) image is not close in orientation to the MNI templates. It is generally advisable to manually orient the structural to match the template (ie MNI space) as close as possible by using the “Display” button, adjusting x/y/z/pitch/roll/yaw, and then pressing the “Reorient” button.

Figure 31.6: *Mutual Information Coregistration of Face data.*

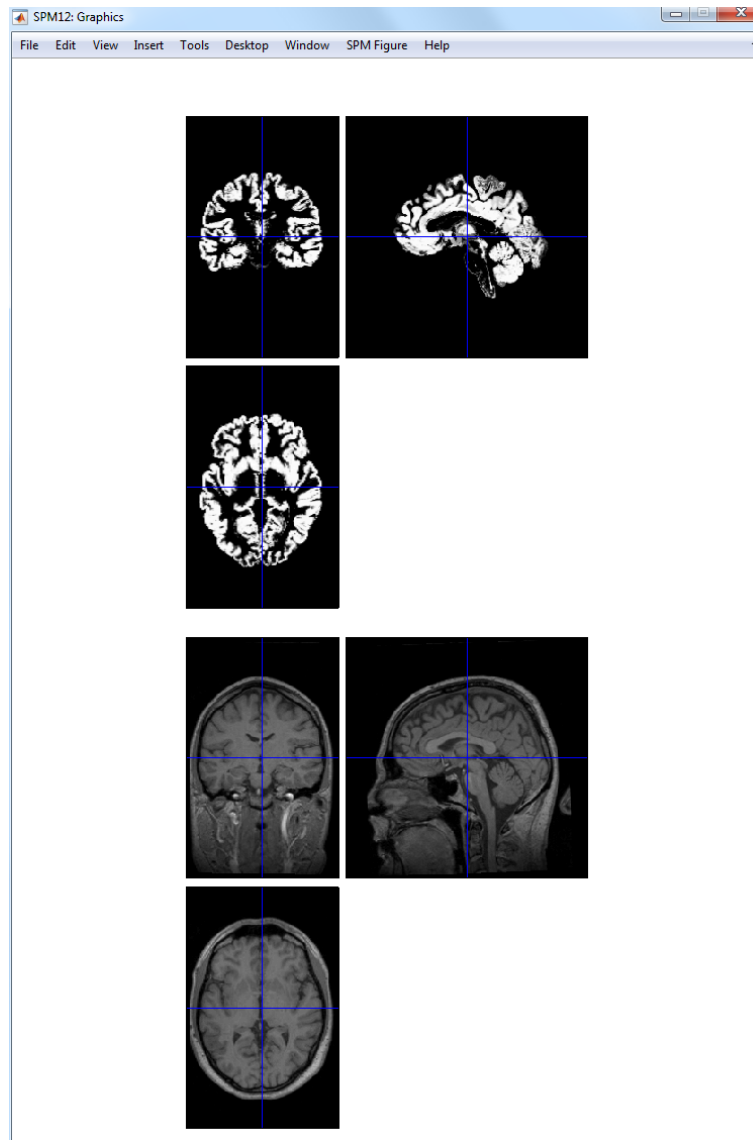


Figure 31.7: *Gray matter (top) produced by segmentation of structural image (below).*

### 31.1.6 Normalise

Select NORMALISE (WRITE) from the NORMALISE pulldown menu. This will call up the specification of a normalise job in the batch editor window.

- Highlight “Data”, select “New Subject”.
- Open “Subject”, highlight “Deformation field” and select the `y_sm03953_0007.nii` file that you created in the previous section.
- Highlight “Images to write” and select all of the slice-time corrected, realigned functional images `arsM*.img`. Note: This can be done efficiently by changing the filter in the SPM file selector to `^ar.*`. You can then right click over the listed files, choose “Select all”. You might also want to select the mean functional image created during realignment (which would not be affected by slice-time correction), i.e, the `meansM03953_0005_006.img`. Then press “Done”.
- Open “Writing Options”, and change “Voxel sizes” from `[2 2 2]` to `[3 3 3]`<sup>3</sup>.
- Press “Save”, save the job as `normalise.mat` and then press the Run button.

SPM will then write spatially normalised files to the functional data directory. These files have the prefix “w”.

If you wish to superimpose a subject’s functional activations on their own anatomy<sup>4</sup> you will also need to apply the spatial normalisation parameters to their (bias-corrected) anatomical image. To do this

- Select NORMALISE (WRITE), highlight ‘Data’, select “New Subject”.
- Highlight “Deformation field”, select the `y_sm03953_0007.nii` file that you created in the previous section, press “Done”.
- Highlight “Images to Write”, select the bias-corrected structural eg. `msM03953_0007.nii`, press “Done”.
- Open “Writing Options”, select voxel sizes and change the default `[2 2 2]` to `[1 1 1]` which better matches the original resolution of the images `[1 1 1.5]`.
- Save the job as `norm_struct.mat` and press Run button.

### 31.1.7 Smoothing

Press the SMOOTH button<sup>5</sup>. This will call up the specification of a smooth job in the batch editor window.

- Select “Images to Smooth” and then select the spatially normalised files created in the last section eg. `war*.img`.
- Save the job as `smooth.mat` and press Run button.

This will smooth the data by (the default) 8mm in each direction, the default smoothing kernel width.

<sup>3</sup>This step is not strictly necessary. It will write images out at a resolution closer to that at which they were acquired. This will speed up subsequent analysis and is necessary, for example, to make Bayesian fMRI analysis computationally efficient.

<sup>4</sup>Beginners may wish to skip this step, and instead just superimpose functional activations on an “canonical structural image”.

<sup>5</sup>The smoothing step is unnecessary if you are only interested in Bayesian analysis of your functional data.

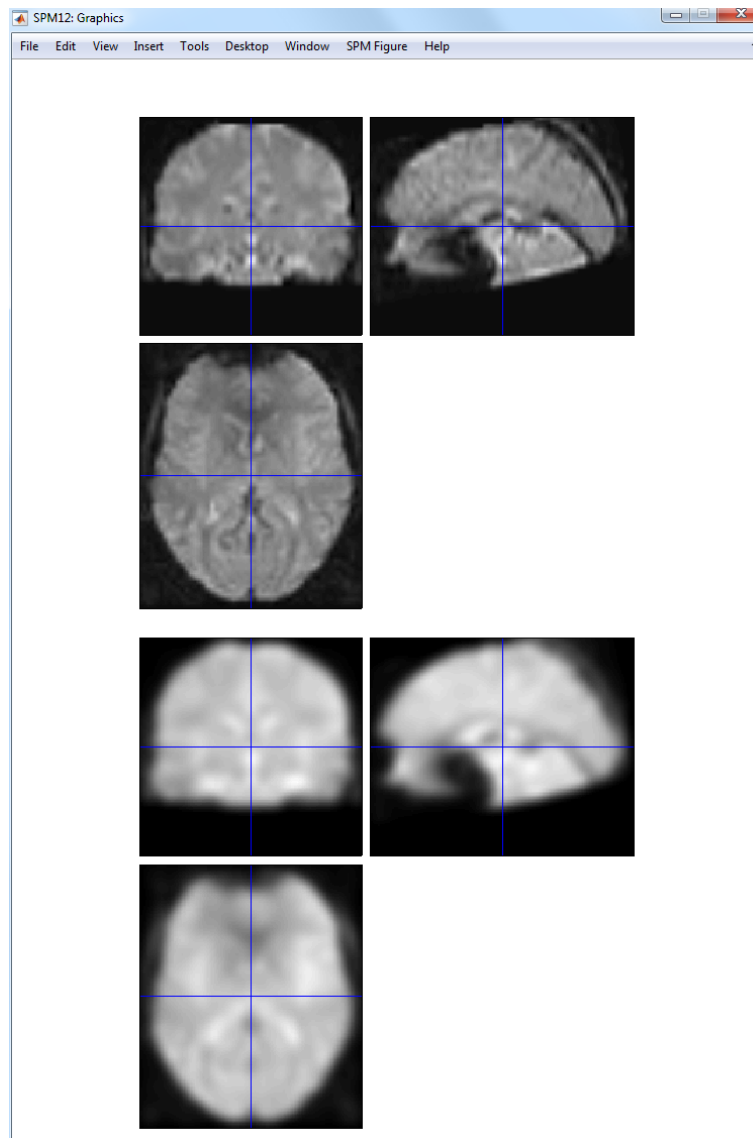


Figure 31.8: *Functional image (top) and 8mm-smoothed functional image (bottom). These images were plotted using SPM's "CheckReg" facility.*



## 31.2 Modelling categorical responses

Before setting up the design matrix we must first load the Stimulus Onsets Times (SOTs) into MATLAB. SOTs are stored in the `sots.mat` file in a cell array such that eg. `sot{1}` contains stimulus onset times in TRs for event type 1, which is N1. Event-types 2, 3 and 4 are N2, F1 and F2.<sup>6</sup>

- At the MATLAB command prompt type `load sots`

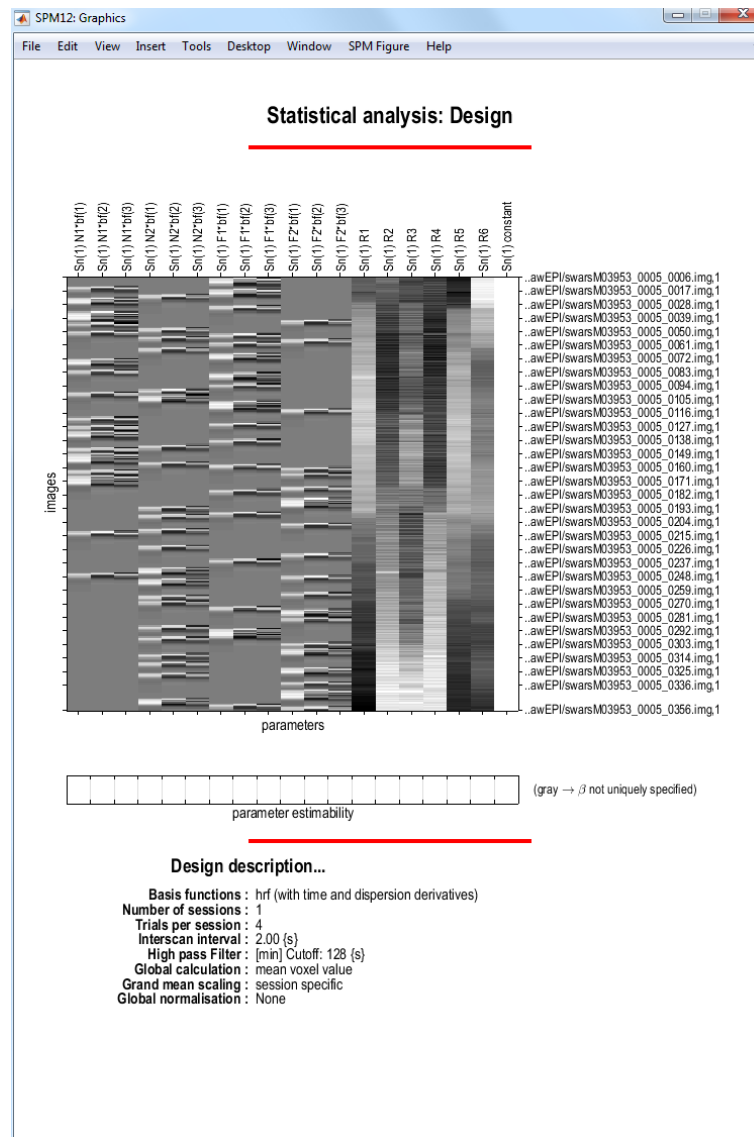
Now press the SPECIFY 1ST-LEVEL button. This will call up the specification of a fMRI specification job in the batch editor window. Then

- For “Directory”, select the “categorical” folder you created earlier,
- In the “Timing parameters” option,
- Highlight “Units for design” and select “Scans”,
- Highlight “Interscan interval” and enter 2,
- Highlight “Microtime resolution” and enter 24,
- Highlight “Microtime onset” and enter 12. These last two options make the creating of regressors commensurate with the slice-time correction we have applied to the data, given that there are 24 slices and that the reference slice to which the data were slice-time corrected was the 12th (middle slice in time).
- Highlight “Data and Design” and select “New Subject/Session”.
- Highlight “Scans” and use SPM’s file selector to choose the 351 smoothed, normalised, slice-time corrected, realigned functional images ie `swarsM.img`. These can be selected easily using the `^swar.*` filter, and select all. Then press “Done”.
- Highlight “Conditions” and select “New condition”<sup>7</sup>.
- Open the newly created “Condition” option. Highlight “Name” and enter “N1”. Highlight “Onsets” and enter `sot{1}`. Highlight “Durations” and enter 0.
- Highlight “Conditions” and select “Replicate condition”.
- Open the newly created “Condition” option (the lowest one). Highlight “Name” and change to “N2”. Highlight “Onsets” and enter `sot{2}`.
- Highlight “Conditions” and select “Replicate condition”.
- Open the newly created “Condition” option (the lowest one). Highlight “Name” and change to “F1”. Highlight “Onsets” and enter `sot{3}`.
- Highlight “Conditions” and select “Replicate condition”.
- Open the newly created “Condition” option (the lowest one). Highlight “Name” and change to “F2”. Highlight “Onsets” and enter `sot{4}`.
- Highlight “Multiple Regressors” and select the realignment parameter file `rp_sm03953_0005_0006.txt` file that was saved during the realignment preprocessing step in the folder containing the fMRI data<sup>8</sup>.

<sup>6</sup>Unlike previous analyses of these data in SPM99 and SPM2, we will not bother with extra event-types for the (rare) error trials.

<sup>7</sup>It is also possible to enter information about all of the conditions in one go. This requires much less button pressing and can be implemented by highlighting the “Multiple conditions” option and then selecting the `all-conditions.mat` file, which is also provided on the webpage.

<sup>8</sup>It is also possible to enter regressors one by one by highlighting “Regressors” and selecting “New Regressor” for each one. Here, we benefit from the fact that the realignment stage produced a text file with the correct number of rows (351) and columns (6) for SPM to add 6 regressors to model (linear) rigid-body movement effects.

Figure 31.9: *Design matrix.*

- Highlight “Factorial Design”, select “New Factor”, open the newly created “Factor” option, highlight “Name” and enter “Fam”, highlight “Levels” and enter 2.
- Highlight “Factorial Design”, select “New Factor”, open the newly created “Factor” option, highlight “Name” and enter “Rep”, highlight “Levels” and enter 2<sup>9</sup>.
- Open “Canonical HRF” under “Basis Functions”. Select “Model derivatives” and select “Time and Dispersion derivatives”.
- Highlight “Directory” and select the DIR/categorical directory you created earlier.
- Save the job as `categorical_spec.mat` and press the Run button.

SPM will then write an `SPM.mat` file to the DIR/categorical directory. It will also plot the design matrix, as shown in Figure 31.9.

<sup>9</sup>The order of naming these factors is important - the factor to be specified first is the one that “changes slowest” ie. as we go through the list of conditions N1, N2, F1, F2 the factor “repetition” changes every condition and the factor “fame” changes every other condition. So “Fam” changes slowest and is entered first.

At this stage it is advisable to check your model specification using SPM's review facility which is accessed via the "Review" button. This brings up a "Design" tab on the interactive window clicking on which produces a pulldown menu. If you select the first item "Design Matrix" SPM will produce the image shown in Figure 31.9. If you select "Explore" then "Session 1" then "N1", SPM will produce the plots shown in Figure 31.10.

### 31.2.1 Estimate

Press the ESTIMATE button. This will call up the specification of an fMRI estimation job in the batch editor window. Then

- Highlight the "Select SPM.mat" option and then choose the SPM.mat file saved in the DIR/categorical directory.
- Save the job as categorical\_est.job and press Run button.

SPM will write a number of files into the selected directory including an SPM.mat file.

### 31.2.2 Inference for categorical design

Press "Results" and select the SPM.mat file from DIR/categorical. This will again invoke the contrast manager. Because we specified that our model was using a "Factorial design" a number of contrasts have been specified automatically, as shown in Figure 31.11.

- Select contrast number 5. This is a t-contrast **Positive effect of condition\_1** This will show regions where the average effect of presenting faces is significantly positive, as modelled by the first regressor (hence the \_1), the canonical HRF. Press 'Done'.
- *Apply masking ? [None/Contrast/Image]*
- Specify None.
- *p value adjustment to control: [FWE/none]*
- Select FWE
- *Corrected p value(family-wise error)*
- Accept the default value, 0.05
- *Extent threshold {voxels} [0]*
- Accept the default value, 0.

SPM will then produce the MIP shown in Figure 31.12.

### 31.2.3 Statistical tables

To get a summary of local maxima, press the "whole brain" button in the p-values section of the interactive window. This will list all clusters above the chosen level of significance as well as separate (>8mm apart) maxima within a cluster, with details of significance thresholds and search volume underneath, as shown in Figure 31.12

The columns in volume table show, from right to left:

- **x, y, z (mm)**: coordinates in MNI space for each maximum.
- **peak-level**: the chance (p) of finding (under the null hypothesis) a peak with this or a greater height (T- or Z-statistic), corrected (FWE or FDR)/ uncorrected for search volume.
- **cluster-level**: the chance (p) of finding a cluster with this many(ke) or a greater number of voxels, corrected (FWE or FDR)/ uncorrected for search volume.

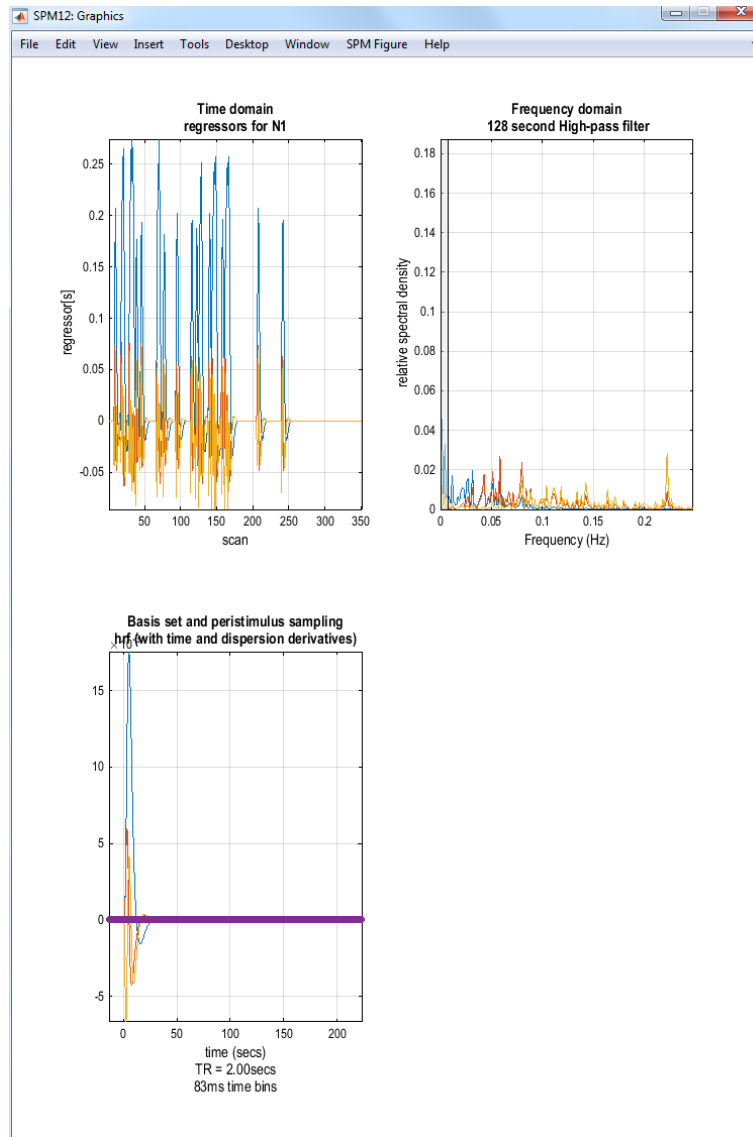


Figure 31.10: *Exploring the design matrix in Figure 31.9.* This shows the time series of the “N1” regressor (top left), the three basis functions used to convert assumed neuronal activity into hemodynamic activity (bottom left), and a frequency domain plot of the three regressors for the basis functions in this condition (top right). The frequency domain plot shows that the frequency content of the “N1” condition is generally above the set frequencies that are removed by the High Pass Filter (HPF) (these are shown in gray - in this model we accepted the default HPF cut-off of 128s or 0.008Hz).

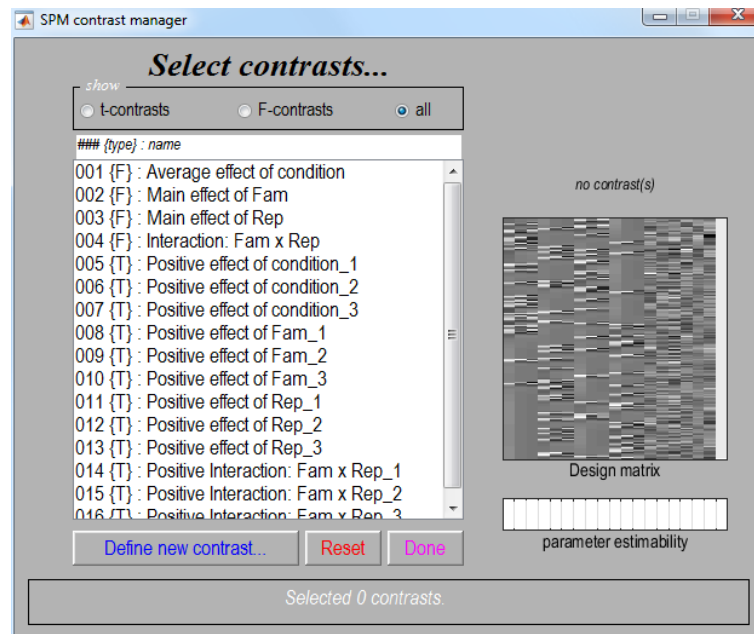


Figure 31.11: Contrast Manager containing default contrasts for categorical design.

- **set-level:** the chance ( $p$ ) of finding this ( $c$ ) or a greater number of clusters in the search volume.

Right-click on the MIP and select “goto global maximum”. The cursor will move to [39 -70 -14]. You can view this activation on the subject’s normalised, bias-corrected structural (`wmsM03953_0007img`), which gives best anatomical precision, or on the normalised mean functional (`wmeansM03953_0005_0006.nii`), which is closer to the true data and spatial resolution (including distortions in the functional EPI data).

If you select “plot” and choose “Contrast of estimates and 90% C.I” (confidence interval), and select the “Average effect of condition” contrast, you will see three bars corresponding to the parameter estimates for each basis function (summed across the 4 conditions). The BOLD impulse response in this voxel loads mainly on the canonical HRF, but also significantly (given that the error bars do not overlap zero) on the temporal and dispersion derivatives (see next Chapter).

### 31.2.4 F-contrasts

To assess the main effect of repeating faces, as characterised by both the hrf *and* its derivatives, an F-contrast is required. This is really asking whether repetition changes the *shape* of the impulse response (e.g, it might affect its latency but not peak amplitude), at least the range of shapes defined by the three basis functions. Because we have told SPM that we have a factorial design, this required contrast will have been created automatically - it is number 3.

- Press “Results” and select the `SPM.mat` file in the `DIR/categorical` directory.
- Select the “F-contrast” toggle and the contrast number 3, as shown in Figure 31.13. Press “Done”.
- *Apply masking ? [None/Contrast/Image].*
- Specify “Contrast”.
- Select contrast 5 - **Positive effect of condition\_1** (the T-contrast of activation versus baseline, collapsed across conditions, that we evaluated above)

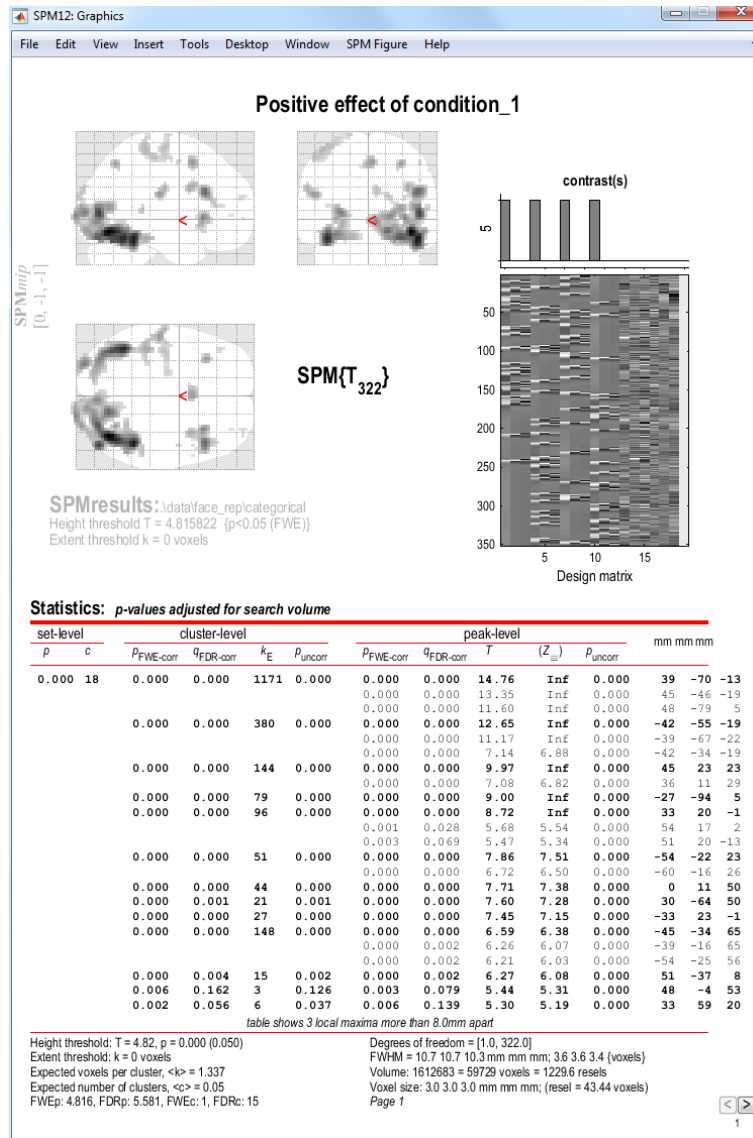


Figure 31.12: MIP and Volume table for Canonical HRF: Faces > Baseline.

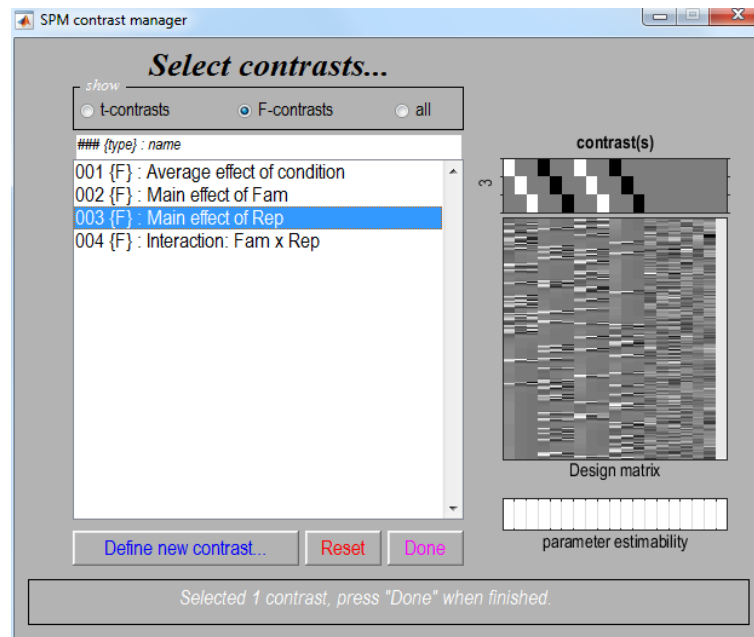


Figure 31.13: Contrast manager showing selection of the first contrast “Main effect of Rep” (repetition: F1 and N1 vs F2 and N2)

- *uncorrected mask p-value ?*
- Change to 0.001
- *nature of mask?*
- Select 'inclusive'
- *p value adjustment to control: [FWE/none]*
- Select none
- *threshold (F or p value)*
- Accept the default value, 0.001
- *Extent threshold {voxels} [0]*
- Accept the default value, 0

A MIP should then appear, the top half of which should look like Figure 31.14.

Note that this contrast will identify regions showing any effect of repetition (e.g, decreased or increased amplitudes) *within* those regions showing activations (on the canonical HRF) to faces versus baseline (at  $p < .05$  uncorrected). Select “goto global max”, which is in right ventral temporal cortex [42 -64 -8].

If you press plot and select “Event-related responses”, then “F1”, then “fitted response and PSTH”, you will see the best fitting linear combination of the canonical HRF and its two derivatives (thin red line), plus the “selectively-averaged” data (peri-stimulus histogram, PSTH), based on an FIR refit (see next Chapter). If you then select the “hold” button on the Interactive window, and then “plot” and repeat the above process for the “F2” rather than “F1” condition, you will see two estimated event-related responses, in which repetition decreases the peak response (ie  $F2 < F1$ ), as shown in Figure 31.14.

You can explore further F-contrasts, which are a powerful tool once you understand them. For example, the MIP produced by the “Average effect of condition” F-contrast looks similar to the earlier T-contrast, but importantly shows the areas for which the sums across conditions of the

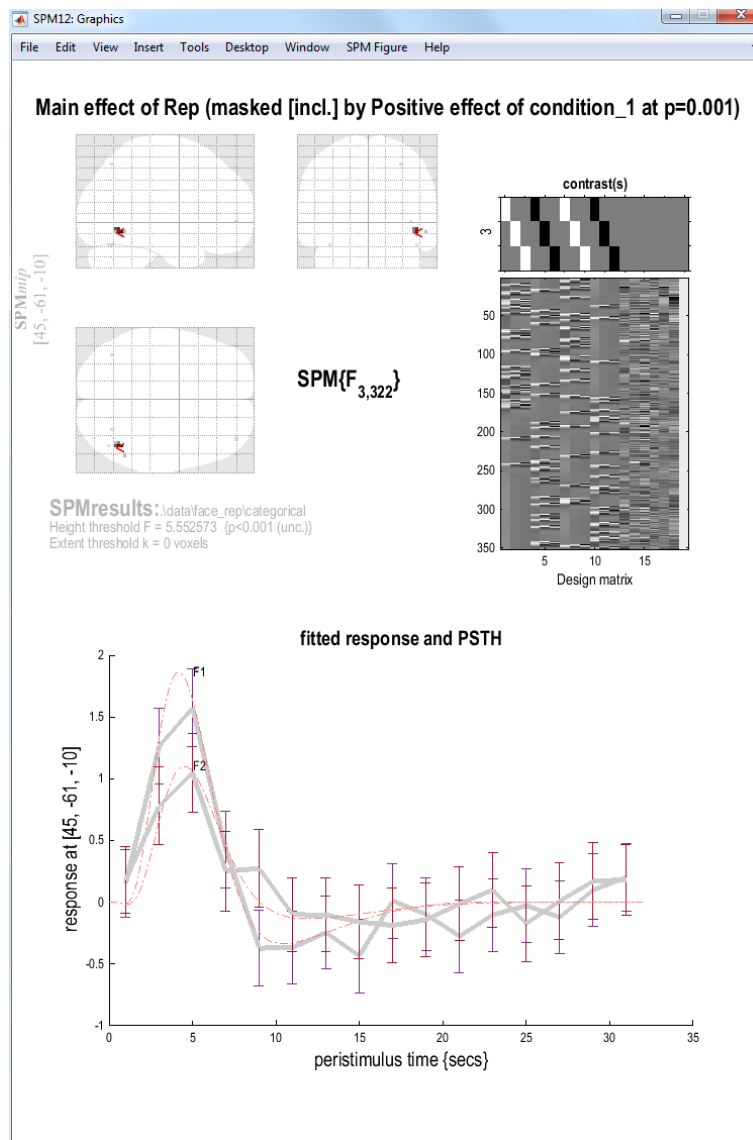


Figure 31.14: MIP for Main effect of Rep, masked inclusively with Canonical HRF: Faces > Baseline at  $p < .001$  uncorrected. Shown below are the best-fitting responses and peri-stimulus histograms (PSTH) for F1 and F2.



parameter estimates for the canonical hrf *and/or* its temporal derivative *and/or* its dispersion derivative are different from zero (baseline). The first row of this F-contrast ([1 0 0 1 0 0 1 0 0 1 0 0]) is also a two-tailed version of the above T-contrast, ie testing for both activations and deactivations versus baseline. This also means that the F-contrasts [1 0 0 1 0 0 1 0 0 1 0 0] and [-1 0 0 -1 0 0 -1 0 0 -1 0 0] are equivalent. Finally, note that an F- (or t-) contrast such as [1 1 1 1 1 1 1 1 1 1], which tests whether the mean of the canonical hrf AND its derivatives for all conditions are different from (larger than) zero is not sensible. This is because the canonical hrf and its temporal derivative may cancel each other out while being significant in their own right. The basis functions are really quite different things, and need to represent separate rows in an F-contrast.

### 31.2.5 F-contrasts for testing effects of movement

To assess movement-related activation

- Press “Results”, select the `SPM.mat` file, select “F-contrast” in the Contrast Manager. Specify e.g. “Movement-related effects” (name) and in the “contrasts weights matrix” window, or “1:12 19” in the “columns for reduced design” window.
- Submit and select the contrast, specify “Apply masking?” (none), “corrected height threshold” (FWE), and “corrected p-value” (accept default).
- When the MIP appears, select “sections” from the “overlays” pulldown menu, and select the normalised structural image (`wmsM03953_0007.nii`).

You will see there is a lot of residual movement-related artifact in the data (despite spatial realignment), which tends to be concentrated near the boundaries of tissue types (eg the edge of the brain; see Figure 31.15). (Note how the MIP can be misleading in this respect, since though it appears that the whole brain is affected, this reflects the nature of the (X-ray like) projections onto each orthogonal view; displaying the same data as sections in 3D shows that not every voxel is suprathreshold.) Even though we are not interested in such artifact, by including the realignment parameters in our design matrix, we “covary out” (linear components) of subject movement, reducing the residual error, and hence improve our statistics for the effects of interest.

## 31.3 Modelling parametric responses

Before setting up the design matrix, we must first load into MATLAB the Stimulus Onsets Times (SOTs), as before, and also the “Lags”, which are specific to this experiment, and which will be used as parametric modulators. The Lags code, for each second presentation of a face (N2 and F2), the number of other faces intervening between this (repeated) presentation and its previous (first) presentation. Both SOTs and Lags are represented by Matlab cell arrays, stored in the `sots.mat` file.

- At the MATLAB command prompt type `load sot`. This loads the stimulus onset times and the lags (the latter in a cell array called `itemlag`).

Now press the SPECIFY 1ST-LEVEL button. This will call up the specification of a fMRI specification job in the batch editor window. Then

- Press “Load” and select the `categorical_spec.mat` job file you created earlier.
- Open “Conditions” and then open the second “Condition”.
- Highlight “Parametric Modulations”, select “New Parameter”.
- Highlight “Name” and enter “Lag”, highlight values and enter `itemlag{2}`, highlight polynomial expansion and “2nd order”.
- Now open the fourth “Condition” under “Conditions”.
- Highlight “Parametric Modulations”, select “New Parameter”.

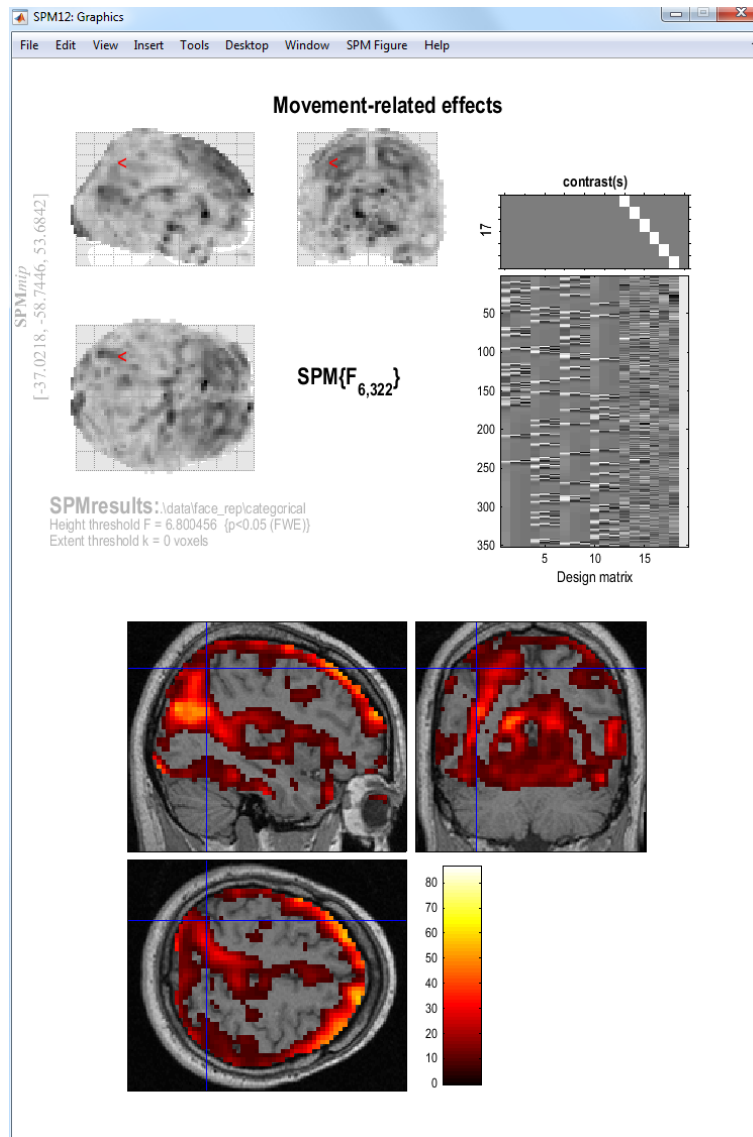


Figure 31.15: *Movement-related activations. These spurious ‘activations’ are due to residual movement of the head during scanning. These effects occur at tissue boundaries and boundaries between brain and non-brain, as this is where contrast differences are greatest. Including these regressors in the design matrix means these effects cannot be falsely attributed to neuronal activity.*

- Highlight “Name” and enter “Lag”, highlight values and enter `itemlag{4}`, highlight polynomial expansion and “2nd order”.
- Open “Canonical HRF” under “Basis Functions”, highlight “Model derivatives” and select “No derivatives” (to make the design matrix a bit simpler for present purposes!).
- Highlight “Directory” and select `DIR/parametric` (having “unselected” the current definition of directory from the Categorical analysis).
- Save the job as `parametric_spec` and press the Run button.

This should produce the design matrix shown in Figure 31.16.

### 31.3.1 Estimate

Press the ESTIMATE button. This will call up the specification of an fMRI estimation job in the batch editor window. Then

- Highlight the “Select SPM.mat” option and then choose the `SPM.mat` file saved in the `DIR/parametric` directory.
- Save the job as `parametric_est.job` and press the Run button.

SPM will write a number of files into the selected directory including an `SPM.mat` file.

### 31.3.2 Plotting parametric responses

We will look at the effect of lag (up to second order, ie using linear and quadratic terms) on the response to repeated Famous faces, within those regions generally activated by faces versus baseline. To do this

- Press “Results” and select the `SPM.mat` file in the `DIR/parametric` directory.
- Press “Define new contrast”, enter the name “Famous Lag”, press the “F-contrast” radio button, enter “1:6 9:15” in the “columns in reduced design” window, press “submit”, “OK” and “Done”.
- Select the “Famous Lag” contrast.
- *Apply masking ? [None/Contrast/Image]*
- Specify “Contrast”.
- Select the “Positive Effect of Condition 1” T contrast.
- Change to an 0.05 uncorrected mask p-value.
- Nature of Mask ? inclusive.
- *p value adjustment to control: [FWE/none]*
- Select None
- *Threshold {F or p value}*
- Accept the default value, 0.001
- *Extent threshold {voxels} [0]*
- Accept the default value, 0.

Figure 31.17 shows the MIP and an overlay of this parametric effect using overlays, sections and selecting the `wmsM03953_0007.nii` image. The effect is plotted in the time domain in figure 31.18. This was obtained by

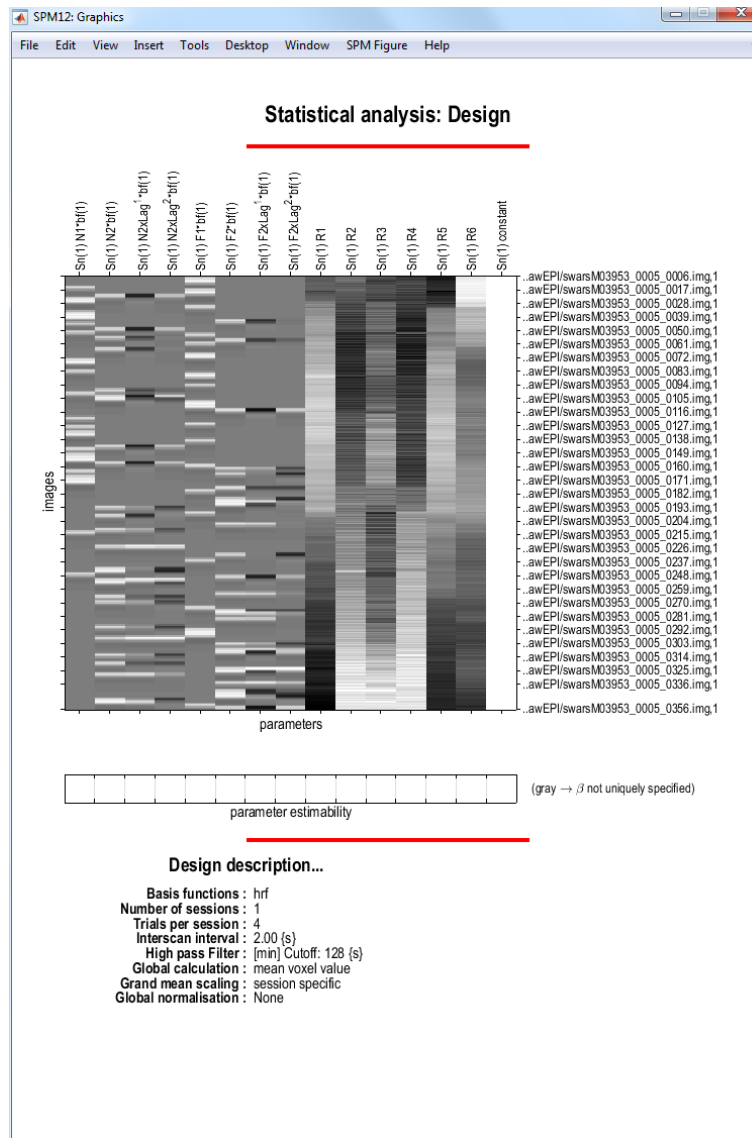


Figure 31.16: *Design matrix for testing repetition effects parametrically.* Regressor 2 indicates the second occurrence of a nonfamous face. Regressor 3 modulates this linearly as a function of lag (i.e. how many faces have been shown since that face was first presented), and regressor 4 modulates this quadratically as a function of lag. Regressors 6, 7 and 8 play the same roles, but for famous faces.

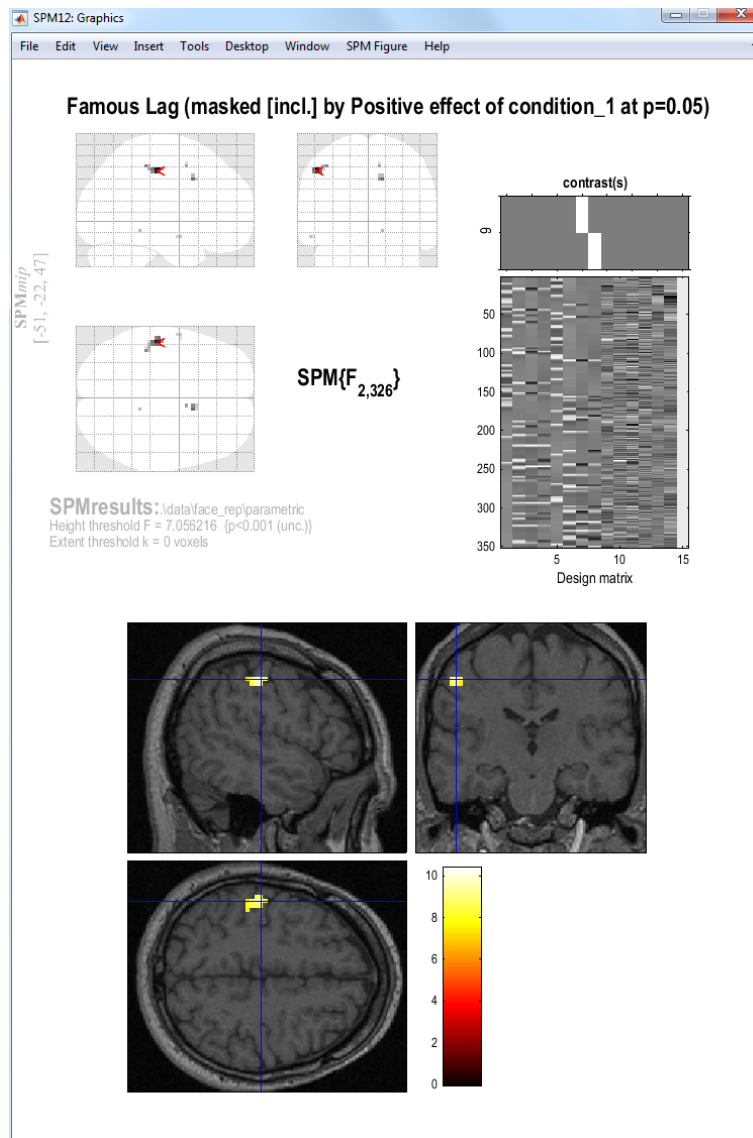


Figure 31.17: MIP and overlay of parametric lag effect in parietal cortex.

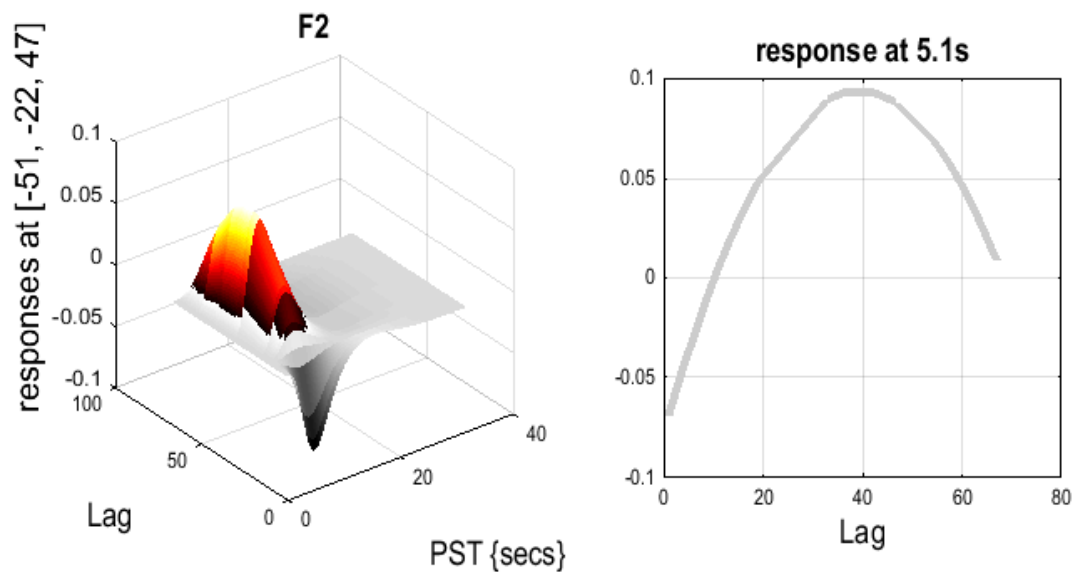


Figure 31.18: *Response as a function of lag.*

- Right clicking on the MIP and selecting “global maxima”.
- Pressing Plot, and selecting “parametric responses” from the pull-down menu.
- Which effect ? select “F2”.

This shows a quadratic effect of lag, in which the response appears negative for short-lags, but positive and maximal for lags of about 40 intervening faces (note that this is a very approximate fit, since there are not many trials, and is also confounded by time during the session, since longer lags necessarily occur later (for further discussion of this issue, see the SPM2 example analysis of these data on the webpage).

## 31.4 Bayesian analysis

### 31.4.1 Specification

Press the SPECIFY 1ST-LEVEL button. This will call up an fMRI specification job in the batch editor window. Then

- Load the `categorical_spec.mat` job file created for the classical analysis.
- Open “Subject/Session”, highlight “Scans”.
- Deselect the smoothed functional images using the ‘unselect all’ option available from a right mouse click in the SPM file selector (bottom window).
- Select the unsmoothed functional images using the `~wa.*` filter and “select all” option available from a right mouse click in the SPM file selector (top right window). The Bayesian analysis uses a spatial prior where the spatial regularity in the signal is estimated from the data. It is therefore not necessary to create smoothed images if you are only going to do a Bayesian analysis.
- Press “Done”.
- Highlight “Directory” and select the `DIR/bayesian` directory you created earlier (you will first need to deselect the `DIR/categorical` directory).

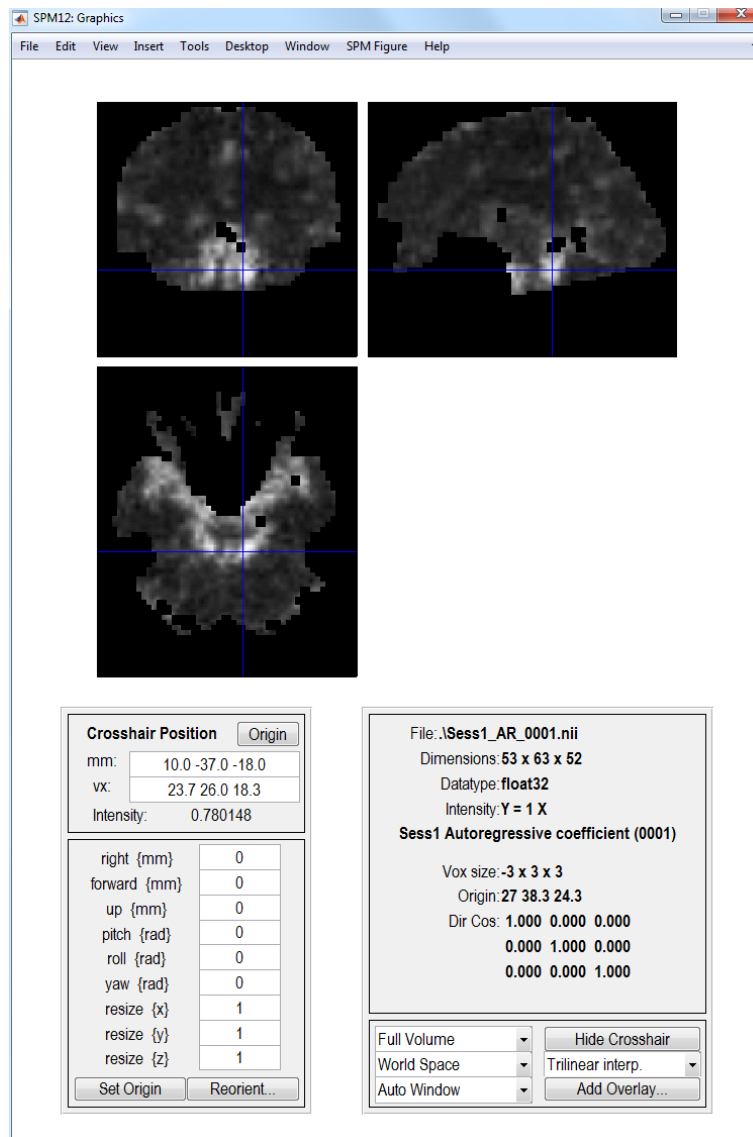


Figure 31.19: Bayesian analysis: Estimated  $AR(1)$  coefficient image indicating heterogeneity near the circle of Willis

- Save the job as `specify_bayesian.mat` and press the Run button.

### 31.4.2 Estimation

Press the ESTIMATE button. This will call up the specification of an fMRI estimation job in the batch editor window. Then

- Highlight the “Select SPM.mat” option and then choose the `SPM.mat` file saved in the `DIR/bayesian` subdirectory
- Highlight “Method” and select the “Choose Bayesian 1st-level” option.
- Save the job as `estimate_bayesian.job` and press the Run button.

SPM will write a number of files into the output directory including

- An `SPM.mat` file.

- Images `Cbeta_k.nii` where  $k$  indexes the  $k$ th estimated regression coefficient. These file-names are prefixed with a “C” indicating that these are the mean values of the “Conditional” or “Posterior” density.
- Images of error bars/standard deviations on the regression coefficients `SDbeta_k.nii`.
- An image of the standard deviation of the error `Sess1_SDerror.nii`.
- An image `mask.nii` indicating which voxels were included in the analysis.
- Images `Sess1_AR_p.nii` where  $p$  indexes the  $p$ th AR coefficient. See eg. Figure 31.19.
- Images `con_i.nii` and `con_sd_i.nii` which are the mean and standard deviation of the  $i$ th pre-defined contrast.

### 31.4.3 Inference

After estimation, we can make a posterior inference using a PPM. Basically, we identify regions in which we have a high probability (level of confidence) that the response exceeds a particular size (eg, % signal change). This is quite different from the classical inferences above, where we look for low probabilities of the null hypothesis that the size of the response is zero.

To determine a particular response size (“size threshold”) in units of PEAK % signal change, we first need to do a bit of calculation concerning the scaling of the parameter estimates. The parameter estimates themselves have arbitrary scaling, since they depend on the scaling of the regressors. The scaling of the regressors in the present examples depends on the scaling of the basis functions. To determine this scaling, load the “SPM.mat” file and type in MATLAB `sf = max(SPM.xBF.bf(:,1))/SPM.xBF.dt` (alternatively, press “Design:Explore:Session 1” and select any of the conditions, then read off the peak height of the canonical HRF basis function (bottom left)).

Then, if you want a size threshold of 1% peak signal change, the value you need to enter for the PPM threshold (ie the number in the units of the parameter estimates) is  $1/sf$  (which should be 4.75 in the present case).<sup>10</sup>

Finally, if we want to ask where is there a signal greater than 1% (with a certain confidence) to faces versus baseline, we need to create a new contrast that takes the AVERAGE of the parameter estimates for the canonical HRF across the four conditions (N1 to F2), rather than the default `Positive effect of condition_1` contrast, which actually calculates the SUM of the parameter estimates for the canonical HRF across conditions (the average vs sum makes no difference for the classical statistics).

- Press “Results”.
- Select the `SPM.mat` file created in the last section.
- Press “Define new contrast”, enter the name “AVERAGE Canonical HRF: Faces > Baseline”, press the “T-contrast” radio button, enter the contrast `[1 0 0 1 0 0 1 0 0 1 0 0]/4`, press “submit”, “OK” and “Done”.
- *Apply masking ? [None/Contrast/Image]*
- Specify None
- *Effect size threshold for PPM*
- Enter the value
- *Log Odds Threshold for PPM*
- Enter the value 10
- *Extent threshold [0]*

<sup>10</sup>Strictly speaking, this is the peak height of the canonical component of the best fitting BOLD impulse response: the peak of the complete fit would need to take into account all three basis functions and their parameter estimates.



- Accept the default value

SPM will then plot a map of effect sizes at voxels where it is 95% sure that the effect size is greater than 1% of the global mean. Then use overlays, sections, select the normalised structural image created earlier and move the cursor to the activation in the left hemisphere. This should create the plot shown in Figure 31.20.

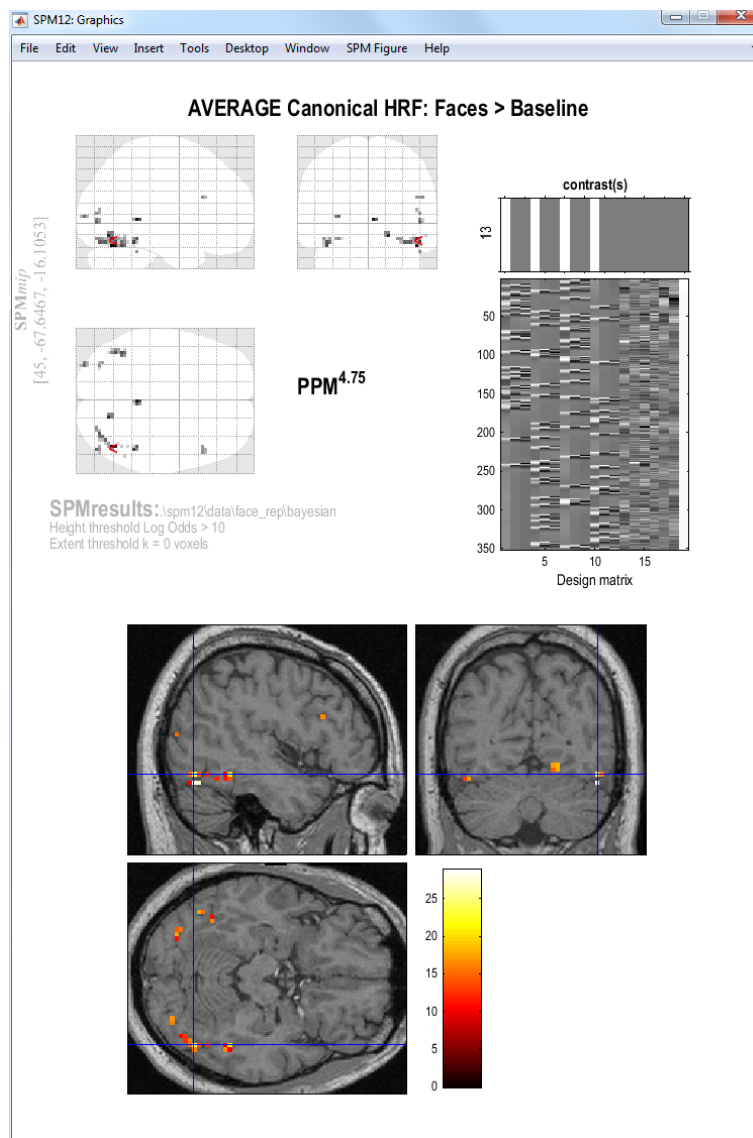


Figure 31.20: Bayesian analysis: MIP and overlay of effect sizes at voxels where PPM is 95% sure that the effect size is greater than 1% of the global mean. The cursor is at the location  $x = 30, y = -82, z = -17\text{mm}$

## Chapter 32

# Face group fMRI data

### 32.1 Introduction

These examples illustrate multisubject “random effects” analyses or “second-level” models of fMRI data [82]<sup>1</sup>. The examples consist of three basic types of 2nd-level model:

1. **M2c**: Using contrast images for the canonical HRF only. This uses a single observation (contrast image) per subject only and data are analysed using a “One-sample t-test”.
2. **M2i**: Using contrast images from an “informed” basis set, consisting of the canonical HRF and its two partial derivatives with respect to time (onset latency) and dispersion. This uses 3 observations (contrast images) per subject and data are analysed using a “One-way ANOVA” with 3 levels.
3. **M2f**: Using contrast images from a very general “Finite Impulse Response” (FIR) basis set, with  $12 \times 2$  second timebins. This uses 12 observations (contrast images) per subject. Data are analysed using a “One-way ANOVA” with 12 levels.

### 32.2 Data

The data come from the “implicit” condition of the Henson et al. study [55]. Although the 1st-level design matrices (and therefore resulting contrast images) used do not correspond exactly to those used in that study.

It is also the same study from which one subject is used to illustrate a single-subject fixed effects analysis (see chapter 31 in this manual).

Unlike the single-subject fixed effects example dataset, only two event-types were modelled: famous and nonfamous faces (initial and repeated presentations were collapsed together, as were correct and incorrect responses). Briefly, greyscale photographs of 52 famous and 52 nonfamous face were presented for 0.5s for fame judgment task (one of two right finger key presses). The minimal SOA (SOA<sub>min</sub>) was 4.5s, with all faces randomly intermixed together with a further 52 null events (ie 2/3 probability of a face every SOA<sub>min</sub>).

Original images were continuous EPI (TE=40ms,TR=2s) 24 descending slices (64×64 3×3 mm<sup>2</sup>), 3mm thick, 1.5mm gap.

2nd-level models **M2c** and **M2i** derive from a 1st-level model (**M1i**), in which the events were modelled with Nf=3 basis functions: the canonical HRF, its partial derivative with respect to onset latency (“temporal derivative”) and its partial derivative with respect to dispersion (“dispersion derivative”).

2nd-level model **M2f** derives from an alternative 1st-level model (**M1f**), in which the same events were modelled with Nf=12 basis functions instead: corresponding to 2s timebins from 0-24s poststimulus (SPM’s “Finite Impulse Response” or FIR basis set).

---

<sup>1</sup>This chapter has been largely cannibalised from an earlier document, available from [http://www.fil.ion.ucl.ac.uk/spm/data/face\\_rfx/spm2\\_face\\_rfx.doc](http://www.fil.ion.ucl.ac.uk/spm/data/face_rfx/spm2_face_rfx.doc), which describes how to analyse this data using SPM2. That document additionally describes the analysis of differential effects, which we have omitted here.

In both first-level models (**M1i** and **M1f**), the contrast images (`con*.img`'s) come from session-specific contrasts within a large (multisession) 1st-level Fixed Effects design matrix, with one session per subject. (Note that the resulting `con*.img`'s could equally well have been produced from 12 separate 1st-level models, one per subject.)

For each type of model, the main effect of faces versus baseline (eg, a  $[0.5 \dots 0.5]$  contrast for each basis function, or `kron([0.5 0.5], eye(Nf))` more generally) was examined.

The 12 (subjects) `con*.imgs` from the 1st-level model using the canonical HRF (**M1c**), the 12 (subjects) x 3 (basis functions) `con*.imgs` from the 1st-level model using the informed basis (**M1i**) set and the 12 (subjects) x 12 (basis functions) x 2 (contrast-types) `con*.imgs` from the 1st-level model using the FIR basis (**M1f**) set are in the zipped file

- [http://www.fil.ion.ucl.ac.uk/spm/download/data/face\\_rfx/face\\_rfx.zip](http://www.fil.ion.ucl.ac.uk/spm/download/data/face_rfx/face_rfx.zip)

Each contrast-type is examined in a separate SPM analysis. This chapter just describes analysis of the main effect of faces versus baseline. To analyse the data, first create a new directory DIR eg. `c:\data\face_group`, in which to place the results of your analysis. Then create 3 subdirectories (i) **Canonical**, (ii) **Informed**, and (iii) **FIR**. As the analysis proceeds these directories will be filled with job-specification files, design matrices and estimated models.

### 32.3 Canonical HRF

For the main effect versus baseline, these happen to correspond to the contrast images numbered 3-14 in 1st-level model **M1i**, ie:

- `con_0006.img` (canonical HRF, subject 1)
- `con_0007.img` (canonical HRF, subject 2)
- ...
- `con_0017.img` (canonical HRF, subject 12)

These images comprise the data for **M2c**, which is simply a “One-sample t-test”. This can be implemented as follows.

- Start up MATLAB and type `spm_fmri` at the prompt
- Press the “Specify 2nd-level” button. This will open the batch editor.
- In the “Design”, “One-sample t-test” option, select “Scans”.
- Choose “Select Files” and use the SPM file selector to choose contrast images 6 to 17.
- Highlight “Directory”, “Select files” and select the subdirectory `canonical`, to place the design matrix in.
- Save the job file as eg. `DIR/canonical.mat`.
- Press the **Run** button (green arrow).

SPM will then show you the design matrix shown in Figure 32.1. This is simply a single column of 1's which will appear as a white box on a white background. This design is encoded in the `SPM.mat` file that is written to the output directory. Then press “Estimate”, select the **SPM.mat** file just created, and press the **Run** button. SPM will now estimate the parameters, that is, the size of the population effect at each voxel. This is simply the average of the `con*.imgs` you have specified.

- Now press the “Results” button.
- Select the `SPM.mat` file.

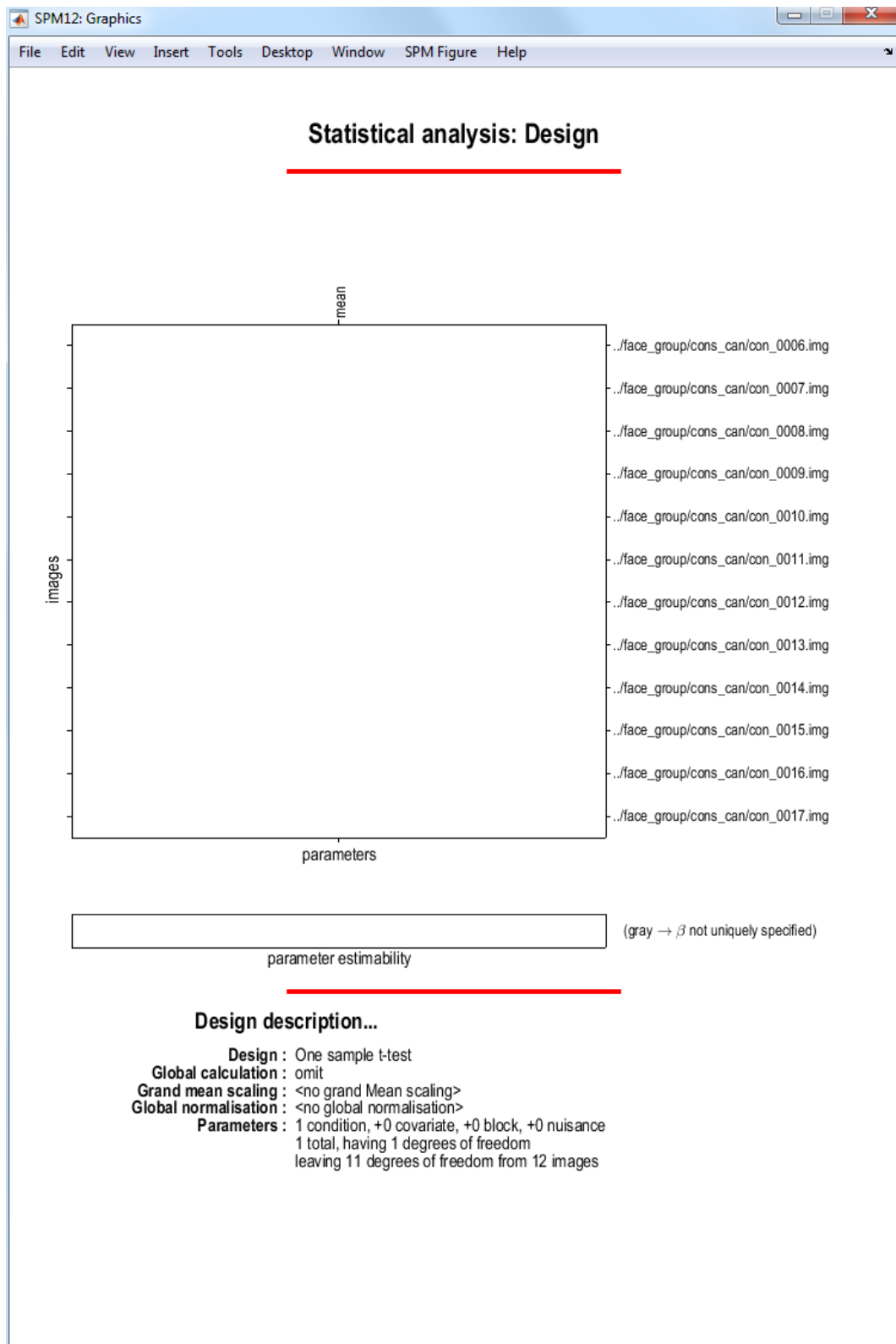


Figure 32.1: *Design matrix for canonical responses. This corresponds to a one-sample t-test.*

- In the contrast manager press “Define new contrast” (select F). Enter [1] in the contrast section and enter “Faces vs Baseline: Canonical HRF” as a “name”. Note: This [1] F-contrast tests for both “activations” and “deactivations” versus the interstimulus baseline, though in the present case, the regions are nearly all activations, as can be seen by entering the same contrast weight [1], but as a T rather than F contrast.
- Press the “submit” button. Press OK.
- Now press the “Done” button.
- Mask with other contrast(s) [No]
- p value adjustment to control [FWE]
- Family-wise p-value [0.05]
- Extent threshold voxels [0]

SPM will now display the thresholded F-statistic image. This shows voxels that are significantly active (correcting for multiple comparisons across all voxels) in the population from which the subjects were drawn. They include bilateral posterior fusiform, SMA, and, at a more liberal threshold, left motor cortex). You can then press the volume to get a table of statistical information for clusters of activated voxels. SPM’s graphics window should look like Figure 32.2.

## 32.4 Informed basis set

For this example, 3 contrast images per subject are taken to the 2nd-level. These are

- `con_0003.img` (canonical HRF, subject 1)
- `con_0004.img` (canonical HRF, subject 2)
- ...
- `con_0014.img` (canonical HRF, subject 12)
- `con_0015.img` (temporal derivative, subject 1)
- `con_0016.img` (temporal derivative, subject 2)
- ...
- `con_0026.img` (temporal derivative, subject 12)
- `con_0027.img` (dispersion derivative, subject 1)
- `con_0028.img` (dispersion derivative, subject 2)
- ...
- `con_0038.img` (dispersion derivative, subject 12)
- ...

These images comprise the data for M2c, which is simply a “One-way ANOVA” with 3-levels. This can be implemented as follows.

- Press the “Specify 2nd-level” button.
- In “Factorial design specification”, highlight “Design” and then choose “Full Factorial”.
- Under “Factors” create a single “New Factor”.
- In this “Factor”, type in “Basis” for Name and enter 3 under “Levels”.

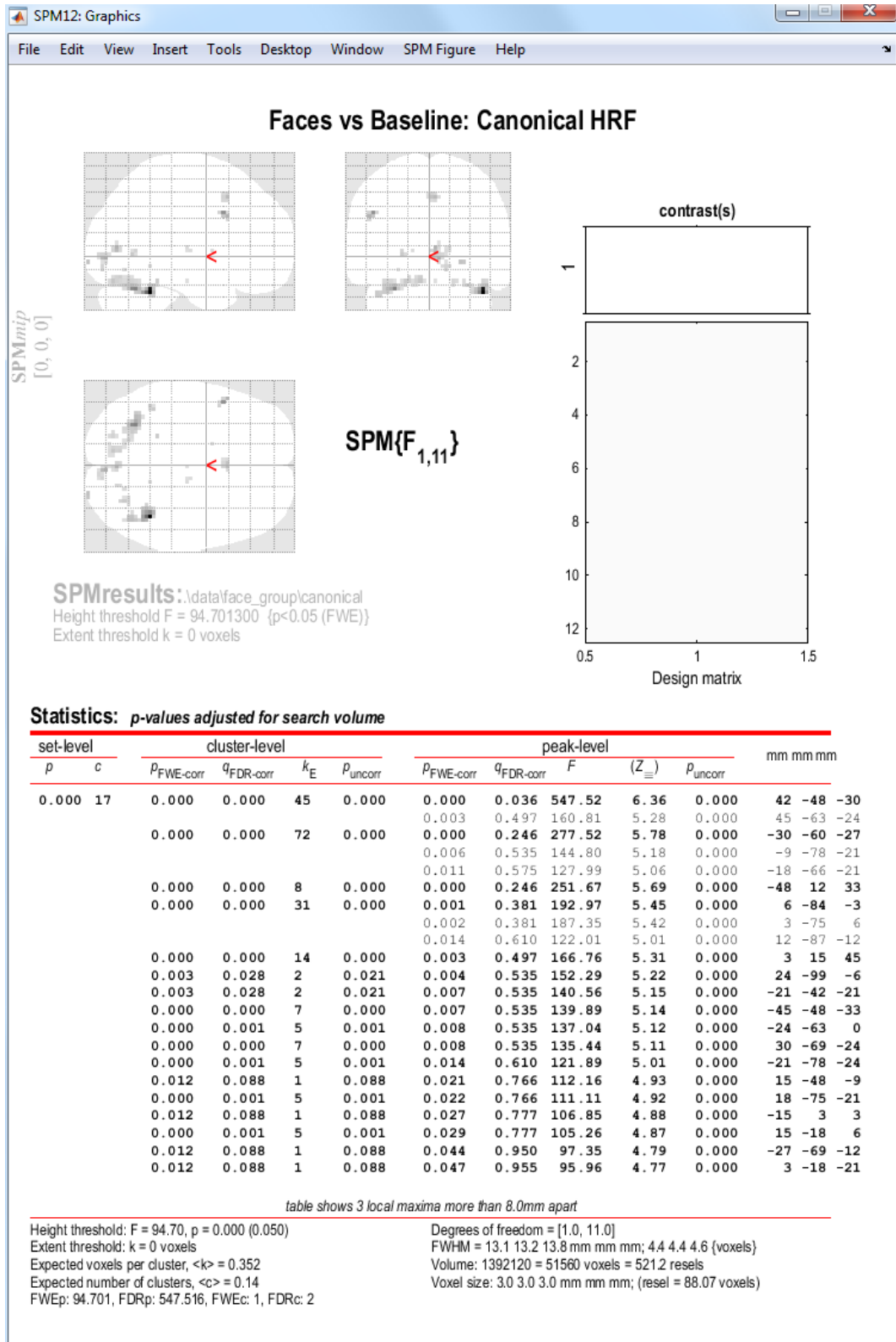


Figure 32.2: Main population effect of faces vs baseline, as characterised using the Canonical HRF.

- Highlight “Independence” and select “No”. SPM will then take into account possible correlations between these repeated measures (see section on Nonsphericity below for further discussion).
- Now highlight “Specify cells”, and create 3 new cells.
- For the first cell, set “Levels” to 1, and enter the canonical contrast images under scans (ie contrast images numbered 0003 to 0014).
- For the second cell, set “Levels” to 2, and enter the temporal derivative contrast images under scans (ie contrast images numbered 0015 to 0026).
- For the third cell, set “Levels” to 3, and enter the dispersion derivative contrast images under scans (ie contrast images numbered 0027 to 0038).
- Highlight “Directory”, “Specify files” and select the subdirectory “informed”, to place the design matrix in.
- Save the job file as eg. DIR/informed.mat.
- Press the Run button in the batch editor.

SPM will then show you the design matrix shown in Figure 32.3. This design is encoded in the `SPM.mat` file that is written to the output directory. Then press “Estimate”, select the `SPM.mat` file just created, and press the Run button. SPM will now estimate the parameters of the model (and hyperparameters governing the nonsphericity).

### 32.4.1 Nonsphericity

Setting the independence option described above to “No” allows SPM to take into account possible correlations between levels of the factor. Note that, by default, SPM assumes different variances for different levels of the factor (you can change this by setting “Variance” to “Equal” under the options for the factor).

In this way SPM can account for possible “non-sphericity” in the data. This is implemented in SPM using a set of matrices (bases) that characterise the covariance matrix. The first three correspond to the variance of each of the canonical, temporal and dispersion derivatives: `SPM.xVi.Vi{1}`, `SPM.xVi.Vi{2}`, and `SPM.xVi.Vi{3}`.

The next three correspond to covariances: `SPM.xVi.Vi{4}` (covariance between canonical and temporal derivative), `SPM.xVi.Vi{5}` (covariance between canonical and dispersion derivative), and `SPM.xVi.Vi{6}` (covariance between temporal and dispersion derivatives).

After estimation the actual covariance values (hyper-parameters) are given by `SPM.xVi.h` (the six entries correspond to the above bases). The corresponding estimated covariance matrix can be shown by pressing Review→Design→Explore→Covariance Structure. The estimated covariance for this data is shown in Figure 32.4. Note that these are “global” values which are scaled by a voxel specific-value to achieve a model covariance that best matches the empirical covariance at each voxel.

### 32.4.2 Informed Results

- Now press the “Results” button.
- Select the `SPM.mat` file.
- In the Contrast Manager press “Define new contrast” (select F). Enter `eye(3)` in the contrast section and enter “Faces vs Baseline: Informed” as a “name”. Note: In MATLAB `eye(3)` evaluates to the identity matrix  $[1\ 0\ 0; 0\ 1\ 0; 0\ 0\ 1]$ .<sup>2</sup>
- Press the “submit” button. Press OK.

<sup>2</sup>SPM will have produced some contrasts automatically, one of them being the “main effect of basis”. This contrast is, however, not appropriate for our purposes.



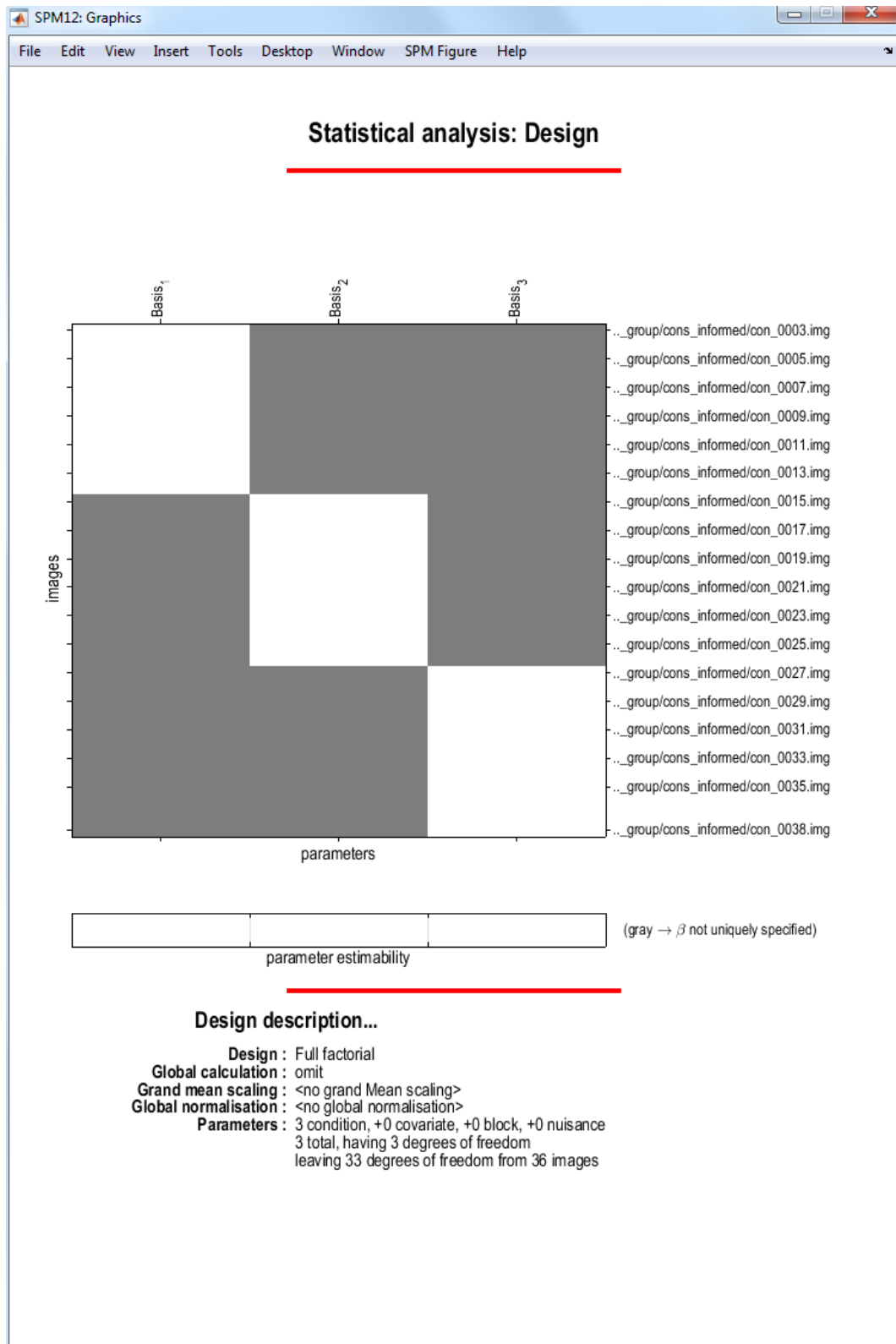


Figure 32.3: *Design matrix for informed basis set.* This corresponds to a one-way ANOVA with three levels (but no constant term, since we want to test whether the basis functions are different from zero, not whether they are different from each other).

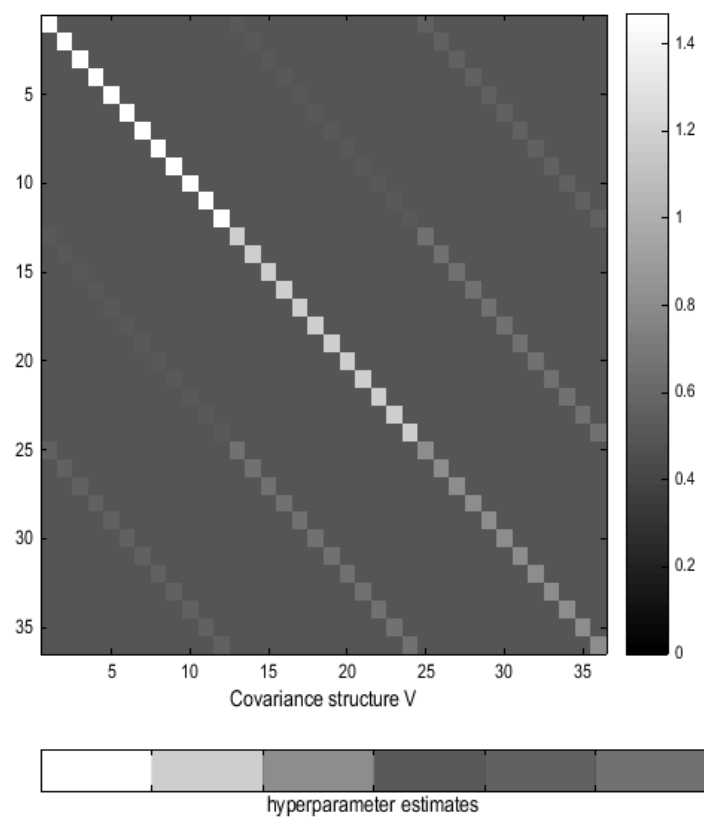


Figure 32.4: *Estimated covariance matrix for informed basis set.* The 6 differently valued hyperparameters are shown in different shades of gray.

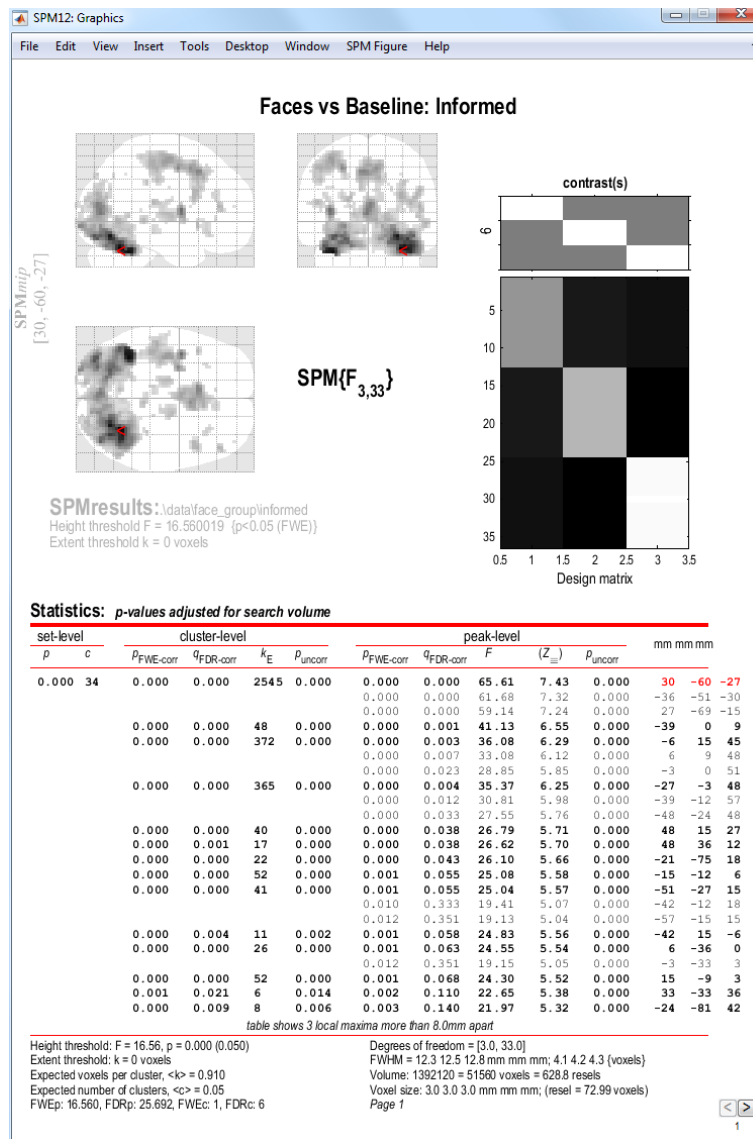


Figure 32.5: Main population effect of faces, as characterised with the informed basis set.

- Now press the “Done” button.
- Mask with other contrast(s) [No]
- p value adjustment to control [FWE]
- Family-wise p-value [0.05]
- Extent threshold voxels [0]

This contrast will reveal voxels that show some form of event-related response that can be captured by (ie, lies in the space spanned by) the three basis functions (e.g, 30 -60 -27, Z=7.43), as shown in Figure 32.5.

Note how the design matrix appears to be different after estimation. This is because it has been pre-whitened (via the estimated nonsphericity). In particular, the (barely visible) off-diagonal entries in the design matrix give an indication of the degree of correlation between the basis functions across subjects. However, because the data have also been pre-whitened our interpretation of the parameter estimates (the “betas”) is unchanged. Effectively the parameters

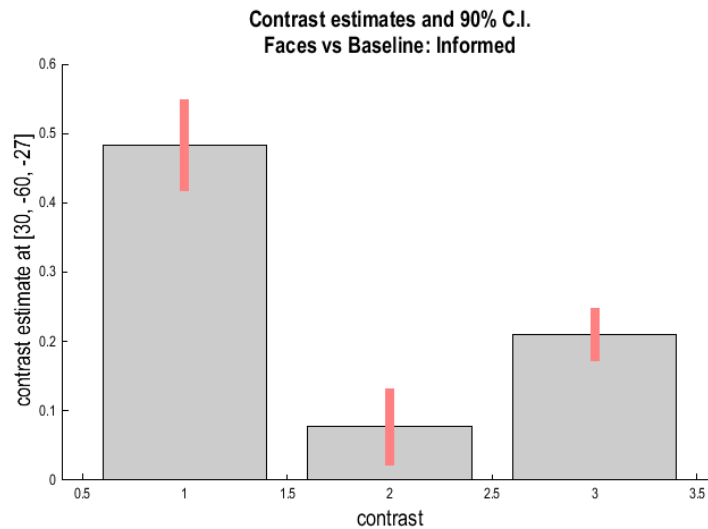


Figure 32.6: Plotting the three basis functions for the global maximum showing reliable effects of the canonical HRF and its time and dispersion derivatives.

have been estimated using “Weighted Least Squares (WLS)”, where the weights relate to the estimated error covariance structure. SPM implements WLS by pre-whitening the data and the design matrix and then using “Ordinary Least Squares” (OLS).

Note also how this F-contrast (Figure 32.5) produces more significant results than the corresponding F-contrast in the model with the canonical HRF shown in Figure 32.2. This suggests significant additional information in the two derivatives of the canonical HRF. If you right-click on the MIP and select “goto global maxima”, then press “plot”, select “Contrast estimates and 90% C.I.”, and select the “Faces vs Baseline: Informed” contrast, you will get three bars and their confidence intervals, as in Figure 32.6. You can see that the canonical HRF (first bar) carries most of the response vs baseline, but nonetheless, both the temporal and dispersion derivatives (second and third bars) contribute significant additional effects (given that the error bars do not overlap zero). Note that the size of the bars cannot be compared directly since they depend on the (different) scaling of the three basis functions (their size RELATIVE TO the error bars is a fairer way to compare the contributions of the different basis functions).

### 32.4.3 T- and F-contrasts

It is also informative to evaluate the T-contrast  $[1\ 0\ 0]$  (ie positive loadings on the canonical HRF only). This is shown in Figure 32.7.

At a FWE correct p-value of 0.05, note more voxels (including now left motor cortex) and higher Z-values (e.g, 39 -57 -30,  $Z=7.53$ ) for this main effect vs baseline compared to the equivalent T-contrast ( $[1]$ ) in the model that uses only the canonical HRF (as in previous Section). The main reason for this increased power is the increase in the degrees of freedom, which entails better estimators of the underlying error (co)variance. The price of this increased power is a stronger assumption about the nonsphericity, namely that it has the same structure across (activated) voxels - the “pooling device”, see Glaser et al. (2003) [49].

Finally, evaluate the F-contrasts  $[0\ 1\ 0]$  and  $[0\ 0\ 1]$ . These are shown in Figures 32.8 and 32.9. These contrasts reveal voxels that load (positively or negatively) on the temporal and dispersion derivatives respectively. These contrasts reveal that there is significant variability (at  $p < .05$  corrected) that is not captured by the canonical HRF alone (see below for more discussion; see also to Henson et al (2000) [54].

In other words, some regions have earlier or later, or wider or narrower, BOLD impulse responses than the canonical HRF. This may reflect differences in vasculature (or even face-related neural differences across regions).

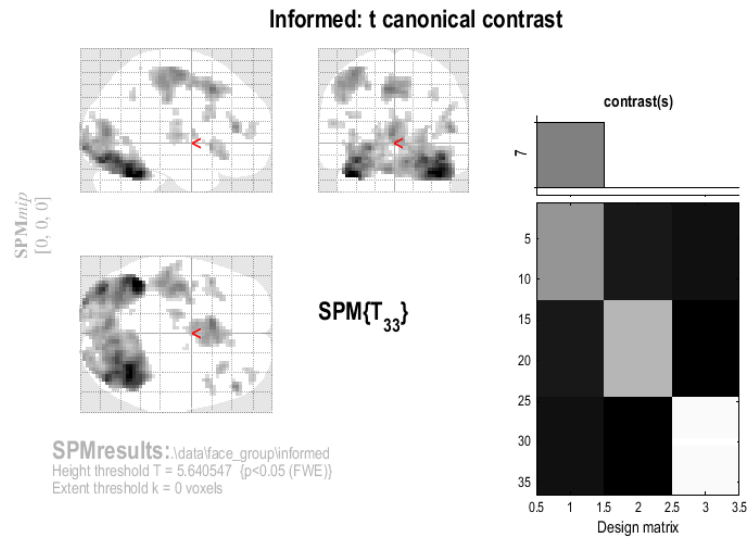


Figure 32.7: Main population effect of faces, as characterised with the canonical HRF using a  $[1\ 0\ 0]$   $t$ -contrast on the informed basis coefficients.

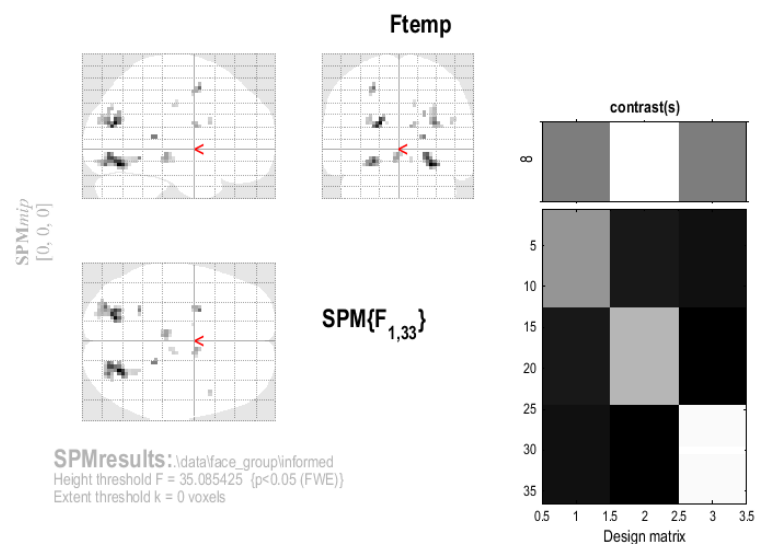


Figure 32.8: Significantly non-zero temporal derivative coefficients. These voxels show responses earlier or later than canonical responses.

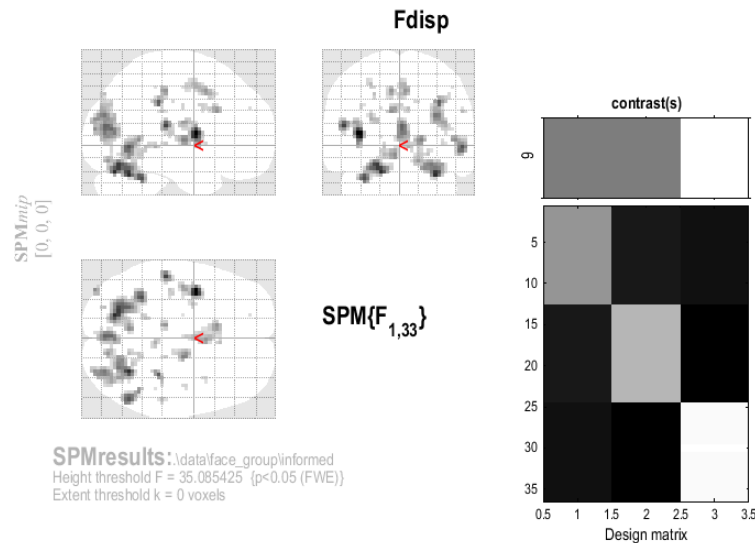


Figure 32.9: *Significantly non-zero dispersion derivative coefficients. These voxels show responses narrower or wider than canonical responses.*

On the other hand, note that most voxels in the above F-contrasts also show a positive loading on the canonical HRF (ie the previous  $[1\ 0\ 0]$  T-contrast), as can be revealed by Inclusive (or Exclusive) masking of the relevant contrasts. This is because the loadings on the derivatives reflect deviations ABOUT the canonical form (via a first-order Taylor expansion; see eg. Henson et al, 2002 [53]). Indeed, loadings on either derivative in the absence of a reliable loading (positive or negative) on the canonical HRF would be difficult to interpret (i.e, the derivative waveforms are probably too high frequency to reflect BOLD changes on their own).

One can also confirm this by going to various voxels in the above F-contrasts, pressing “plot”, “contrast estimates” and selecting the “Can+Tem+Dis” F-contrast. The three bars indicate the loadings (and 90% confidence intervals) on the three different basis functions. Note that a positive estimate for the temporal derivative corresponds to an earlier response than the canonical (and negative for later), while a positive estimate for the dispersion derivative corresponds to a narrower (less dispersed) response (and negative for wider).

## 32.5 FIR basis set

For this example, 12 contrast images per subject are taken to the 2nd-level. These are the contrast images:

- `con_fir_bin01_sub01.img` (FIR bin 1, subject 1)
- `con_fir_bin01_sub02.img` (FIR bin 1, subject 2)
- ...
- `con_fir_bin02_sub01.img` (FIR bin 2, subject 1)
- ...

These images comprise the data for `M2f`, which is simply a “One-way ANOVA” with 12-levels (one for each time-bin). This can be implemented as follows.

- Start up MATLAB and type `spm_fmri` at the prompt.
- Press the “Specify 2nd-level” button.

- The options for “Factorial design specification”<sup>3</sup> appear.
- Highlight “Design” and then choose “Full Factorial”.
- Under “Full Factorial” and ‘Factors’ create a single “New Factor”.
- In this “Factor”, type in “TimeBin” for “Name” and enter 12 under “Levels”.
- Highlight “Independence” and select “No”. SPM will then take into account possible correlations between these repeated measures.
- Now highlight “Specify cells”, and create 12 new cells.
- For the first cell, set “Levels” to 1, and enter the contrast images for time bin 1 under scans. This is most easily done by changing the filter to `.*fir_bin01.*`.
- For the second cell, set “Levels” to 2, and, under scans, enter the contrast images for time bin 2. This is most easily done by changing the filter to `.*fir_bin02.*`.
- Similarly for Levels 3 to 12.
- Highlight “Directory”, “Specify files” and select the subdirectory FIR, to place the design matrix in.
- Save the job file as eg. `DIR/fir.mat`.
- Press the Run button in the batch editor.

SPM will then show you the design matrix shown in Figure 32.10. This design is encoded in the `SPM.mat` file that is written to the output directory. Then press “Estimate”, select the `SPM.mat` file just created, and press the button Run. SPM will now estimate the parameters of the model.

### 32.5.1 Nonsphericity again

Setting the independence option to “No” allows SPM to take into account possible correlations between levels of the factor. Note that, by default, SPM assumes different variances for different levels of the factor (you can change this by setting “Variance” to “Equal” under the options for the factor).

In this way SPM can account for possible “non-sphericity” in the data. This is implemented in SPM using a set of matrices (bases) that characterise the covariance matrix. The first 12 correspond to the variance of each of the responses in each of the 12 time bins. The ones that follow correspond to covariances between different time bins.

After estimation the actual covariance values (hyper-parameters) are given by `SPM.xVi.h`. The corresponding estimated covariance matrix can be shown by pressing Review→Design→Explore→Covariance Structure. The estimated covariance for this data is shown in Figure 32.11. Note that these are “global” values which are scaled by a voxel specific-value to achieve a model covariance that best matches the empirical covariance at each voxel.

You can see the highest values on the leading diagonal occur for timebins 2-4 (scans 13-48). This is where the peak response occurs, and the large values imply that, as expected, the variance tends to increase with the mean. This “inhomogeneity of variance” is a problem for conventional ANOVAs, but not here, where it is explicitly modelled.

Notice also the high values close to the diagonal, which reflect the positive correlation between the error across adjacent timebins (as also expected).

<sup>3</sup>In SPM2, this data was analysed using the “One-way ANOVA without a constant” design. This option is no longer available in SPM5, as one-way ANOVA’s are considered as factorial designs with a single factor.

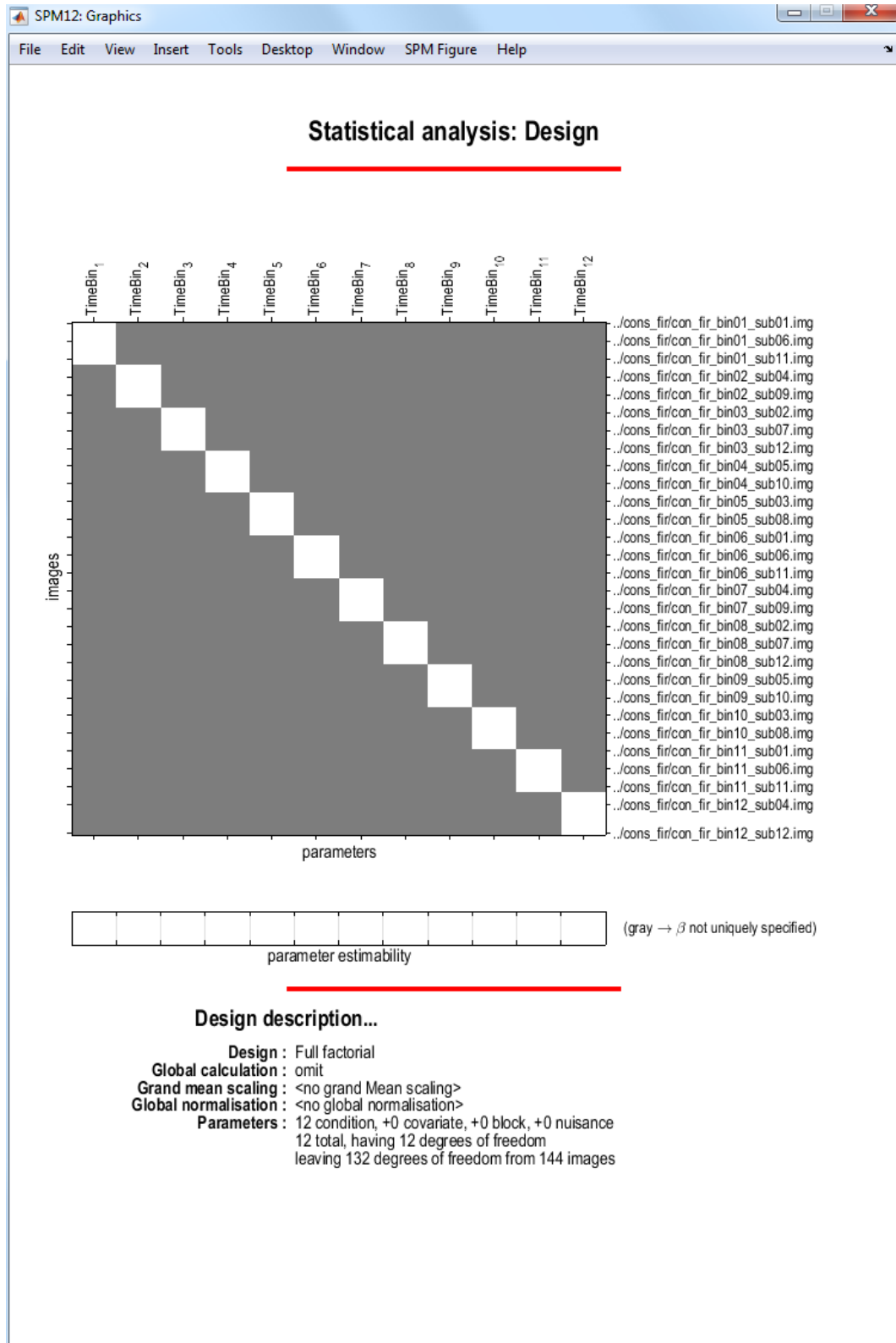


Figure 32.10: Design matrix for FIR basis set. This corresponds to a one-way ANOVA with 12 levels.



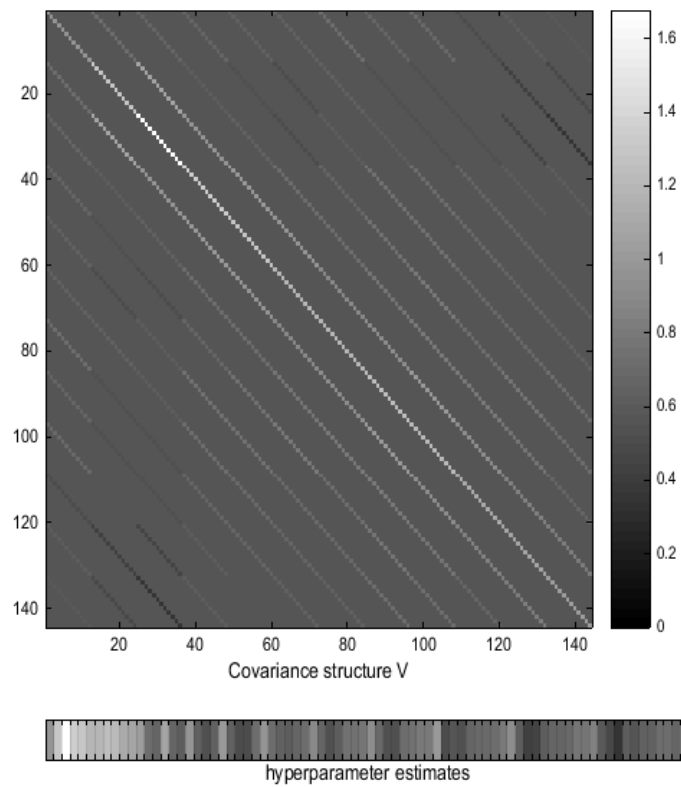


Figure 32.11: *Estimated covariance matrix for FIR basis set. The differently valued hyperparameters are shown in different shades of gray. Notice that the most variable responses occur in the third time bin (scans 25 to 36) corresponding to responses 4-6 seconds post stimulus, ie. at the peak of the hemodynamic response, as expected.*

### 32.5.2 FIR Results

- Now press the “Results” button.
- Select the `SPM.mat` file.
- In the contrast manager press “Define new contrast” (select F). Enter `eye(12)` in the contrast section and enter “Faces vs Baseline: FIR” as a “name”.<sup>4</sup>
- Press the “submit” button. Press OK.
- Now press the “Done” button.
- Mask with other contrast(s) [No]
- p value adjustment to control [FWE]
- Family-wise p-value [0.05]
- Extent threshold voxels [0]

Note how the design matrix, shown in Figure 32.12 appears to be different after estimation. This is because it has been pre-whitened. In particular, the off-diagonal entries in the design matrix give an indication of the degree of correlation between the time bins across subjects (this is displayed explicitly in the covariance matrix in Figure 32.11).

The above contrast will reveal voxels that show *any* form of event-related response, within the range 0-24s post-stimulus and with 2s resolution, as shown in Figure 32.12. Selecting a voxel and plotting this contrast (using the `plot` button) will reveal that most voxels have a fairly “canonical” shape over the 12 timebins. One can also test for more constrained shapes of event-related responses within this model. For example, one can test for “canonical-shaped” responses by evaluating a contrast whose weights trace out SPM’s canonical HRF (every 2s). To do this, switch to the MATLAB window for a moment and type:

- `xBF.dt = 1`
- `xBF.name = 'hrf (with time and dispersion derivatives)';`
- `xBF.length = 32;`
- `xBF.order = 1;`
- `xBF = spm_get_bf(xBF);`

This returns the canonical and two derivatives in the matrix `xBF.bf` (type `help spm_get_bf` for more info), with one value every 1 second. For convenience, then define:

- `all = xBF.bf(2:2:24,:)';`
- `can = all(1,:);`
- `tem = all(2,:);`
- `dis = all(3,:);`

These commands downsample the basis functions every 2s, which is the bin-width of the FIR. If you type `corrcoef(all')`, you will see that the basis functions are slightly correlated (in the off-diagonal terms), due to this undersampling every 2s.

- In the contrast manager press “Define new contrast” (select T).
- Enter `can` as the contrast weights (defined in MATLAB workspace as above), and “Can-weighted FIR” as the name.

---

<sup>4</sup>SPM will have produced some contrasts automatically, one of them being the “main effect of TimeBin”. This contrast is, however, not appropriate for our purposes.

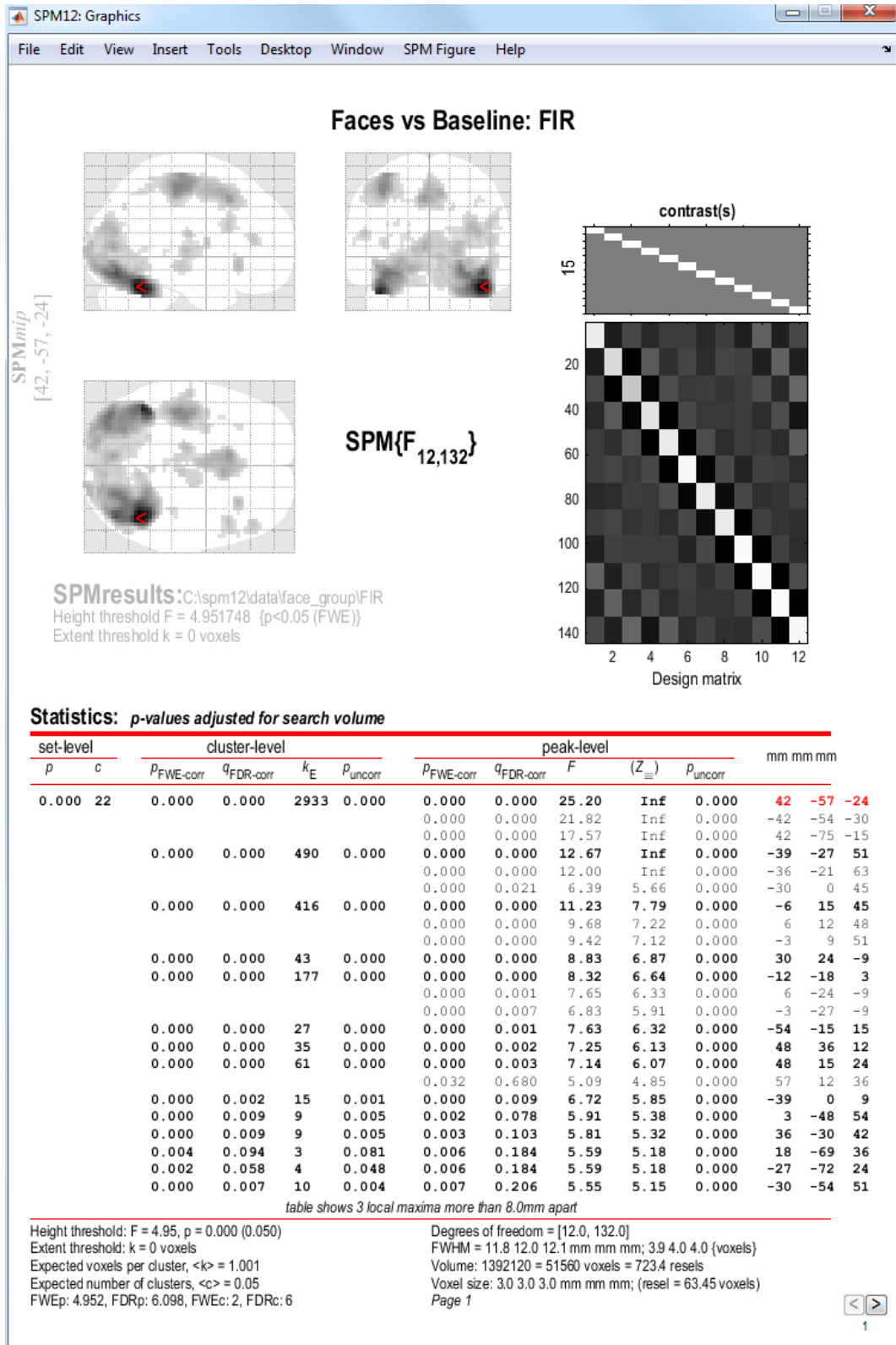


Figure 32.12: Main population effect of faces, as characterised with the FIR basis set.

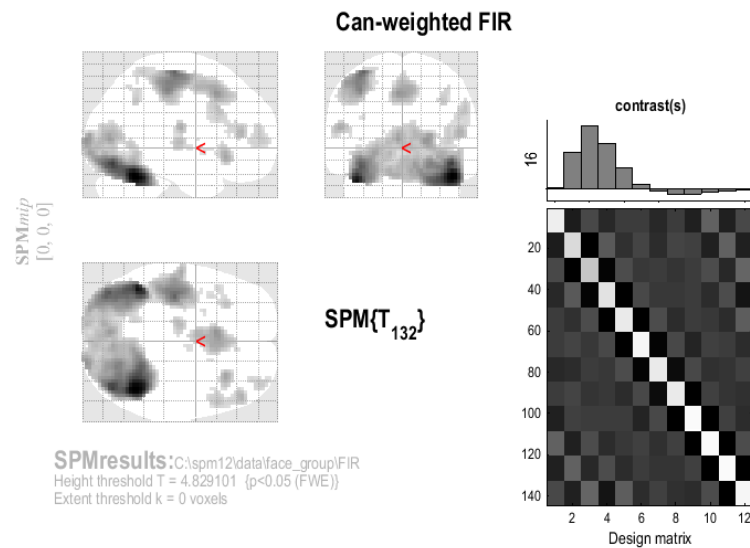


Figure 32.13: Main population effect of faces, as characterised with a canonically weighted contrast of FIR bases.

This produces the MIP in Figure 32.13. At a FWE correct  $p$  value of 0.05, there are many more voxels compared to the equivalent T-contrast [1] in the model using only canonical HRF. The main reason for this increased power is again the increase in the degrees of freedom, which entails better estimators of the underlying error (co)variance (though if the FIR parameters were estimated very inefficiently, the extra contrast images might add more noise, outweighing any advantage of higher degrees of freedom). Again, this increased power comes with a stronger assumption about the nonsphericity, namely that it has the same structure across (activated) voxels [49]. One can also test the variance captured by the temporal and dispersion derivatives by creating new contrasts (though as F rather than T contrasts) and simply typing “tem” and “dis” respectively as the contrast weights.

More interesting is the ability to ask, within this model, how much event-related variance is *not* captured by the canonical HRF. To do this, first create the variable in MATLAB

- `nullcan = eye(12) - pinv(can)*can;`

This creates a matrix for an F-contrast that spans the “null space” of the canonical HRF.

- In the contrast manager press “Define new contrast” (select F).
- Enter `nullcan` as the contrast weights (defined in MATLAB workspace as above), and “Null space of canonical HRF” as the name.

You can see, in Figure 32.14 that several regions express variability not captured by the canonical HRF. This is not surprising, because you will notice that many of these regions appeared in the individual F-tests on the temporal and dispersion derivatives above, suggesting that what is not captured by the canonical HRF is captured by its two derivatives.

Yet even more interesting is the ability to ask how much event-related variance is *not* captured by the canonical HRF or its two derivatives (ie. not captured by SPM’s ‘informed’ basis set). To do this, first create the variable in MATLAB

- `nullall = eye(12) - pinv(all)*all;`

This creates a matrix for an F-contrast that spans the “null space” of all three informed basis functions.

- In the contrast manager press “Define new contrast” (select F).

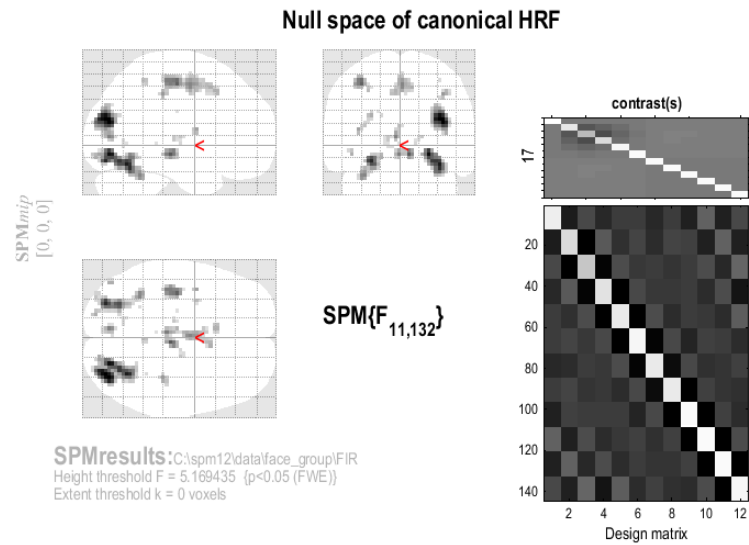


Figure 32.14: *Regions expressing variability across subjects not captured by canonical HRF.*

- Enter `nullall` as the contrast weights (defined in MATLAB workspace as above), and “Null space of informed basis set” as the name.

You will see, in Figure 32.15 that only 2 voxels (in one cluster with maximum -21 -18 27) express variability not captured by the informed basis set. This reinforces the point that, while there is certainly variability in the HRF across different brain regions, the canonical HRF and its two derivatives are sufficient to capture the majority of this regional variability (at least on average across the 12 subjects in this dataset). See [54] for further details.

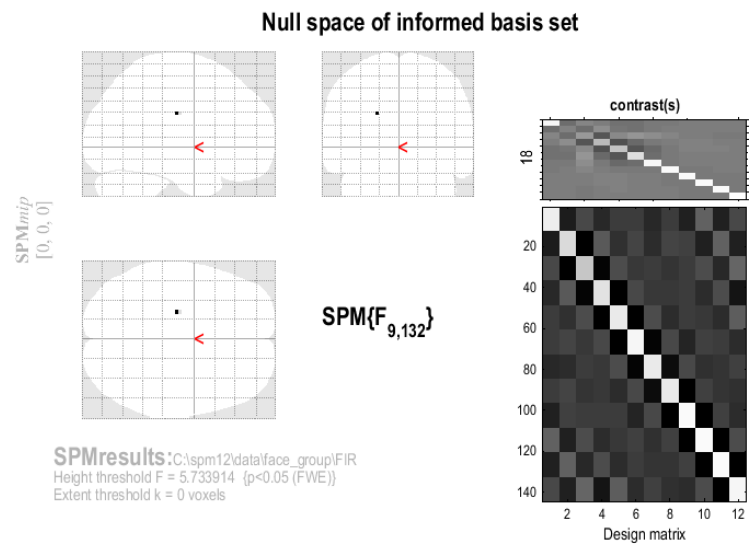


Figure 32.15: *Regions expressing variability across subjects not captured by informed basis set.*

## Chapter 33

# Mixed Effects Analysis

### 33.1 Introduction

This chapter describes Mixed Effects (MFX) analysis of fMRI data. The algorithm on which this is based is described fully in [43].

Before doing an MFX analysis you will need to have previously implemented a within-subject analysis for each subject. See, for example, chapter 31 for how to do this. If you have 12 subjects you should have 12 separate within-subject SPM analyses.

The results of these within-subject analyses can then be used in a summary statistic approach to random effects inference. This entails using contrasts from the group of subjects as data in a 'second-level' design as described in the previous chapter.

Alternatively you can implement a Mixed Effects (MFX) analysis. There are five steps to a MFX analysis.

1. **Specify FFX** This step will create a Fixed Effects model with data from all subjects, and a single design matrix comprising partitions for each subject. In the SPM batch editor (press the Batch button in the SPM top left (command) window) go to *SPM, Stats, Mixed Effects Analysis, FFX Specification*. Select the directory where you want the FFX model to be saved - we will refer to this as DIR. Then select the SPM.mat files that contain the analyses for each individual subject. If you have 12 subjects you should select 12 SPM.mat files.

It is essential that the design matrices contained in the SPM.mat files have the same number of columns. More specifically it is required that, over subjects, there be the same number of sessions, conditions per session, and columns per condition (eg parametric/time modulators if any). This information is written out to the command window so, in the event of an error, you can see which subjects are the odd ones out.

2. **Estimate FFX** This is a very simple step. Press the Estimate button in the top-left (command) SPM window, select the DIR/SPM.mat file created in the previous step, and then press the green play button. SPM will now estimate the group FFX model.
3. **Specify MFX** In the SPM batch editor go to *SPM, Stats, Mixed Effects Analysis, MFX Specification*. Select the DIR/SPM.mat file that was estimated in the previous step, and then press the green play button.

SPM will now specify a second level model that is equivalent to a two-level hierarchical model. This equivalence is derived in equation four of [43]. The second level model comprises (a) a second level design matrix, (b) data, which are the regression coefficient images from the estimated FFX model and (c) an error covariance matrix, whose variance components are computed using the Restricted Maximum Likelihood (ReML) algorithm.

It is the structure of this covariance matrix that makes an MFX analysis different from the alternative summary statistic implementation of RFX. The difference is that the MFX error covariance structure also contains a contribution from within-subject errors from the first level that have been projected through (the pseudo-inverse of) the first level design matrix.

SPM will create a subdirectory DIR/mfx and place an SPM.mat file there. This is the MFX model.

4. **Estimate MFX** This is a very simple step. Press the Estimate button in the top-left SPM window, select the DIR/mfx/SPM.mat file created in the previous step, and then press the green play button. SPM will now estimate the MFX model.
5. **Results** The estimated MFX model can be interrogated in the usual way. Press the Results button in the SPM command window. This will bring up the SPM contrasts manager, where you can specify effects to test as described in previous chapters.



# Chapter 34

## Verbal Fluency PET data

### 34.1 Introduction

These data come from a 5 subject PET study of a verbal fluency with two alternating word generation conditions: A (baseline) - word shadowing; B - (activation) - paced orthographic word generation. This involved responding with a word beginning with an aurally presented letter. Both conditions were identically paced at 1 word every 2 seconds. The presentation order alternated between AB and BA across subjects as shown in Table 34.1.

Scan:	1	2	3	4	5	6	7	8	9	10	11	12
Subject 1	A	B	A	B	A	B	A	B	A	B	A	B
Subject 2	B	A	B	A	B	A	B	A	B	A	B	A
Subject 3	A	B	A	B	A	B	A	B	A	B	A	B
Subject 4	B	A	B	A	B	A	B	A	B	A	B	A
Subject 5	A	B	A	B	A	B	A	B	A	B	A	B

Table 34.1: *Conditions for PET data: (A) word shadowing and (B) word generation.*

The files are named `./p#/snrp#_##.{img,hdr}` and are SPM compatible (Analyze) images following realignment, normalization and smoothing with a 16mm isotropic Gaussian kernel with # indicating the subject and ## the scan. The data set is available from the SPM website<sup>1</sup>.

To analyse the data, first create a new directory DIR, eg. `c:\data\pet`, in which to place the results of your analysis. Then create 4 subdirectories (i) `single`, (ii) `subject-condition`, (iii) `subject-time` and (iv) `multiple`. As the analysis proceeds these directories will be filled with job-specification files, design matrices and estimated models.

### 34.2 Single subject

Firstly, we will analyse the data from a single subject. This can be implemented as follows.

- Start up MATLAB and type `spm pet` at the prompt
- Press the “Basic models” button.
- In ‘Factorial design specification’, choose the ‘Flexible Factorial’ design.
- Highlight ‘Factors’ and create a new Factor and enter ‘Word’ for the name.
- Then, under ‘Specify Subject or all Scans and Factors’, highlight ‘Subjects’ and create a new subject.

<sup>1</sup>Verbal Fluency PET dataset: <http://www.fil.ion.ucl.ac.uk/spm/data/fluency/>

- Highlight ‘Scans’, select ‘Specify Files’ and use the SPM file selector to choose the 12 images for that subject. This can be most easily achieved by specifying ‘.\*snrp1.\*’ as a filter in the file selector.
- Under ‘Conditions’ enter the vector [1 2 1 2 1 2 1 2 1 2 1 2].
- Under ‘Main effects and interactions’, create a single main effect with factor number equal to 1
- Under ‘Covariates’, create a new covariate and enter ‘Time’ for ‘Name’ and the vector ‘1:12’.
- Under ‘Global calculation’ choose ‘Mean’
- Under Global normalisation and Normalisation, choose ‘Proportional’ scaling.<sup>2</sup>
- Under Global normalisation and Overall grand mean scaling, select YES.
- Highlight Directory, Specify files and select the subdirectory ‘single’, to place the design matrix in.
- Save the job file as eg. DIR/single\_design.mat.
- Press the Run button (green arrow).

SPM will then show you the design matrix shown in Figure 34.1. This design is encoded in the SPM.mat file that is written to the output directory. Then press ‘Estimate’ and select the SPM.mat file just created. SPM will now estimate the parameters, that is, the size of the population effect at each voxel.

- Now press the ‘Results’ button.
- Select the SPM.mat file.
- In the contrast manager press ‘Define new contrast’ (select T). Enter [-1 1] in the contrast section and enter ‘activation’ as a ‘name’.
- Press the ‘submit’ button. Press OK.
- Now press the ‘Done’ button.
- Mask with other contrast(s) [No]
- p value adjustment to control [FWE]
- Family-wise p-value [0.05]
- Extent threshold voxels [0]

You should see a blank MIP as, sadly, we rarely have enough sensitivity to find activations in single subject PET data. This is why we scan multiple subjects.

### 34.3 Multiple subjects

The data set can be analysed in several ways which are discussed in [61].

---

<sup>2</sup>Normalisation using ANCOVA is advised for multi-subject studies unless differences in global flow are large eg. due to variability in injected tracer dose. Because ANCOVA uses one degree of freedom for each subject/group, proportional scaling may be preferable for single-subject studies.

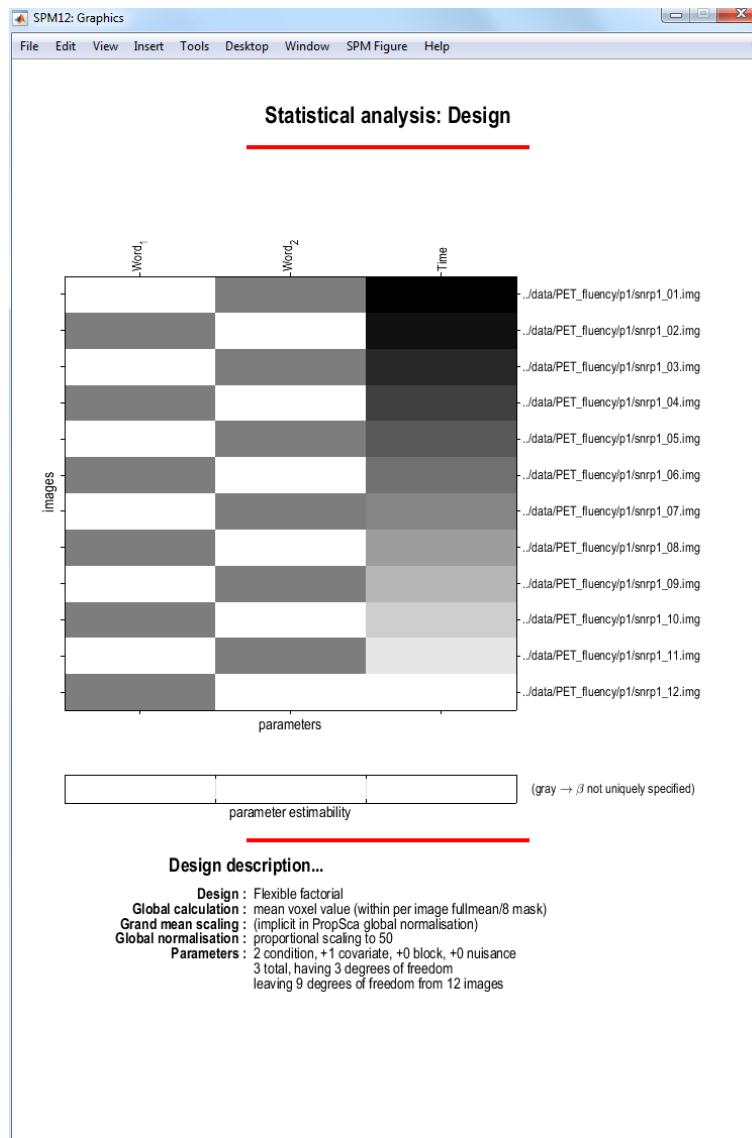


Figure 34.1: Design matrix for single-subject data. The first two columns model responses to word shadowing and word generation. The third column models time-varying responses.

### 34.3.1 Subject and Condition design

First we set up a design that allows us to test for the main effects of ‘Subject’ and ‘Condition’. The design can be set-up as follows.

- Start up MATLAB and type `spm pet` at the prompt
- Press the ‘Basic Models’ button.
- In ‘Factorial design specification’, under ‘Design’, choose the ‘Flexible Factorial’ design.
- Highlight ‘Factors’ and create a new Factor.
- Enter ‘Subject’ for the name and select ‘Equal’ under ‘Variance’.
- Then create another factor and call it ‘Word’
- Then, under ‘Specify Subject or all Scans and Factors’, highlight ‘Subjects’ and create a 5 new subjects.
- For the first subject, highlight ‘Scans’, select ‘Specify Files’ and use the SPM file selector to choose the 12 images for the first subject. This can be most easily achieved by specifying `.*snrp1.*` as a filter in the file selector and then using a right click to ‘select all’.
- Under ‘Conditions’ enter the vector `[1 2 1 2 1 2 1 2 1 2 1 2]`.
- Repeat the specification of scans and conditions for each of the four other subjects, remembering that the order of conditions has been balanced across the group (see Table 34.1).
- Under ‘Main effects and interactions’, create two main effects, the first with factor number 1 (ie. Subject) and the second with factor number 2 (ie. Word).
- Under Masking, select ‘Relative’ for ‘Threshold Masking’ and accept the default value of 0.8. Voxels with mean value less than 0.8 of the mean are deemed extra-cranial and will be excluded from the analysis.
- Under ‘Global calculation’ choose ‘Mean’
- Under, Global normalisation, and Normalisation, select ‘ANCOVA’.
- Highlight Directory, Specify files and select the subdirectory ‘subject-condition’, to place the design matrix in.
- Save the job file as eg. `DIR/sc_design.mat`.
- Press the Run button.

SPM will then show you the design matrix shown in Figure 34.2. This design is encoded in the `SPM.mat` file that is written to the output directory.

### 34.3.2 Subject and Time design

We now set up a design that allows us to test for the effects of Time (ie. scan number) and Subject. If you have already specified the Subject and Conditions design, then you can set up the Subject and Time design by editing the `sc_design.mat` file (and just changing the name of the second factor, Conditions vector and output directory - see below). Otherwise, the design can be set-up as follows.

- Start up MATLAB and type `spm pet` at the prompt
- Press the ‘Basic Models’ button.
- In ‘Factorial design specification’, under ‘Design’, choose the ‘Flexible Factorial’ design.
- Highlight ‘Factors’ and create a new Factor.

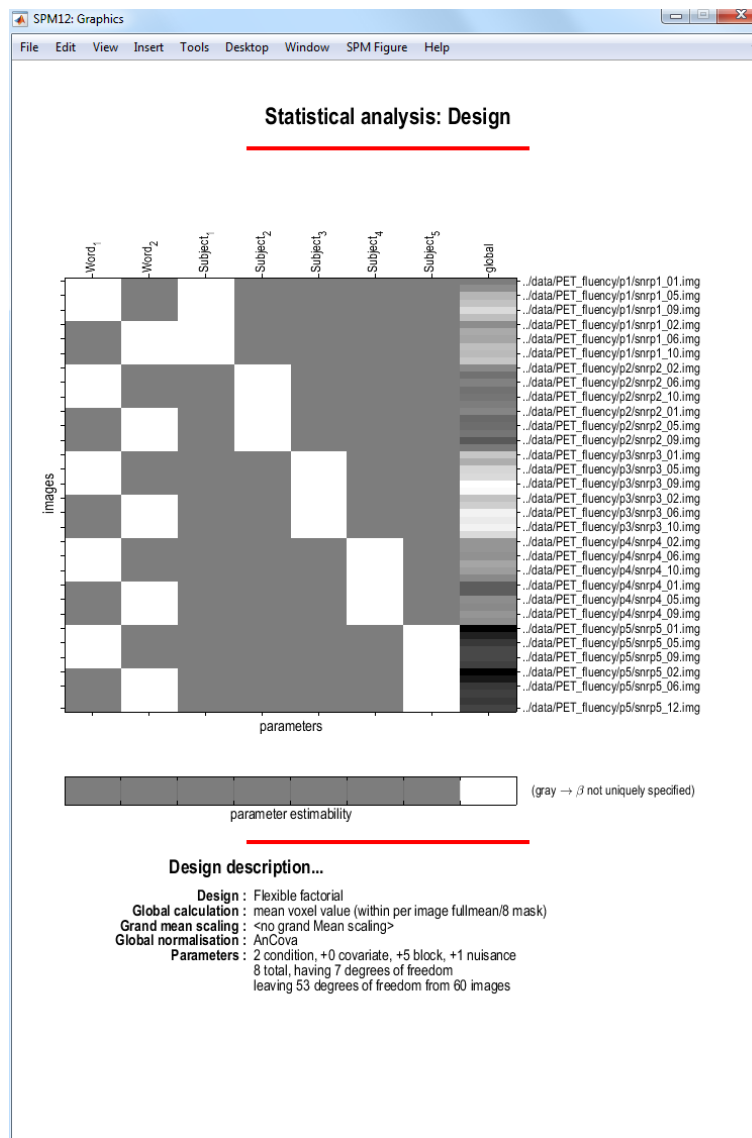


Figure 34.2: *Subjects and Conditions design for multiple-subject data. The first five columns model effect and the next two columns model condition effects. The last column models global effects (ANCOVA).*

- Enter 'Subject' for the name and select 'Equal' under 'Variance'.
- Then create another factor and call it 'Time'. This factor extends over time for each subject.
- Then, under 'Specify Subject or all Scans and Factors', highlight 'Subjects' and create a 5 new subjects.
- For the first subject, highlight 'Scans', select 'Specify Files' and use the SPM file selector to choose the 12 images for the first subject. This can be most easily achieved by specifying `.*snrp1.*` as a filter in the file selector and then using a right click to 'select all'.
- Under 'Conditions' enter the vector [1:12].
- Repeat the specification of scans and conditions for each of the four other subjects.
- Under 'Main effects and interactions', create two main effects, the first with factor number 1 (ie. Subject) and the second with factor number 2 (ie. Time).
- Under Masking, select 'Relative' for 'Threshold Masking' and accept the default value of 0.8. Voxels with mean value less than 0.8 of the mean are deemed extra-cranial and will be excluded from the analysis.
- Under 'Global calculation' choose 'Mean'
- Under, Global normalisation, and Normalisation, select 'ANCOVA'.
- Highlight Directory, Specify files and select the subdirectory 'subject-condition', to place the design matrix in.
- Save the job file as eg. `DIR/st_design.mat`.
- Press the Run button.

SPM will then show you the design matrix shown in Figure 34.3. This design is encoded in the `SPM.mat` file that is written to the output directory.

### 34.3.3 Subject by Condition design

This design models the interacts between 'Subject' and 'Condition'. It allows effects to be assessed separately for each subject. It will also allow us to implement a conjunction analysis over subjects.

If you have already specified the Subject and Conditions or Subject and Time designs then this design can be more easily specified by editing the `sc_design.mat` or `st_design.mat` files (and changing the name of the second factor, removing main effects, adding the interaction term and specifying a new output directory - see below). Otherwise, the design can be set-up as follows.

- Start up MATLAB and type `spm pet` at the prompt
- Press the "Basic Models" button.
- In 'Factorial design specification', under 'Design', choose the 'Flexible Factorial' design.
- Highlight 'Factors' and create a new Factor.
- Enter 'Subject' for the name and select 'Yes' under ANCOVA, as we will be implementing ANCOVA-by-subject. Select 'Equal' under 'Variance'.
- Then create another factor and call it 'Word'
- Then, under 'Specify Subject or all Scans and Factors', highlight 'Subjects' and create a 5 new subjects.
- For the first subject, highlight 'Scans', select 'Specify Files' and use the SPM file selector to choose the 12 images for the first subject. This can be most easily achieved by specifying `.*snrp1.*` as a filter in the file selector and then using a right click to 'select all'.

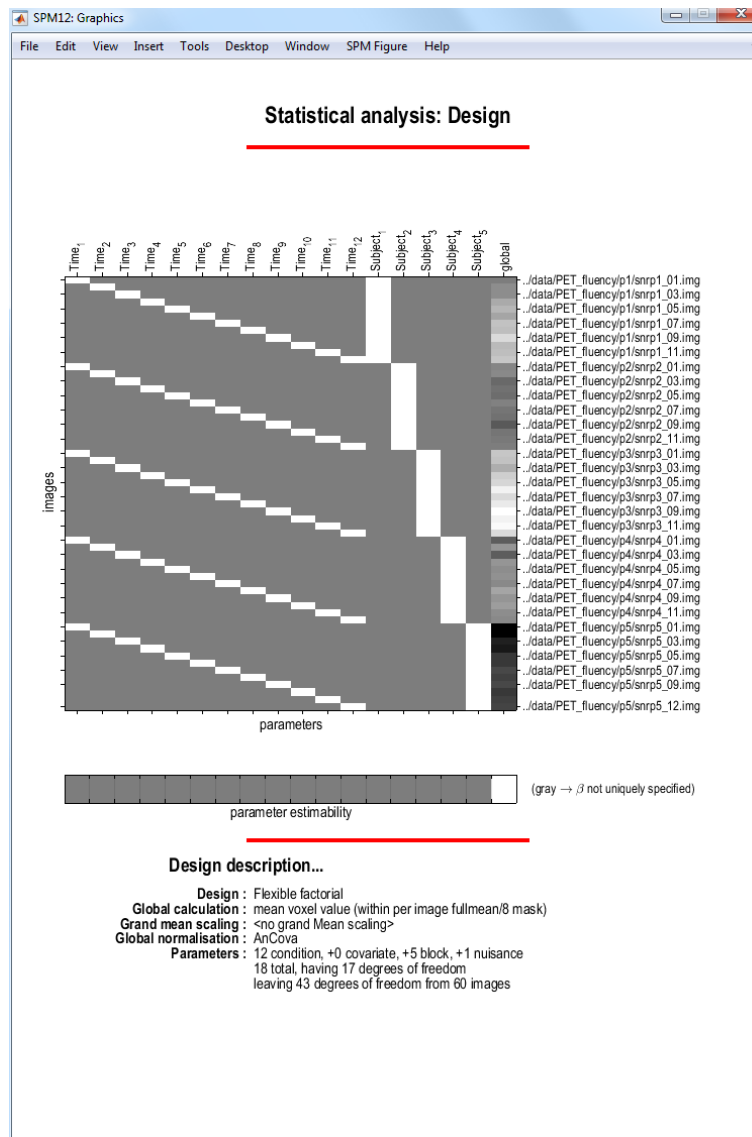


Figure 34.3: *Subjects and Time design for multiple-subject data. The first five columns model subjects effects and the next 12 model time effects. The last column models global effects (ANCOVA).*

- Under ‘Conditions’ enter the vector [1 2 1 2 1 2 1 2 1 2 1 2].
- Repeat the specification of scans and conditions for each of the four other subjects, remembering that the order of conditions has been balanced across the group (see Table 34.1).
- Under ‘Main effects and interactions’, create an interaction with factor numbers equal to [1 2]. This will create a block in the design matrix that models interactions between the factors ‘Subject’ and ‘Word’.
- Under Masking, select ‘Relative’ for ‘Threshold Masking’ and accept the default value of 0.8. Voxels with mean value less than 0.8 of the mean are deemed extra-cranial and will be excluded from the analysis.
- Under ‘Global calculation’ choose ‘Mean’
- Highlight Directory, Specify files and select the subdirectory `multiple`, to place the design matrix in.
- Save the job file as eg. `DIR/multi_design.mat`.
- Press the Run button.

SPM will then show you the design matrix shown in Figure 34.4. This design is encoded in the ‘SPM.mat’ file that is written to the output directory. Then press ‘Estimate’ and select the `SPM.mat` file just created. SPM will now estimate the parameters, that is, the size of the effect at each voxel. The rest of this chapter pursues the ‘Subject-by-Condition’ design.

### 34.3.4 Contrast manager

We can then examine relative activations, that is, regions which respond more strongly during word generation than word shadowing, for each subject. For subject 2:

- Press the ‘Results’ button.
- Select the `SPM.mat` file.
- In the contrast manager press ‘Define new contrast’ (select T)
- Specify e.g. `Subject 2: Gen > Shad` (name) and ‘0 0 -1 1’ (contrast).
- Press the ‘submit’ button. Press OK.
- Now press the ‘Done’ button.
- Mask with other contrast(s) [No]
- p value adjustment to control [FWE]
- Family-wise p-value [0.05]
- Extent threshold voxels [0]

This should produce the contrast in Figure 34.5. As shown, SPM will automatically pad ‘0 0 -1 1’ with zeros at the end. To examine group effects:

- Press the ‘Results’ button.
- Select the `SPM.mat` file.
- In the contrast manager press ‘Define new contrast’ (select T)
- Specify e.g. `All: Gen > Shad` (name) and ‘-1 1 -1 1 -1 1 -1 1 -1 1’ and select it (press ‘Done’) (contrast).
- Mask with other contrast(s) [No]



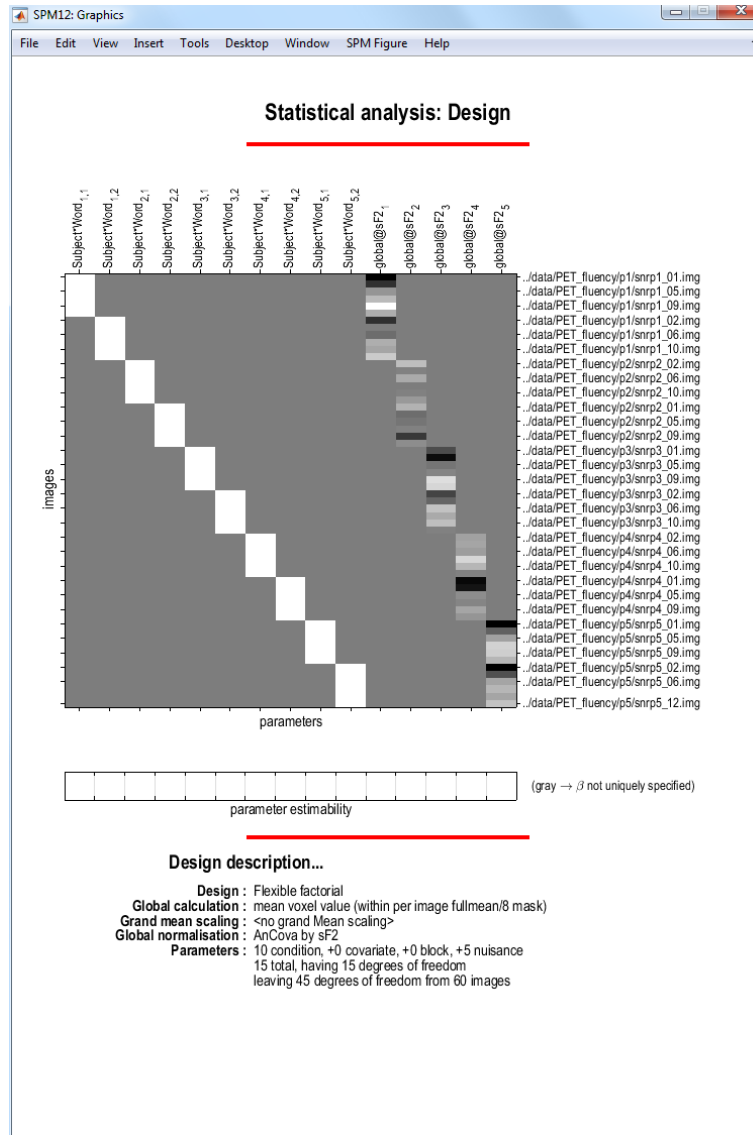


Figure 34.4: Subject by Condition design for multiple-subject data. The first ten columns model interactions between ‘Subject’ and ‘Word’. The last five columns model out global effects for each subject. Inclusion of these last five regressors implements a so-called ‘ANCOVA-by-subject’ normalisation.

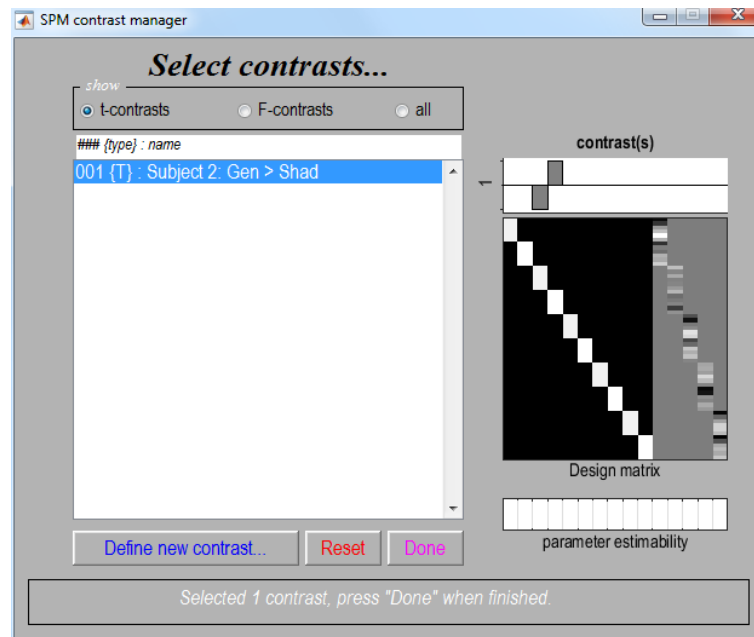


Figure 34.5: Activation contrast for subject 2. Note that the block of the design matrix encoding the experimental conditions is now coloured differently. This is because we have allowed the variance of responses over subjects to be different between word shadowing and generation conditions. This ‘nonsphericity’ affects parameter estimation in a way that is implemented in SPM by first ‘colouring’ the design matrix and then implementing ordinary least squares. This, in no way however, affects interpretation of effects.

- Title for comparison [activation]
- p value adjustment to control [FWE]
- Family-wise p-value [0.05]
- Extent threshold voxels [0]

Before looking at the results we describe the masking and thresholding options in more detail.

### 34.3.5 Masking and thresholds

Masking implies selecting voxels specified by other contrasts. If ‘yes’, SPM will prompt for (one or more) masking contrasts, the significance level of the mask (default  $p = 0.05$  uncorrected), and will ask whether an inclusive or exclusive mask should be used. Exclusive will remove all voxels which reach the default level of significance in the masking contrast, inclusive will remove all voxels which do not reach the default level of significance in the masking contrast. Masking does not affect p-values of the ‘target’ contrast.

Selecting a height threshold for examine results uses either a threshold corrected for multiple comparisons (‘yes’), or uncorrected (‘no’). The latter will produce many false positives (FPs) in the SPM. On average, the number of false positives will be equal to the number of voxels in the volume times the p-value (eg.  $50,000 \times 0.001 = 50$ ). If you correct for multiple comparisons, however, then there will typically be only one FP *anywhere* in 20 SPMs. Correcting for multiple comparisons is the recommended option.

Specifying an extent threshold  $x$  tells SPM not to plot clusters containing fewer than  $x$  voxels. The default,  $x = 0$  allows single voxel activations to be displayed.



Figure 34.6: Activation contrast for all subjects.

### 34.3.6 MIPs and results tables

The above contrast specifications should configure the contrast manager to appear as in Figure 34.6 and will configure SPM's graphics window to look like Figure 34.7.

SPM will also produce a number of files: images containing weighted parameter estimates are saved as `con_0002.hdr/img`, `con_0003.hdr/img`, etc. in the output directory. Images of T-statistics are saved as `spmT_0002.hdr/img`, `spmT_0003.hdr/img`, etc., also in the output directory. A number of further options are available from SPM Interactive window shown in Figure 34.8.

In the SPM Interactive window (lower left panel) a button box appears with various options for displaying statistical results (p-values panel) and creating plots/overlays (visualisation panel). Clicking 'Design' (upper left) will activate a pulldown menu as in the 'Explore design' option. To get a summary of local maxima, press 'volume'. This will produce the table shown in Figure 34.9. As in the previous example, this will list all clusters above the chosen level of significance as well as separate (>8mm apart) maxima within a cluster, with details of significance thresholds and search volume underneath. The columns show, from right to left:

- **x, y, z (mm):** coordinates in Talairach space for each maximum.
- **peak-level:** the chance (p) of finding (under the null hypothesis) a peak with this or a greater height (T- or Z-statistic), corrected / uncorrected for search volume.
- **cluster-level:** the chance (p) of finding a cluster with this or a greater size (ke), corrected / uncorrected for search volume.
- **set-level:** the chance (p) of finding this or a greater number of clusters (c) in the search volume.

Its also worth noting that

- The table is surfable: clicking a row of cluster coordinates will move the pointer in the MIP to that cluster, clicking other numbers will display the exact value in the Matlab window (e.g.  $0.000 = 6.1971e-07$ ).
- To inspect a specific cluster, either move the cursor in the MIP (by L-clicking & dragging the cursor, or R-clicking the MIP background which will activate a pulldown menu).

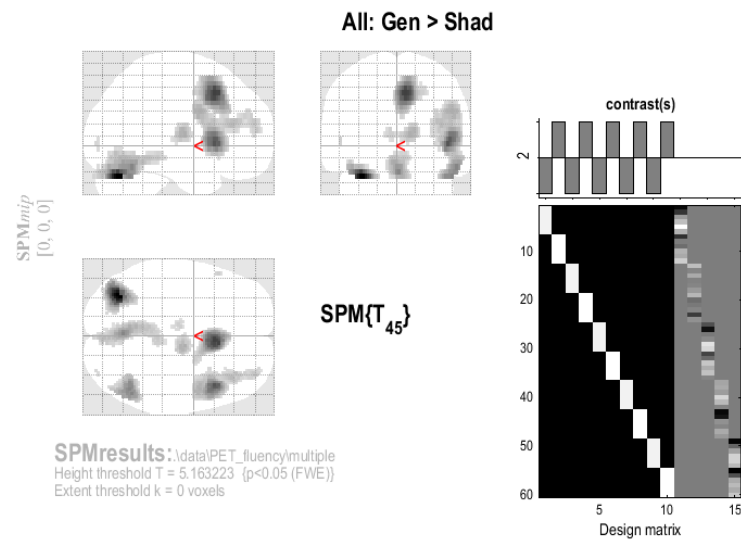


Figure 34.7: *SPMs graphics window displays (Left) a maximum intensity projection (MIP) on a glass brain in three orthogonal planes. The MIP is surfable: right-clicking in the MIP will activate a pulldown menu, left-clicking on the red cursor will allow it to be dragged to a new position, (Right) the design matrix (showing the selected contrast). The design matrix is also surfable: right-clicking will show parameter names, left-clicking will show design matrix values for each scan.*

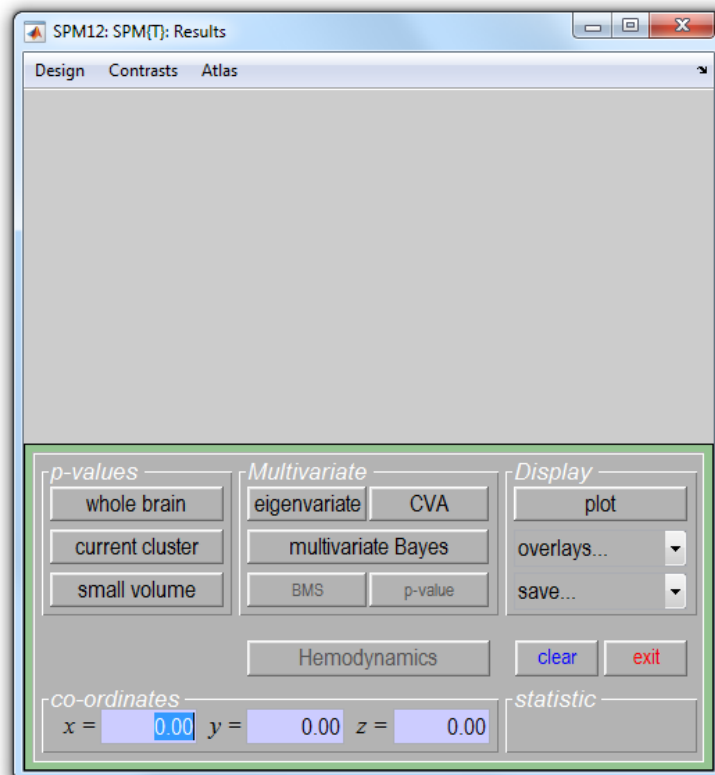


Figure 34.8: *SPM's interactive window.*

**Statistics: p-values adjusted for search volume**

set-level		cluster-level				peak-level					mm mm mm		
p	c	P <sub>FWE-corr</sub>	q <sub>FDR-corr</sub>	k <sub>E</sub>	P <sub>uncorr</sub>	P <sub>FWE-corr</sub>	q <sub>FDR-corr</sub>	T	(Z <sub>max</sub> )	P <sub>uncorr</sub>			
0.000	10	0.000	0.000	234	0.000	0.000	0.000	13.18	Inf	0.000	-34	-70	-28
		0.000	0.000	651	0.000	0.000	0.000	10.77	7.54	0.000	6	16	44
						0.000	0.001	8.09	6.33	0.000	2	24	36
						0.007	0.194	5.85	5.02	0.000	20	0	44
		0.000	0.000	888	0.000	0.000	0.000	9.97	7.21	0.000	44	16	0
						0.000	0.022	6.95	5.70	0.000	48	4	28
						0.001	0.050	6.49	5.42	0.000	38	32	16
		0.000	0.000	478	0.000	0.000	0.001	8.34	6.45	0.000	0	-66	-24
						0.000	0.014	7.12	5.80	0.000	-4	-60	-16
						0.001	0.044	6.55	5.46	0.000	-6	-48	-16
		0.000	0.000	271	0.000	0.000	0.001	8.32	6.44	0.000	52	-58	-20
						0.000	0.001	8.30	6.43	0.000	48	-60	-28
						0.000	0.010	7.28	5.89	0.000	54	-66	-16
		0.000	0.000	110	0.000	0.000	0.024	6.90	5.67	0.000	10	-10	8
						0.009	0.227	5.76	4.96	0.000	2	-12	8
		0.009	0.229	5	0.183	0.008	0.204	5.82	5.00	0.000	-52	20	4
		0.001	0.036	16	0.025	0.009	0.215	5.79	4.98	0.000	-8	-16	8
		0.012	0.258	4	0.232	0.022	0.476	5.45	4.75	0.000	32	-90	0
		0.020	0.398	2	0.398	0.030	0.595	5.35	4.68	0.000	-6	-20	4

table shows 3 local maxima more than 8.0mm apart

Height threshold: T = 5.16, p = 0.000 (0.050)	Degrees of freedom = [1.0, 45.0]
Extent threshold: k = 0 voxels	FWHM = 10.3 11.2 16.4 mm mm mm; 52 5.6 4.1 (voxels)
Expected voxels per cluster, <k> = 3.008	Volume: 880432 = 55027 voxels = 407.1 resels
Expected number of clusters, <c> = 0.05	Voxel size: 2.0 2.0 4.0 mm mm mm; (resel = 118.63 voxels)
FWEp: 5.163, FDRp: 6.487, FWEc: 2, FDRc: 16	

Figure 34.9: SPM results table. This appears below the MIP, shown in Figure 34.7, in the graphics window.

- Alternatively, click the cluster coordinates in the volume table, or type the coordinates in the lower left windows of the SPM Interactive window.

Selecting 'cluster' will show coordinates and voxel-level statistics for local maxima (>4mm apart) in the selected cluster. See Figure 34.10. The table is also surfable. Both in the 'volume' and 'cluster' options, p-values are corrected for the entire search volume.

### 34.3.7 Small volume correction

If one has an a priori anatomical hypothesis, eg. in the present example Broca's area will likely be activated during word generation, one may use the small volume correction option. Press the "small volume" button in SPM Interactive (bottom left) window and select a suitable region, e.g., a 30mm sphere with its centre at 44 16 0. The region can also be defined using mask images derived from previous imaging data. The corrected p-values will change, as shown in Figure 34.11.

### 34.3.8 Extracting data from regions

To extract a time course for data in this region of interest (this uses the SPM function `spm_regions.m`):

- Select "eigenvariate" from the "Multivariate" section in the Interactive window
- Specify 'Broca' for name of region and 0 for the VOI radius.
- Select ('don't adjust')

SPM displays a graph of the first eigenvariate of the data in or centered around the chosen voxel, as shown in Figure 34.12. It also lists the eigenvariate values Y in the Matlab window. Adjustment is with respect to the null space of a selected contrast. This means that any effects not spanned by the chosen contrast are removed from the data, before extraction. Adjustment can be omitted by selecting 'don't adjust', as above.

SPM extracts the eigenvariate values in a region, rather than the mean values, as the former is more robust to heterogeneity of response within a cluster. The mean value can be thought of as a special case of the eigenvariate if the corresponding eigenvector weights all voxels in a cluster equally. Effectively, the eigenvariate provides a weighted mean where atypical voxels are downweighted.

**Statistics: p-values adjusted for search volume**

cluster-level				peak-level					mm mm mm		
$p_{FWECorr}$	$q_{FDRcorr}$	$k_E$	$p_{Uncorr}$	$p_{FWECorr}$	$q_{FDRcorr}$	$T$	$(Z_{\text{max}})$	$p_{Uncorr}$			
0.000	0.000	888	0.000	0.000	0.000	9.97	7.21	0.000	44	16	0
				0.000	0.022	6.95	5.70	0.000	48	4	28
				0.000	0.028	6.82	5.62	0.000	50	0	28
				0.001	0.050	6.49	5.42	0.000	38	32	16
				0.002	0.068	6.33	5.32	0.000	34	36	20
				0.003	0.100	6.18	5.23	0.000	36	28	20
				0.003	0.100	6.17	5.23	0.000	38	12	16
				0.003	0.103	6.15	5.21	0.000	40	14	20
				0.003	0.113	6.10	5.18	0.000	38	38	20
				0.004	0.118	6.07	5.16	0.000	34	52	20
				0.004	0.128	6.01	5.12	0.000	36	54	12
				0.008	0.214	5.80	4.98	0.000	34	44	24
				0.010	0.247	5.72	4.94	0.000	36	20	16
				0.019	0.417	5.51	4.79	0.000	30	12	12
				0.019	0.418	5.50	4.79	0.000	32	28	16

table shows 16 local maxima more than 4.0mm apart

Height threshold: $T = 5.16$ , $p = 0.000$ (0.050)	Degrees of freedom = [1.0, 45.0]
Extent threshold: $k = 0$ voxels	FWHM = 10.3 11.2 16.4 mm mm mm; 5.2 5.6 4.1 (voxels)
Expected voxels per cluster, $\langle k \rangle = 3.008$	Volume: 880432 = 55027 voxels = 407.1 resels
Expected number of clusters, $\langle c \rangle = 0.05$	Voxel size: 2.0 2.0 4.0 mm mm mm; (resel = 118.63 voxels)
FWEp: 5.163, FDRp: 6.487, FWEC: 2, FDRc: 16	

Figure 34.10: SPM results table for a single cluster with p-values corrected for the whole brain.

**Statistics: search volume: 30.0mm sphere at [44,16,0]**

cluster-level				peak-level					mm mm mm		
$p_{FWECorr}$	$q_{FDRcorr}$	$k_E$	$p_{Uncorr}$	$p_{FWECorr}$	$q_{FDRcorr}$	$T$	$(Z_{\text{max}})$	$p_{Uncorr}$			
0.000	0.000	736	0.000	0.000	0.000	9.97	7.21	0.000	44	16	0
				0.000	0.042	6.72	5.56	0.000	50	2	24
				0.000	0.051	6.56	5.46	0.000	44	6	28
				0.000	0.051	6.49	5.42	0.000	38	32	16
				0.000	0.068	6.33	5.32	0.000	34	36	20
				0.000	0.076	6.18	5.23	0.000	36	28	20
				0.000	0.076	6.17	5.23	0.000	38	12	16
				0.000	0.076	6.15	5.21	0.000	40	14	20
				0.002	0.241	5.72	4.94	0.000	36	20	16
				0.003	0.379	5.51	4.79	0.000	30	12	12
				0.003	0.379	5.50	4.79	0.000	32	28	16

table shows 16 local maxima more than 4.0mm apart

Height threshold: $T = 5.16$ , $p = 0.000$ (0.008)	Degrees of freedom = [1.0, 45.0]
Extent threshold: $k = 0$ voxels	FWHM = 10.3 11.2 16.4 mm mm mm; 5.2 5.6 4.1 (voxels)
Expected voxels per cluster, $\langle k \rangle = 3.008$	Volume: 66128 = 4133 voxels = 59.6 resels
Expected number of clusters, $\langle c \rangle = 0.01$	Voxel size: 2.0 2.0 4.0 mm mm mm; (resel = 118.63 voxels)
FWEp: 4.457, FDRp: 6.719, FWEC: 736, FDRc: 736	

Figure 34.11: SPM results table for a single cluster with p-values corrected using the Small Volume Correction (SVC) option. This used a 30mm sphere centred at 44 16 0. Note the reduced number of voxels in the search volume (bottom right text in Figure) and more significant p-values as compared to Figure 34.10.

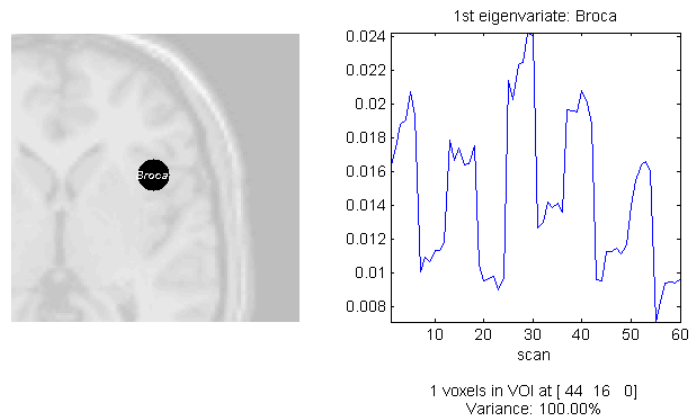


Figure 34.12: *Data extracted from a Volume of Interest (VOI).*

A file called `VOI_regionname.mat` is created in the working directory containing `Y` and `VOI` details (in the data structure `xY`).

### 34.3.9 Inclusive Masking

We have so far looked at the *average* effect over the five subjects in our group using the ‘All: Gen > Shad’ contrast. To assess condition effects that are *common* to all subjects, one can either mask (inclusively) the ‘All: Gen > Shad’ contrast with the individual contrasts, or perform a conjunction analysis. Firstly we’ll use the inclusive masking approach.

- Press the ‘Results’ button.
- Select the `SPM.mat` file.
- Select the `All: Gen > Shad` contrast and press ‘Done’.
- Mask with other contrast(s) [Yes]
- Then hold down the [control] button whilst selecting all the individual contrasts. The contrast manager should then appear as in Figure 34.13.
- Uncorrected mask p-value [0.05]
- Nature of mask [inclusive]
- p value adjustment to control [FWE]
- Family-wise p-value [0.05]
- Extent threshold voxels [0]

This should produce the MIP and results table shown in Figure 34.14.

### 34.3.10 Conjunctions

To perform a conjunction approach across subjects:

- Press the ‘Results’ button.
- Select the `SPM.mat` file.

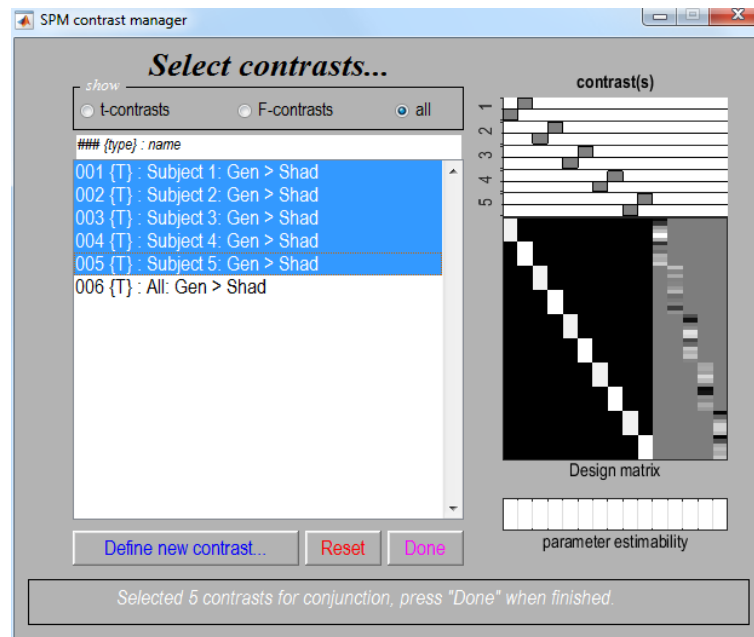


Figure 34.13: SPM can produce maps based on multiple contrasts by holding down [control] whilst selecting contrasts. This can be used during masking and when making a conjunction inference.

- Then hold down the [control] button whilst selecting all the individual contrasts. The contrast manager should then appear as in Figure 34.13 (except that, in the white text at the bottom, it should indicate that a conjunction will be performed).
- Null hyp. to assess [Global]
- Mask with other contrasts [No]
- Title for comparison [accept the default]
- p value adjustment to control [FWE]
- Family-wise p-value [0.05]
- Extent threshold voxels [0]

SPM checks whether the contrasts are orthogonal and, if not, makes them so. Contrasts are orthogonalized with respect to the first contrast specified.

SPM should produce the MIP and table of results shown in Figure 34.15. The p-value (corrected or uncorrected) refers to the threshold of the conjunction. SPM will compute corresponding thresholds for individual contrasts. For uncorrected thresholds, the individual threshold will be  $p^1/n$ , where  $p$  is the individual threshold and  $n$  is the number of contrasts in the conjunction.

Height, and not extent, is used to specify thresholding because the distributional approximations for the spatial extent of a conjunction SPM are not known (at present), so that inference based on spatial extent is precluded.

Although the MIP's of the masked group contrast and the conjunction are similar, for the conjunction an intersection SPM or 'minimum T-field' is computed. This intersection is the same as thresholding a map of the minimum T-values. If the smallest T-value is above the specified threshold then all the T-values associated with the component SPMs are above threshold.

Conjunction SPMs are very useful for testing multiple hypotheses (each component hypothesis being specified by a contrast). In this example, we have chosen to use the Global Null Hypothesis. The set of hypotheses tested jointly is that the first subject did not activate, the second subject did not activate and so on.



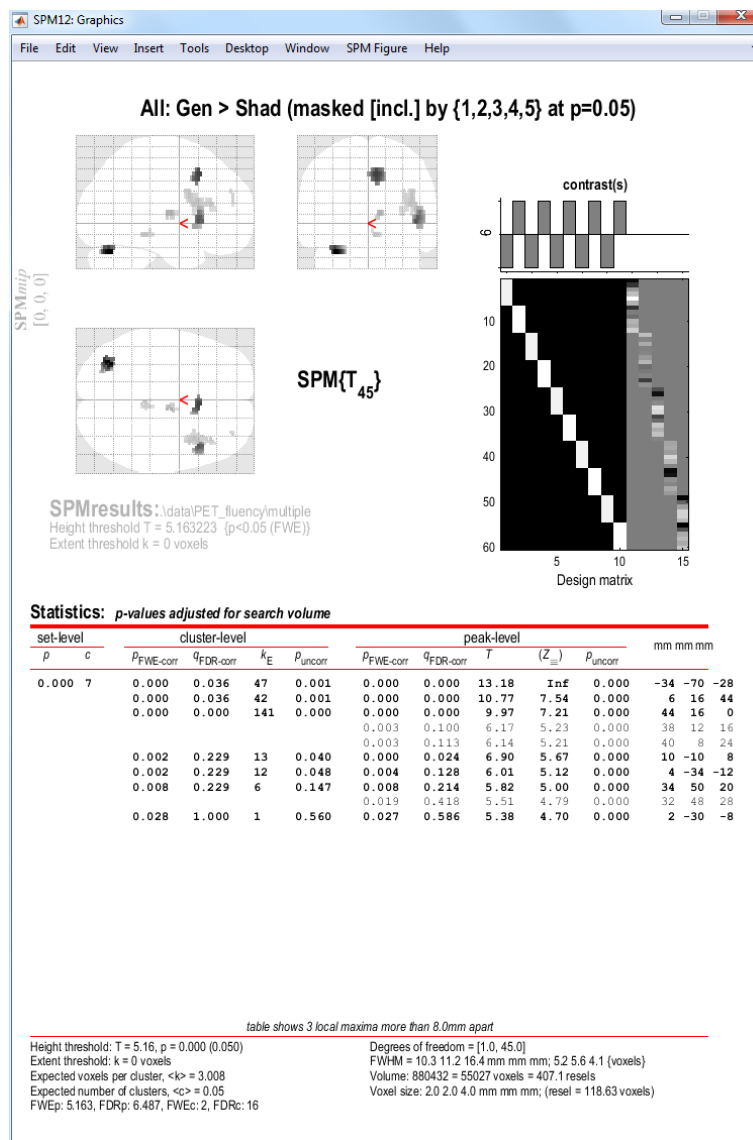


Figure 34.14: The SPM shows results from the inclusive masking approach. It shows all voxels which are (a) significant at  $p < 0.05$  corrected across all subjects and (b) significant at  $p < 0.05$  uncorrected for each subject individually.

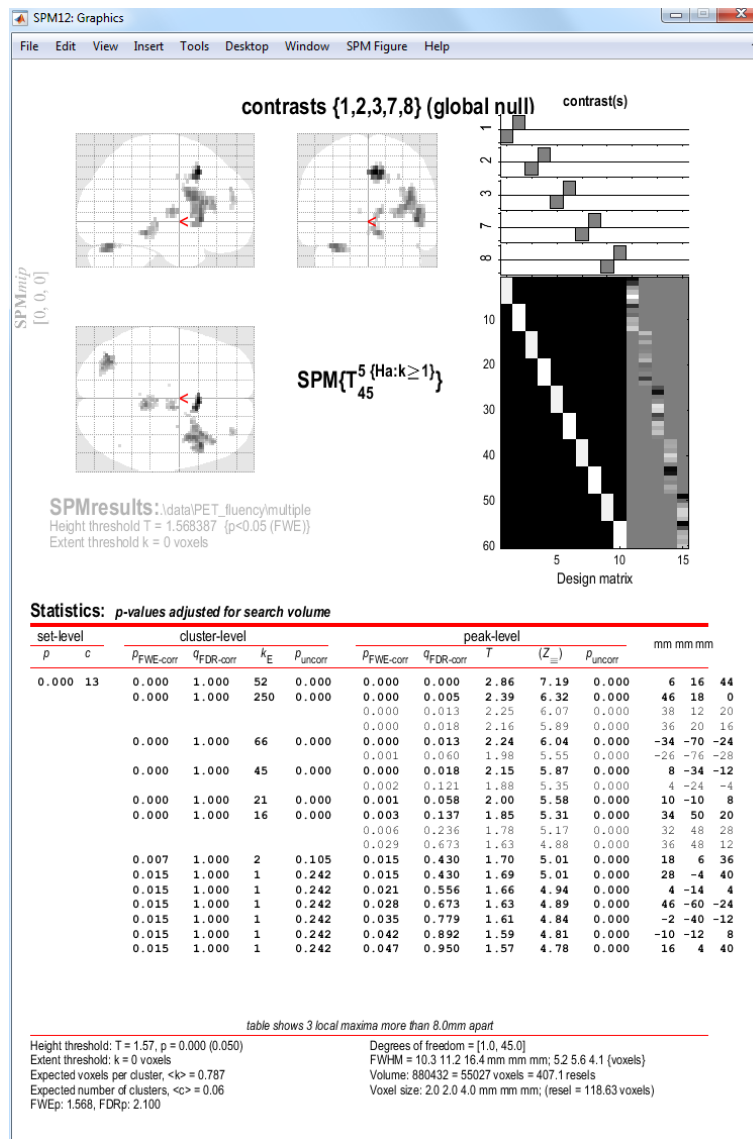


Figure 34.15: Conjunction SPM.

SPM also provides an option to use the Conjunction Null hypothesis. This can be thought of as enabling an inference that subject 1 activated AND subject 2 activated AND subject 3... etc. For more discussion on this issue, see [40] and [79].

Gaussian field theory results are available for SPMs of minimum T- (or F-) statistics and therefore corrected p-values can be computed. Note that the minimum T-values do not have the usual Student's T-distribution and small minimum T-values can be very significant.

## Chapter 35

# Dynamic Causal Modeling for fMRI

### 35.1 Theoretical background

Dynamic Causal Modelling (DCM) is a method for making inferences about neural processes that underlie measured time series, e.g. fMRI data. The general idea is to estimate the parameters of a reasonably realistic neuronal system model such that the predicted blood oxygen level dependent (BOLD) signal, which results from converting the modeled neural dynamics into hemodynamic responses, corresponds as closely as possible to the observed BOLD time series. This section gives a short introduction to the theoretical background of DCM for fMRI; details can be found in [34]. Note that DCMs can be formulated, in principle, for any measurement technique. Depending on the spatio-temporal properties of a given measurement technique, one needs to define an adequate state equation and an observation model. See Fig 35.1 for a summary of the differences between DCM implementations for fMRI and Event-Related Potentials (ERPs). For a gentle introduction to DCM, written for non-technical imaging researchers, see [60].

As in state-space models, two distinct levels constitute a DCM (see Figure 35.2). The hidden level, which cannot be directly observed using fMRI, represents a simple model of neural dynamics in a system of  $k$  coupled brain regions. Each system element  $i$  is represented by a single state variable  $z_i$ , and the dynamics of the system is described by the change of the neural state vector over time.

The neural state variables do not correspond directly to any common neurophysiological measurement (such as spiking rates or local field potentials) but represent a summary index of neural population dynamics in the respective regions. Importantly, DCM models how the neural dynamics are driven by external perturbations that result from experimentally controlled manipulations. These perturbations are described by means of external inputs  $u$  that enter the model in two different ways: they can elicit responses through direct influences on specific regions (“driving” inputs, e.g. evoked responses in early sensory areas) or they can change the strength of coupling among regions (“modulatory” inputs, e.g. during learning or attention).

Overall, DCM models the temporal evolution of the neural state vector, i.e.  $z$ , as a function of the current state, the inputs  $u$  and some parameters that define the functional architecture and interactions among brain regions at a neuronal level ( $n$  denotes “neural”):

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dots \\ \dot{z}_k \end{bmatrix} = \dot{z} = \frac{dz}{dt} = F(z, u, \theta^n) \quad (35.1)$$

In this neural state equation, the state  $z$  and the inputs  $u$  are time-dependent whereas the parameters are time-invariant. In DCM,  $F$  has the bilinear form

$$\dot{z} = Az + \sum_{j=1}^m u_j B_j z + Cu \quad (35.2)$$

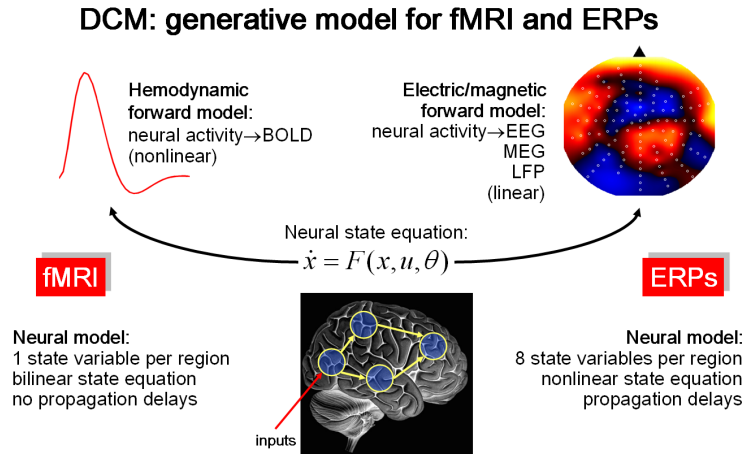


Figure 35.1: A schematic overview of the differences between the DCM implementations for fMRI and ERPs (as measured by EEG or MEG). Whereas the state equation of DCM for fMRI is bilinear and uses only a single state variable per region, that for ERPs is more complex and requires 8 state variables per region. Moreover, DCM for ERPs models the delays of activity propagation between areas. At the level of the observation model, DCM for fMRI is more complex than DCM for ERPs. While the former uses a non-linear model of the hemodynamic response that contains a cascade of differential equations with five state variables per region, the latter uses a simple linear model for predicting observed scalp data.

The parameters of this bilinear neural state equation,  $\theta^n = \{A, B_1, \dots, B_m, C\}$ , can be expressed as partial derivatives of  $F$ :

$$\begin{aligned}
 A &= \frac{\partial F}{\partial z} = \frac{\partial \dot{z}}{\partial z} \\
 B_j &= \frac{\partial^2 F}{\partial z \partial u_j} = \frac{\partial}{\partial u_j} \frac{\partial \dot{z}}{\partial z} \\
 C &= \frac{\partial F}{\partial u}
 \end{aligned} \tag{35.3}$$

These parameter matrices describe the nature of the three causal components which underlie the modeled neural dynamics: (i) context-independent effective connectivity among brain regions, mediated by anatomical connections ( $k \times k$  matrix  $A$ ), (ii) context-dependent changes in effective connectivity induced by the  $j$ th input  $u_j$  ( $k \times k$  matrices  $B_1, \dots, B_m$ ), and (iii) direct inputs into the system that drive regional activity ( $k \times m$  matrix  $C$ ). As will be demonstrated below, the posterior distributions of these parameters can inform us about the impact that different mechanisms have on determining the dynamics of the model. Notably, the distinction between “driving” and “modulatory” is neurobiologically relevant: driving inputs exert their effects through direct synaptic responses in the target area, whereas modulatory inputs change synaptic responses in the target area in response to inputs from another area. This distinction represents an analogy, at the level of large neural populations, to the concept of driving and modulatory afferents in studies of single neurons.

DCM combines this model of neural dynamics with a biophysically plausible and experimentally validated hemodynamic model that describes the transformation of neuronal activity into a BOLD response. This so-called “Balloon model” was initially formulated by Buxton and colleagues and later extended by [38]. Briefly summarized, it consists of a set of differential equations that describe the relations between four hemodynamic state variables, using five parameters ( $\theta^h$ ). More specifically, changes in neural activity elicit a vasodilatory signal that leads to increases in blood flow and subsequently to changes in blood volume  $v$  and deoxyhemoglobin content  $q$ . The predicted BOLD signal  $y$  is a non-linear function of blood volume and deoxyhemoglobine content.

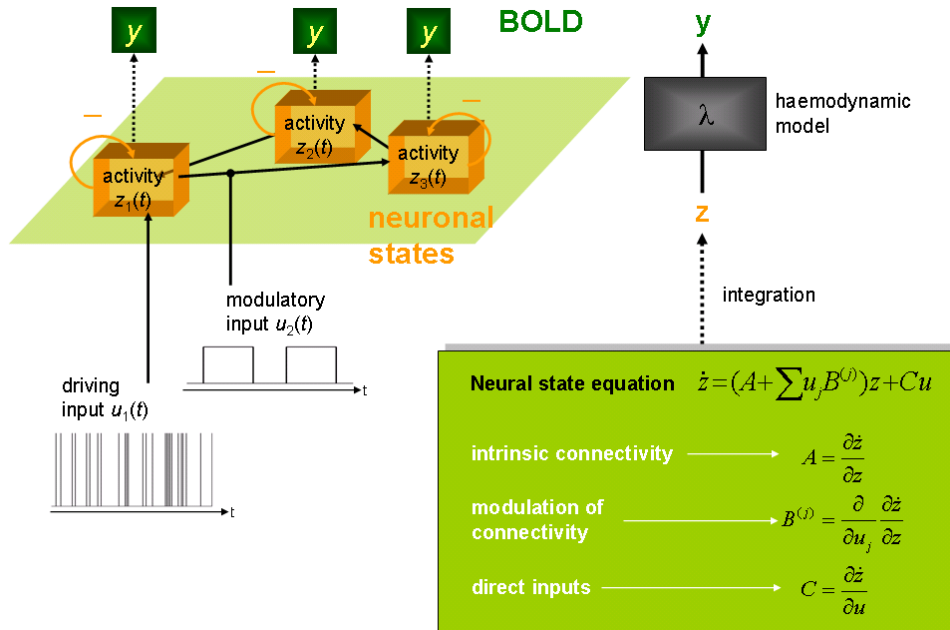


Figure 35.2: Schematic summary of the conceptual basis of DCM. The dynamics in a system of interacting neuronal populations (orange boxes), which are not directly observable by fMRI, is modeled using a bilinear state equation (grey box). Integrating the state equation gives predicted neural dynamics ( $z$ ) that enter a model of the hemodynamic response ( $\lambda$ ) to give predicted BOLD responses ( $y$ ) (green boxes). The parameters at both neural and hemodynamic levels are adjusted such that the differences between predicted and measured BOLD series are minimized. Critically, the neural dynamics are determined by experimental manipulations. These enter the model in the form of “external” or “driving” inputs. Driving inputs ( $u_1$ ; e.g. sensory stimuli) elicit local responses directly that are propagated through the system according to the endogenous (or latent) coupling. The strengths of these connections can be changed by modulatory inputs ( $u_2$ ; e.g. changes in cognitive set, attention, or learning).

Details of the hemodynamic model can be found in other publications [38]. By combining the neural and hemodynamic states into a joint state vector  $x$  and the neural and hemodynamic parameters into a joint parameter vector  $\theta = [\theta^n, \theta^h]^T$ , we obtain the full forward model that is defined by the neural and hemodynamic state equations

$$\begin{aligned} \dot{x} &= F(x, u, \theta) \\ y &= \lambda(x) \end{aligned} \quad (35.4)$$

For any given set of parameters  $\theta$  and inputs  $u$ , the joint state equation can be integrated and passed through the output nonlinearity  $\lambda$  to give a predicted BOLD response  $h(u, \theta)$ . This can be extended to an observation model that includes observation error  $e$  and confounding effects  $X$  (e.g. scanner-related low-frequency drifts):

$$y = h(u, \theta) + X\beta + e \quad (35.5)$$

This formulation is the basis for estimating the neural and hemodynamic parameters from the measured BOLD data, using a fully Bayesian approach with empirical priors for the hemodynamic parameters and conservative shrinkage priors for the neural coupling parameters.

Details of the parameter estimation scheme, which rests on a Fisher scoring gradient ascent scheme with Levenburg-Marquardt regularisation, embedded in an expectation maximization (EM) algorithm, can be found in the original DCM publication [34]. In brief, under Gaussian assumptions about the posterior distributions, this scheme returns the posterior expectations  $\eta_{\theta|y}$

and posterior covariance  $C_{\theta|y}$  for the parameters as well as hyperparameters for the covariance of the observation noise,  $C_e$ .

After fitting the model to measured BOLD data, the posterior distributions of the parameters can be used to test hypotheses about the size and nature of effects at the neural level. Although inferences could be made about any of the parameters in the model, hypothesis testing usually concerns context-dependent changes in coupling (i.e. specific parameters from the  $B$  matrices; see Fig. 35.5). As will be demonstrated below, at the single-subject level, these inferences concern the question of how certain one can be that a particular parameter or, more generally, a contrast of parameters,  $c^T \eta_{\theta|y}$ , exceeds a particular threshold  $\gamma$  (e.g. zero).

Under the assumptions of the Laplace approximation, this is easy to test ( $\Phi_N$  denotes the cumulative normal distribution):

$$p(c^T \eta_{\theta|y} > \gamma) = \Phi_N \left( \frac{c^T \eta_{\theta|y} - \gamma}{\sqrt{c^T C_{\theta|y} c}} \right) \quad (35.6)$$

For example, for the special case  $c^T \eta_{\theta|y} = \gamma$  the probability is  $p(c^T \eta_{\theta|y} > \gamma) = 0.5$ , i.e. it is equally likely that the parameter is smaller or larger than the chosen threshold  $\gamma$ . We conclude this section on the theoretical foundations of DCM by noting that the parameters can be understood as rate constants (units:  $1/s = Hz$ ) of neural population responses that have an exponential nature. This is easily understood if one considers that the solution to a linear ordinary differential equation of the form  $\dot{z} = Az$  is an exponential function (see Fig. 35.3).

**Integration of a first-order linear differential equation gives an exponential function:**

$$\frac{dz}{dt} = az \quad \longrightarrow \quad z(t) = z_0 \exp(at)$$

Coupling parameter  $a$  is inversely proportional to the half life  $\tau$  of  $z(t)$ :

$$\begin{aligned} z(\tau) &= 0.5z_0 \\ &= z_0 \exp(a\tau) \end{aligned}$$

$$\longrightarrow \quad a = \ln 2 / \tau$$

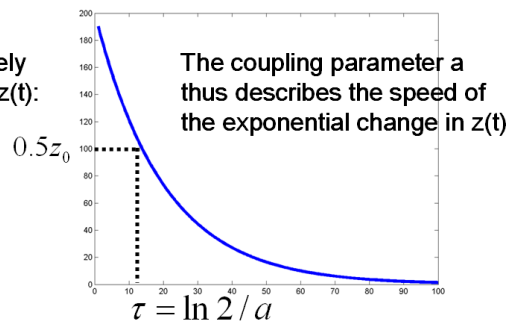


Figure 35.3: A short mathematical demonstration, using a simple linear first-order differential equation as an example, explaining why the coupling parameters in a DCM are inversely proportional to the half-life of the modelled neural responses and are therefore in units of  $1/s = \text{Hertz}$ .

## 35.2 Bayesian model selection

A generic problem encountered by any kind of modeling approach is the question of model selection: given some observed data, which of several alternative models is the optimal one? This problem is not trivial because the decision cannot be made solely by comparing the relative fit of the competing models. One also needs to take into account the relative complexity of the models as expressed, for example, by the number of free parameters in each model.

Model complexity is important to consider because there is a trade-off between model fit and generalizability (i.e. how well the model explains different data sets that were all generated from the same underlying process). As the number of free parameters is increased, model fit increases monotonically whereas beyond a certain point model generalizability decreases. The reason for

this is “overfitting”: an increasingly complex model will, at some point, start to fit noise that is specific to one data set and thus become less generalizable across multiple realizations of the same underlying generative process.

Therefore, the question “What is the optimal model?” can be reformulated more precisely as “What is the model that represents the best balance between fit and complexity?”. In a Bayesian context, the latter question can be addressed by comparing the evidence,  $p(y|m)$ , of different models. According to Bayes theorem

$$p(\theta|y, m) = \frac{p(y|\theta, m)p(\theta|m)}{p(y|m)} \quad (35.7)$$

the model evidence can be considered as a normalization constant for the product of the likelihood of the data and the prior probability of the parameters, therefore

$$p(y|m) = \int p(\theta|y, m)p(\theta|m)d\theta \quad (35.8)$$

Here, the number of free parameters (as well as the functional form) are considered by the integration. Unfortunately, this integral cannot usually be solved analytically, therefore an approximation to the model evidence is needed. One such approximation used by DCM, and many other models in SPM, is to make use of the Laplace approximation <sup>1</sup>.

As shown in [86], this yields the following expression for the natural logarithm ( $\ln$ ) of the model evidence ( $\eta_{\theta|y}$  denotes the posterior mean,  $C_{\theta|y}$  is the posterior covariance of the parameters,  $C_e$  is the error covariance,  $\theta_p$  is the prior mean of the parameters, and  $C_p$  is the prior covariance):

$$\begin{aligned} \ln p(y|m) &= \text{accuracy}(m) - \text{complexity}(m) \quad (35.9) \\ &= \left[ -\frac{1}{2}\ln|C_e| - \frac{1}{2}(y - h(u, \eta_{\theta|y}))^T C_e^{-1} (y - h(u, \eta_{\theta|y})) \right] \\ &\quad - \left[ \frac{1}{2}\ln|C_p| - \frac{1}{2}\ln|C_{\theta|y}| + \frac{1}{2}(\eta_{\theta|y} - \theta_p)^T C_p^{-1} (\eta_{\theta|y} - \theta_p) \right] \end{aligned}$$

This expression properly reflects the requirement, as discussed above, that the optimal model should represent the best compromise between model fit (accuracy) and model complexity. The complexity term depends on the prior density, for example, the prior covariance of the intrinsic connections.

Two models can then be compared using the Bayes factor:

$$BF_{ij} = \frac{p(y|m_i)}{p(y|m_j)} \quad (35.10)$$

Given uniform priors over models, the posterior probability for model  $i$  is greater 0.95 if  $BF_{ij}$  is greater than twenty.

This results in a robust procedure for deciding between competing hypotheses represented by different DCMs. These hypotheses can concern any part of the structure of the modeled system, e.g. the pattern of endogenous connections (A-matrix) or which inputs affect the system and where they enter (C-matrix). Note, however, that this comparison is only valid if the data  $y$  are identical in all models. This means that in DCM for fMRI, where the data vector results from a concatenation of the time series of all areas in the model, only models can be compared that contain the same areas. Therefore, model selection cannot be used to address whether or not to include a particular area in the model. In contrast, in DCM for ERPs, the data measured at the sensor level are independent of how many neuronal sources are assumed in a given model. Here, model selection could also be used to decide which sources should be included.

### 35.3 Practical example

The following example refers to the “attention to visual motion” data set available from the SPM web site<sup>2</sup>. This data set was obtained by Christian Buchel and is described in [13].

<sup>1</sup>This should perhaps more correctly be referred to as a fixed-form variational approximation, where the fixed form is chosen to be a Gaussian. The model evidence is approximated by the negative free energy,  $F$ .

<sup>2</sup>Attention to visual motion dataset: <http://www.fil.ion.ucl.ac.uk/spm/data/attention/>

The archive contains the smoothed, spatially normalised, realigned, slice-time corrected images in the directory `functional`. The directory `structural` contains a spatially normalised structural image. All processing took place using SPM99, but the image files have been converted into NIFTI format.

Making a DCM requires two ingredients: (i) a design matrix and (ii) the time series, stored in VOI files. The regressors of the design matrix define the inputs for the DCM. Note that this means that the design matrix that is optimal for a given DCM is often somewhat different than the one for the corresponding GLM. DCM does not require the design matrix to be part of an estimated model, however. It just needs to be defined.

### 35.3.1 Defining the GLM

The present experiment consisted of 4 conditions: (i) fixation (F), (ii) static (S, non-moving dots), (iii) no attention (N, moving dots but no attention required), (iv) attention (A). The GLM analyses by Christian showed that activity in area V5 was not only enhanced by moving stimuli, but also by attention to motion. In the following, we will try to model this effect in V5, and explain it as a context-dependent modulation or “enabling” of V5 afferents, using a DCM. First, we need to set up the GLM analysis and extract our time series from the results. In this example, we want to use the same design matrix for GLM and DCM, therefore we recombine the above regressors to get the following three conditions:

1. **photoc**: this comprises all conditions with visual input, i.e. S, N, and A.
2. **motion**: this includes all conditions with moving dots, i.e. N and A.
3. **attention**: this includes the attention-to-motion (A) condition only.

Now we need to define and estimate the GLM. This is not the main topic of this chapter so you should already be familiar with these procedures, see 8 and 9 for more information. Here are the relevant details for this data set that you need to set up the GLM:

- The onsets for the conditions can be found in the file `factors.mat`. They are named `stat` (static), `natt` (no attention) and `att` (attention) and are defined in scans (not seconds). They are blocks of 10 TRs each.
- The TR is 3.22 seconds.
- There are 360 scans.

Let’s specify a batch that will specify the model and estimate it.

1. The analysis directory you have downloaded should include
  - (a) A directory named `functional`, which includes the preprocessed fMRI volumes.
  - (b) A directory named `structural`, which includes a normalised T1 structural volume
  - (c) File `factors.mat`.
  - (d) You will also need to make a new directory called `GLM` that will contain the analysis.
2. In MATLAB type
 

```
>> cd GLM
>> spm fmri
```
3. From the main SPM window, click on the BATCH button.
4. From the SPM menu at the top of the Batch Editor, select “Stats > fMRI model specification”.
5. Click DIRECTORY and choose the GLM directory that you made above.
6. UNITS FOR DESIGN [SCANS]



7. INTERSCAN INTERVAL [3.22]
8. Click DATA & DESIGN, Choose NEW "SUBJECT/SESSION"
9. Click SCANS and choose all the functional scans `snffM00587_00xx.img`. There should be 360 `*.img` files.
10. Go back to the main MATLAB workspace and load the MAT-file containing the experimental conditions:

```
>> load ../factors.mat
```

You can look at the loaded variables by typing the variable names. (`stat` = stationary, `natt` = no attention, `att` = attention)

```
>> stat
>> natt
>> att
```

11. Return to the batch editor. Click CONDITIONS then double click NEW: CONDITION three times. Enter the following details for each:
  - Condition 1: NAME = `Photic`, ONSETS = [`att natt stat`] and DURATIONS = 10.
  - Condition 2: NAME = `Motion`, ONSETS = [`att natt`] and DURATIONS = 10.
  - Condition 3: NAME = `Attention`, ONSETS = `att` and DURATIONS = 10.
12. From the SPM menu at the top of the Batch Editor, select "Stats > model estimation".
13. For SELECT SPM.MAT, click on the DEPENDENCY button and choose the proposed item (the output from the previous module).
14. You should now be able to press the RUN green arrow at the top of the Batch Editor window. This will specify and estimate the GLM.

### 35.3.2 Extracting time series

Once you have specified and estimated the GLM, you should define t-contrasts that test for photic, motion, and attention, respectively. These serve to locate areas that show effects due to visual stimulation (e.g. in V1), motion (e.g. V5) and attention (e.g. V5 and superior parietal cortex, SPC). Because V5 shows both motion and attention effects, it is useful to mask the motion-contrast with the attention-contrast when extracting time series for V5. You should also compute an "effects of interest" F-contrast, which is needed for mean-correcting the extracted time series (see below).

1. From the main SPM window, click on the BATCH button.
2. Add a module "SPM > Stats > Contrast manager".
3. For SELECT SPM.MAT, enter the one that has been created in the previous step.
4. Under CONTRAST SESSIONS, choose one NEW: F-CONTRAST and three NEW: T-CONTRAST and enter
  - F-contrast: NAME = `Effects of interest`, F CONTRAST VECTOR = `eye(3)`.
  - T-contrast: NAME = `Photic`, T CONTRAST VECTOR = [`1 0 0`].
  - T-contrast: NAME = `Motion`, T CONTRAST VECTOR = [`0 1 0`].
  - T-contrast: NAME = `Attention`, T CONTRAST VECTOR = [`0 0 1`].
5. Press the RUN green arrow at the top of the Batch Editor window. This will specify and estimate these 4 contrasts.

Here is now a step-by-step example for extracting the V5 time series:

1. Press **RESULTS**.
2. Select the **SPM.mat** file.
3. Choose the t-contrast for the **Motion** condition.
4. Apply masking: **contrast**
5. Choose the t-contrast for the **Attention** condition.
6. Uncorrected mask p-value  $p \leq 0.05$  and nature of mask: **inclusive**.
7. p value adjustment to control: **none** with a threshold of 0.001 and extent 0
8. To overlay these results on a structural scan, click “overlays...” in the SPM Results window, then click “sections”. Navigate to the structural folder and select the file named “nsM00587\_0002.img”.
9. Select the global maxima that looks V5-ish, e.g. [-36 -87 -3].
10. Press the **EIGENVARIATE** button.
11. Name of region: **V5**
12. Adjust data for: **Effects of interest** (this effectively mean-corrects the time series)
13. VOI definition: **sphere**
14. VOI radius(mm): e.g. **8 mm**

SPM now computes the first principal component of the time series from all voxels included in the sphere. The result is stored (together with the original time series) in a file named **VOI\_V5\_1.mat** in the working directory (the “1” refers to session 1).

You can now proceed to select time series for V1 (using the **PHOTIC** contrast with “Apply masking” set to none) with an 8 mm sphere centered on V1 (here we are using coordinates [0 -93 18]). Do the same thing with **SPC** (using the **ATTENTION** contrast) with a sphere centred on [-27 -84 36]. This will create files **VOI\_V1\_1.mat** and **VOI\_SPC\_1.mat**.

### 35.3.3 Specifying and estimating the DCM

Now we have defined the inputs (via the design matrix) and the time series, we are ready to build the DCM. We will look at a simplified version of the model described in [34]. In our example here, we will model a hierarchically connected system comprising V1, V5 and SPC, i.e. reciprocal connections between V1-V5 and V5-SPC, but not between V1-SPC. We will assume that (i) V1 is driven by any kind of visual stimulation (direct input “photic”), (ii) motion-related responses in V5 can be explained through an increase in the influence of V1 onto V5 whenever the stimuli are moving (i.e. “motion” acts as modulatory input onto the  $V1 \rightarrow V5$  connection) and (iii) attention enhances the influence of SPC onto V5 (i.e. “attention” acts as modulatory input onto the  $SPC \rightarrow V5$  connection). This DCM is shown schematically in Figure 35.4, and can be made as follows:

1. Press the large **Dynamic Causal Modelling** button.
2. Choose **SPECIFY**.
3. Select the **SPM.mat** file you just created when specifying the GLM.
4. Name for **DCM\_???.mat**: e.g. **mod\_bwd** (for “attentional modulation of backward connection”).
5. Select all VOIs in order **VOI\_V1\_1**, **VOI\_V5\_1**, **VOI\_SPC\_1**.

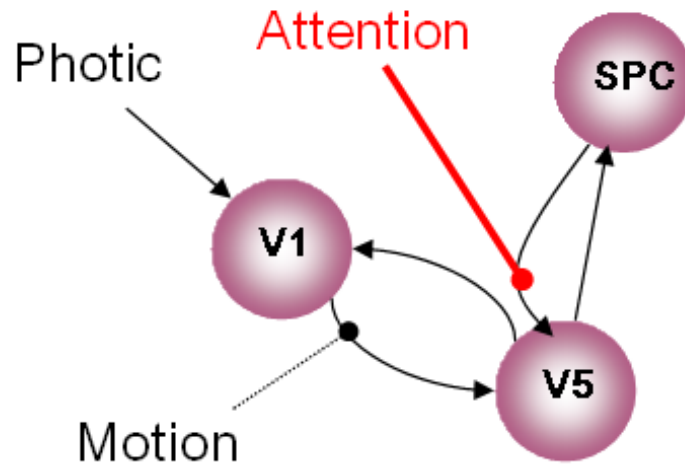


Figure 35.4: *DCM with attentional modulation of backwards connection. Dotted lines denote modulatory connections.*

6. Include **Photic**: Yes
7. Include **Motion**: Yes
8. Include **Attention**: Yes
9. Specify slice timings for each area. The default values are set to the last slice of the data, which was the default in the original DCM version. For sequential (as opposed to interleaved) data, this modelling option allows to use DCM in combination with any TR (slice timing differences) [66]. Here, we proceed with the default values.
10. Enter 0.04 for “Echo Time, TE[s]”.
11. Modulatory effects: **bilinear**
12. States per region: **one**
13. Stochastic effects: **no**
14. Centre input: **no**
15. Define the following extrinsic connections: V1 to V5, V5 to V1, V5 to SPC, SPC to V5, i.e. a hierarchy with reciprocal connections between neighbouring areas. Note that the columns specify the source of the connection and the rows specify its target. Your connectivity matrix should look like the one in Fig. 35.5.
16. Specify **Photic** as a driving input into V1. See Fig. 35.5
17. Specify **Motion** to modulate the connection from V1 to V5. See Fig. 35.5
18. Specify **Attention** to modulate the connection from SPC to V5. See Fig. 35.5

A polite “Thank you” completes the model specification process. A file called `DCM_mod_bwd.mat` will have been generated.

You can now estimate the model parameters, either by pressing the DCM button again and choosing `ESTIMATE (TIME-SERIES)`, or by typing

```
>> spm_dcm_estimate('DCM_mod_bwd');
```



Figure 35.5: Specification of model depicted in Fig 35.4. **Top left:** Filled circles define the structure of the extrinsic connections  $A$  such that eg. there are no connections from  $V1$  to  $SPC$  or from  $SPC$  to  $V1$ . **Top right:** The filled circle specifies that the input *Photic* connects to region  $V1$ . **Bottom left:** The filled circle indicates that the input *Motion* can modulate the connection from  $V1$  to  $V5$ . This specifies a “modulatory” connection. **Bottom right:** The filled circle indicates that *Attention* can modulate the connection from  $SPC$  to  $V5$ .

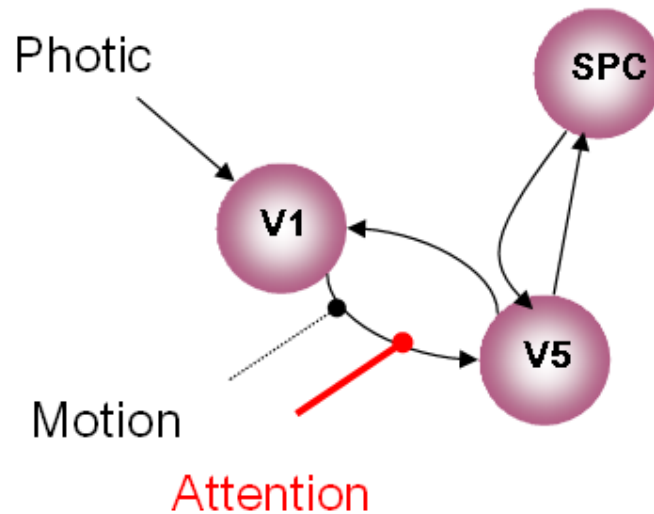


Figure 35.6: DCM with attentional modulation of forwards connection. Dotted lines denote modulatory connections.

from the MATLAB command line.

Once this is completed, you can review the results as follows:

1. Press the DCM button.
2. Choose REVIEW.
3. Select DCM\_mod\_bwd.mat

By clicking “review...” you can now select from multiple options, e.g. you can revisit the fit of the model (“Outputs”) or look at the parameter estimates for the endogenous coupling (“Intrinsic connections”) or for the parameters associated with the driving or modulatory inputs (“Effects of Photic”, “Effects of Motion”, “Effects of Attention”).

Also, you can use the “Contrasts” option to determine how confident you can be that a contrast of certain parameter estimates exceeds the threshold you chose in step 4. Of course, you can also explore the model results at the level of the MATLAB command line by loading the model and inspecting the parameter estimates directly. These can be found in DCM.Ep.A (endogenous coupling), DCM.Ep.B (modulatory inputs) and DCM.Ep.C (driving inputs).

### 35.3.4 Comparing models

Let us now specify an alternative model and compare it against the one that we defined and estimated above. The change that we are going to make is to assume that attention modulates the  $V1 \rightarrow V5$  connection (as opposed to the  $SPC \rightarrow V5$  connection in the previous model). For defining this model, you repeat all the steps from the above example, the only difference being that the model gets a new name (e.g. `mod_fwd`) and that attention now acts on the forward connection. This DCM is shown schematically in Figure 35.6.

Estimate this new model as above. You can then perform a Bayesian model comparison as follows:

1. Press the “DCM” button.
2. Choose COMPARE.
3. In the Batch Editor window that opened, fill in the “BMS: DCM” module:

- (a) **Directory:** choose current directory,
  - (b) **Data:** add a New Subject with a New Session and select the two models, e.g. in the order `DCM_mod_bwd.mat` and `DCM_mod_fwd.mat`,
  - (c) **Inference method:** choose “Fixed effects (FFX)”.
4. Press Run (the green triangle in the Batch Editor).

The Graphics window, Fig. 35.8, now shows a bar plot of the model evidence. You can see that our second model is better than the first one.

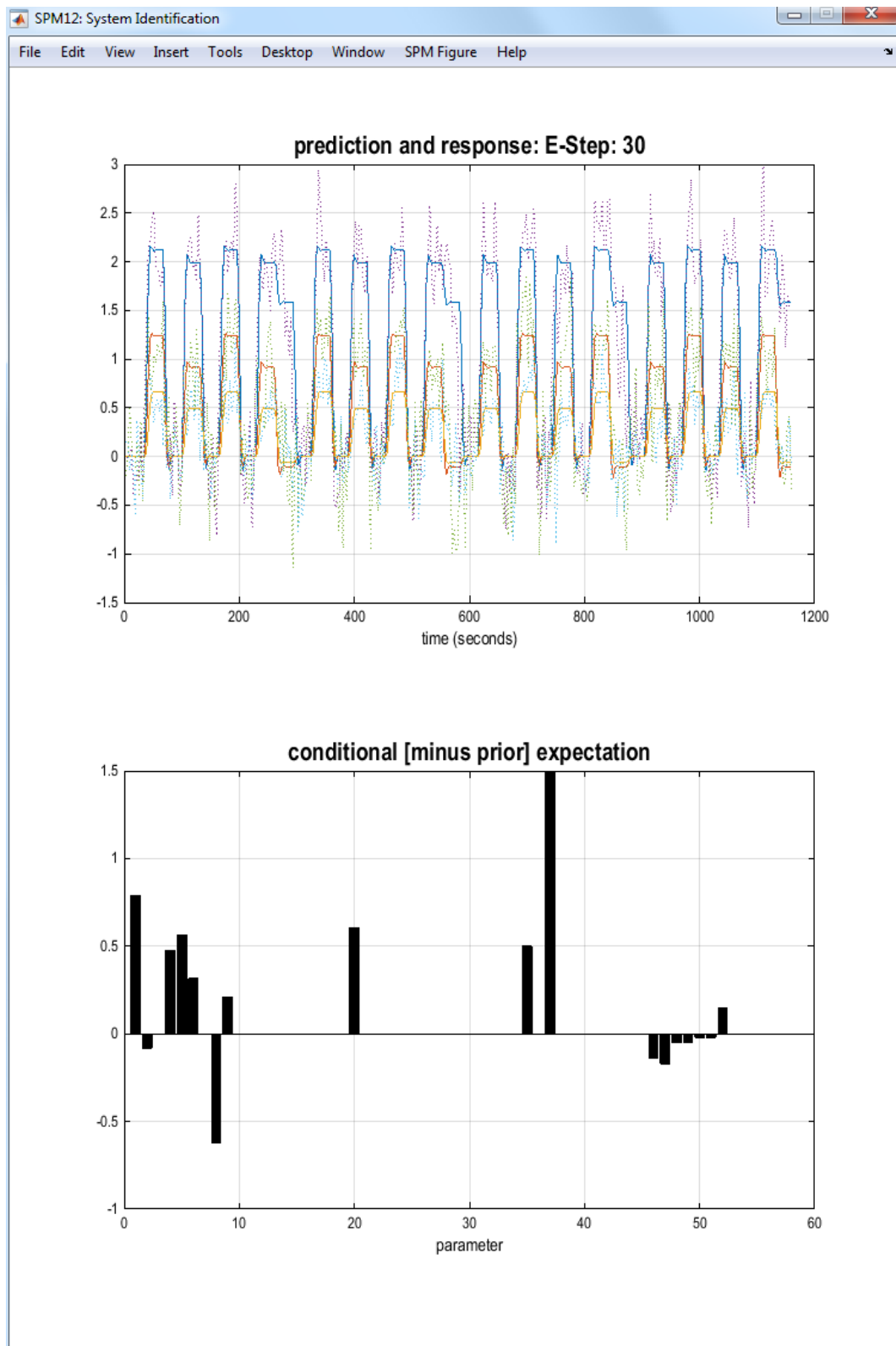


Figure 35.7: Plot of predicted and observed response, after convergence and conditional expectation of the parameters.

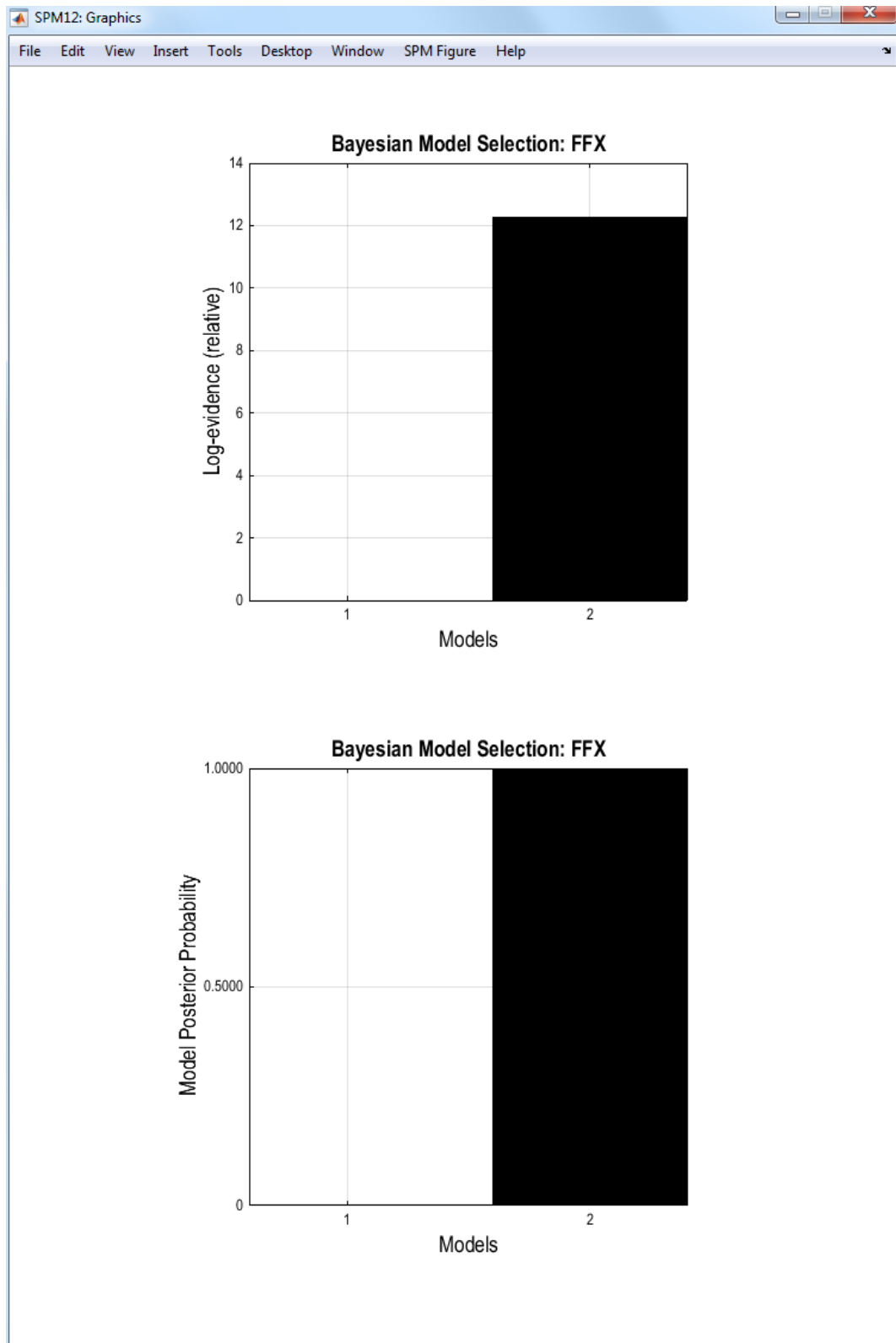


Figure 35.8: Model 2 (shown in Fig 35.6) is preferred to model 1 (shown in Fig 35.4).



## Chapter 36

# Psychophysiological Interactions (PPI)

### 36.1 Theoretical background

Psychophysiological interactions (PPI) and the related technique of physiophysiological interactions ( $\Phi$ PI) are based on extensions to statistical models of factorial designs. Table 1 illustrates a classic  $2 \times 2$  factorial design.

Table 36.1.  $2 \times 2$  factorial design in Table format

		Factor A	
		Level 1	Level 2
Factor B	Level 1	$A_1/B_1$	$A_2/B_1$
	Level 2	$A_1/B_2$	$A_2/B_2$

The equation for factorial design is given by [36.1](#).

$$y = (A_2 - A_1)\beta_1 + (B_2 - B_1)\beta_2 + (A_2 - A_1)(B_2 - B_1)\beta_3 + G\beta_4 + \epsilon \quad (36.1)$$

Notice that this equation includes both of the main effects terms  $(A_2 - A_1)\beta_1$  for factor A, and  $(B_2 - B_1)\beta_2$  for factor B, as well as the interaction term  $(A_2 - A_1)(B_2 - B_1)\beta_3$ . It also contains a term for the confounds  $G\beta_4$  such as movement parameters, session effects, etc. The inclusion of main effects when estimating interactions is very important, and their inclusion in the design cannot be stressed enough. If the main effects are not included, then we cannot be sure that estimates of the interaction term are not confounded by main effects.

To extend the concept of factorial designs to PPI's the basic idea is to substitute (neural) activity from one cerebral region for one of the factors. Equation [36.2](#) illustrates this concept after substituting activity in area V1 for factor A.

$$y = V1\beta_1 + (B_2 - B_1)\beta_2 + (V1 \times (B_2 - B_1))\beta_3 + G\beta_4 + \epsilon \quad (36.2)$$

Similarly, for psychophysiological interactions activity from 2 cerebral regions (V1 and posterior parietal (PP)) are used as the main effects, as shown in equation [36.3](#)

$$y = V1\beta_1 + PP\beta_2 + (V1 \times PP)\beta_3 + G\beta_4 + \epsilon \quad (36.3)$$

Again, notice that all 3 equations [36.1](#), [36.2](#) and [36.3](#) have 3 terms (aside from confounds and error) – the two main effects and the interaction. Therefore, the design matrix must include at

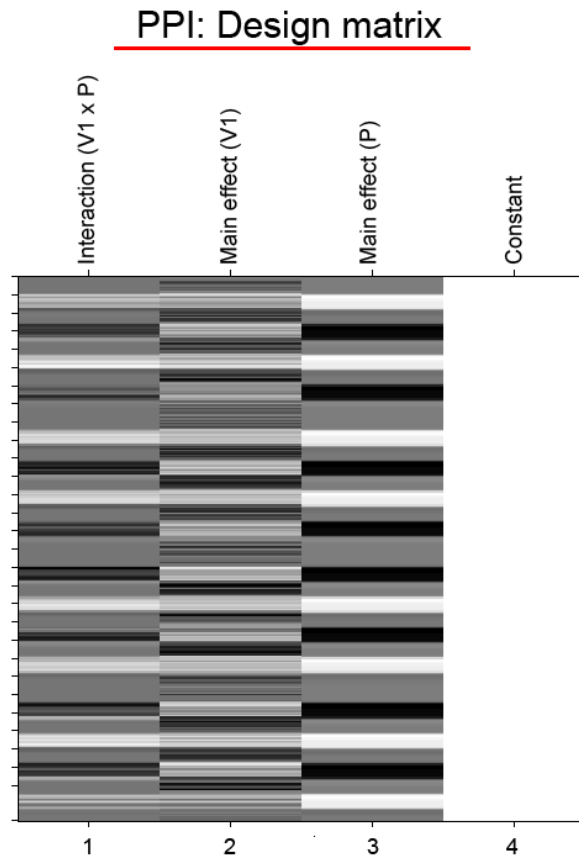


Figure 36.1: Example design matrix for a PPI (or  $\Phi$ PPI). The main effects are BOLD activity from area V1, in column 2, and a psychological vector, e.g., attention vs. no attention (P), in column 3. Inference would typically focus on the interaction term, in column 1, using a contrast vector of  $[1 \ 0 \ 0 \ 0]$ . In  $\Phi$ PPIs the third column would be BOLD activity from a second source region rather than the psychological factor.

least 3 columns, one for each main effect and one for the interaction. A basic design matrix for PPI's is shown in Figure 36.1.

Both PPIs and  $\Phi$ PPIs can be conceived of as models of “contribution”. PPIs occupy middle-ground between models of functional vs. effective connectivity [30]. Functional connectivity (FC) is defined as the temporal correlation between spatially separated neurophysiological events [30]. FC analyses are typically model-free and do not specify a direction of influence, i.e., the influence of A on B is indistinguishable from the influence of B on A. In contrast, PPI's are based on regression models, and therefore a direction of influence is chosen based on the model. Effective connectivity (EC) is defined as the influence one neural system has on another [27]. PPIs are closely related to EC models, but because PPIs are generally very simple (i.e., 1 source region and 1 experimental factor, or 2 source regions in the case of  $\Phi$ PPIs) they are very limited models of EC.

The interaction between the source region and experimental context (or two source regions) can be interpreted in 2 different ways: 1) as demonstrating how the contribution of one region to another is altered by the experimental context or task, or 2) as an example of how an area's response to an experimental context is modulated by input from another region, Figure 36.2.

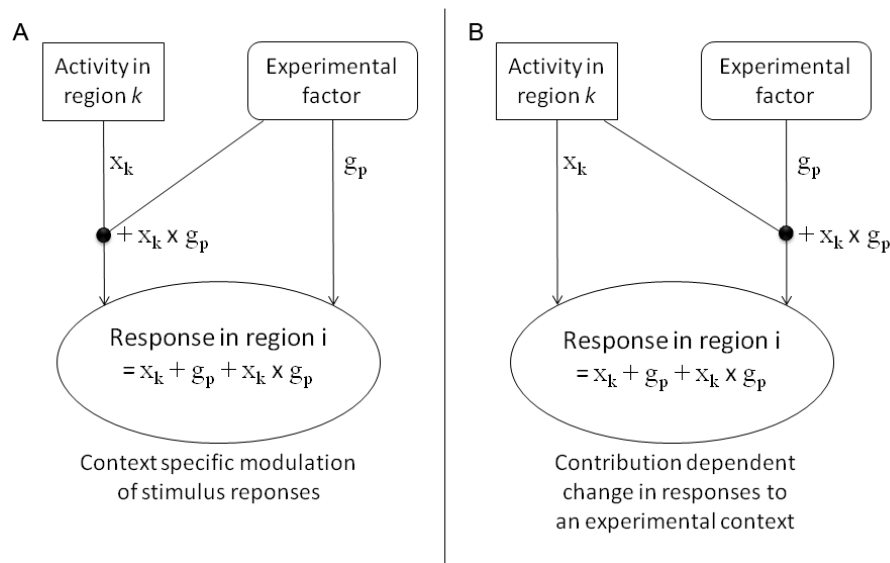


Figure 36.2: Two alternative interpretations of PPI effects. A) The contribution of one area ( $k$ ) to another ( $i$ ) is altered by the experimental (psychological) context. B) The response of an area ( $i$ ) to an experimental (psychological) context due to the contribution of region ( $k$ ). (Adapted from [30])

## 36.2 Psycho-Physiologic Interaction Analysis: Summary of Steps

Mechanistically, a PPI analysis involves the following steps.

1. Performing a standard GLM analysis.
2. Extracting BOLD signal from a source region identified in the GLM analysis.
3. Forming the interaction term (source signal  $\times$  experimental treatment)
4. Performing a second GLM analysis that includes the interaction term, the source region's extracted signal and the experimental vector in the design. The inclusion of the source region's signal and the experimental vector is analogous to including the main effects in an ANOVA in order to make an inference on the interaction.

Forming the proper interaction term turns out to be a challenge because of the unique characteristics of fMRI (BOLD) data in which the underlying neural signal is convolved with a hemodynamic response function. However, interactions in the brain take place at the neural and not the hemodynamic level. Therefore, appropriate models of the interactions require the neural signal, which is not measured directly, but instead must be derived by deconvolving the HRF. The PPI software (`spm_peb_ppi.m`) was developed in order to provide robust deconvolution of the HRF and the proper derivation of the interaction term [48].

## 36.3 Practical example

The dataset in this exercise is from one subject who was studied in the [15] report and refers to the “attention to motion” dataset available from the SPM website<sup>1</sup>. It has already been described in the previous chapter for DCM.

The goal is to use PPI to examine the change in effective connectivity between V2 and V5 while the subject observes visual motion (radially moving dots) under the experimental treatments of attending vs. not attending to the speed of the dots. The psychophysiological interaction can be

<sup>1</sup><http://www.fil.ion.ucl.ac.uk/spm/data/attention/>

conceived of as looking for a significant difference in the regression slopes of V1 vs. V5 activity under the influence of the different attentional states [30].

### 36.3.1 GLM analysis - Design setup and estimation

This dataset has already been preprocessed (coregistered, normalised and smoothed) using an earlier version of SPM.

1. The analysis directory should include
  - (a) A directory named `functional`, which includes the preprocessed fMRI volumes.
  - (b) A directory named `structural`, which includes a T1 structural volume
  - (c) Files: `factors.mat`, `block_regressors.mat`, `multi_condition.mat` and `multi_block_regressors.mat`.
  - (d) You will also need to make 2 empty directories called `GLM` and `PPI` for performing the analyses.
2. In MATLAB type
 

```
>> cd GLM
>> spm_fmri
```
3. Start the Batch system by clicking the `BATCH` button.
4. From the SPM menu in the Batch window, click `STATS` and then select the modules `FMRI MODEL SPECIFICATION`, `MODEL ESTIMATION` and `CONTRAST MANAGER`, Figure 36.3.

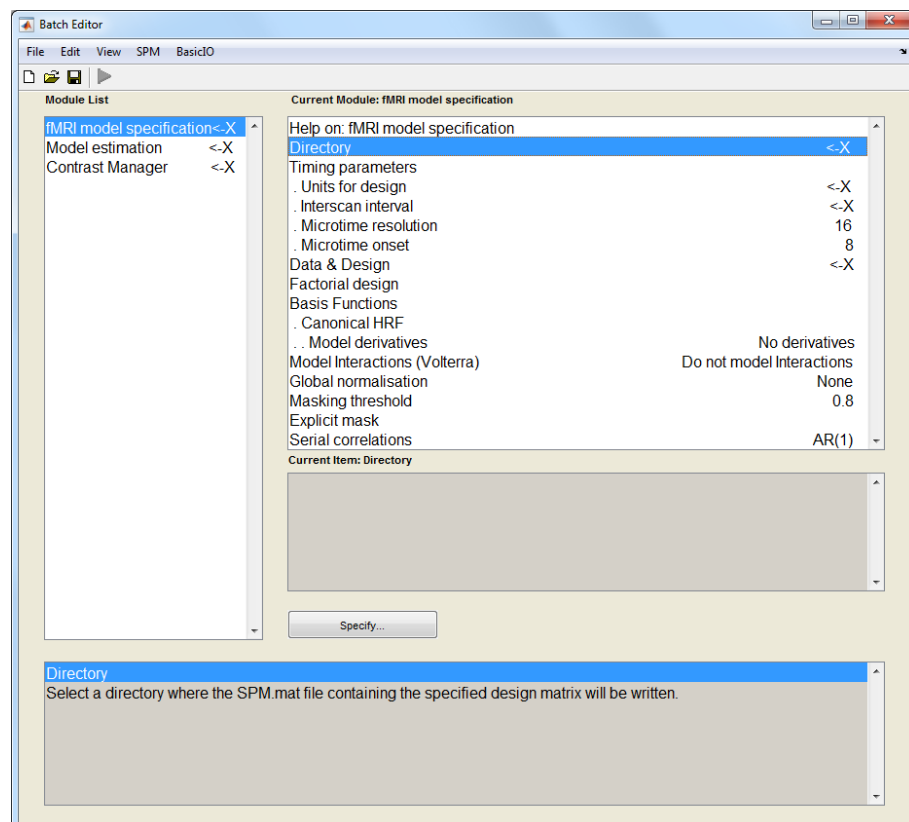
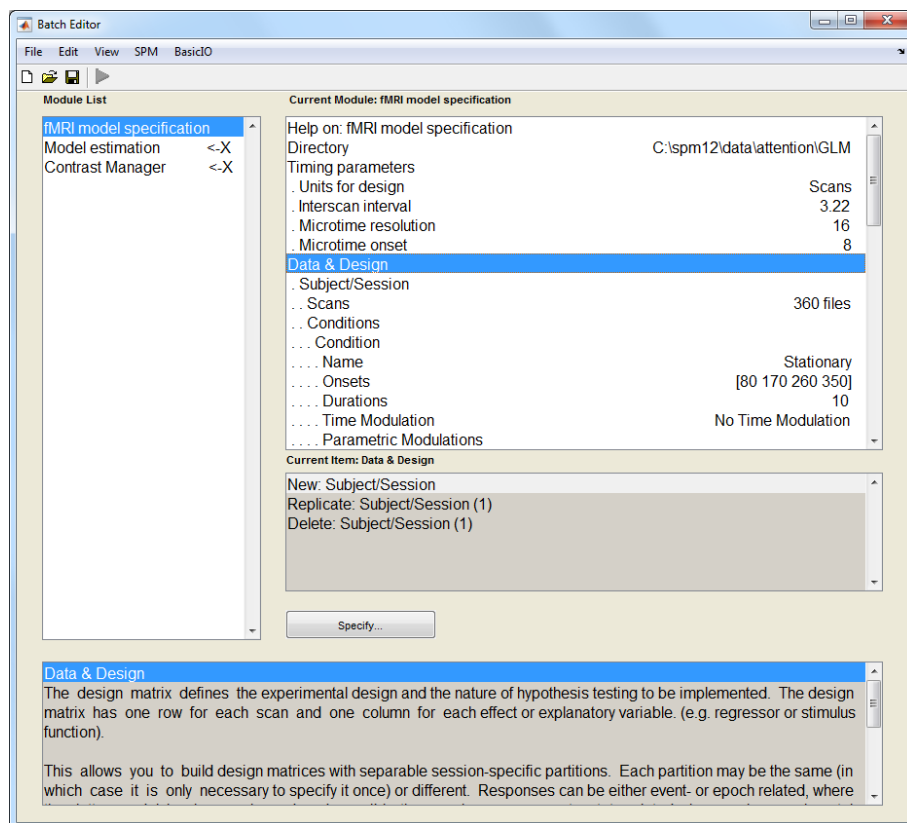


Figure 36.3: Batch Editor showing the FMRI MODEL SPECIFICATION, MODEL ESTIMATION and CONTRAST MANAGER modules.

#### Fill in the FMRI MODEL SPECIFICATION

5. Click DIRECTORY and choose the GLM directory that you made above.
6. UNITS FOR DESIGN [SCANS]
7. INTERSCAN INTERVAL [3.22]
8. Click DATA & DESIGN. Then in the CURRENT ITEM box click NEW: SUBJECT/SESSION, Figure 36.4.

Figure 36.4: *Fill in the Data & Design*

9. Click SCANS and choose all the functional scans `snffM00587_00xx.img`. There should be 360 \*.img files.
10. The experimental conditions can be defined either individually or using a multiple condition `mat`-file. This exercise shows both methods for educational purposes. When doing an actual analysis you can just follow one of the two approaches below.

### Define conditions individually

11. Load the `mat` file containing the individual conditions:

```
>> load factors.mat
```

You can look at the loaded variables by typing the variable names. (`stat` = stationary, `natt` = no attention, `att` = attention)

```
>> stat
>> natt
>> att
```

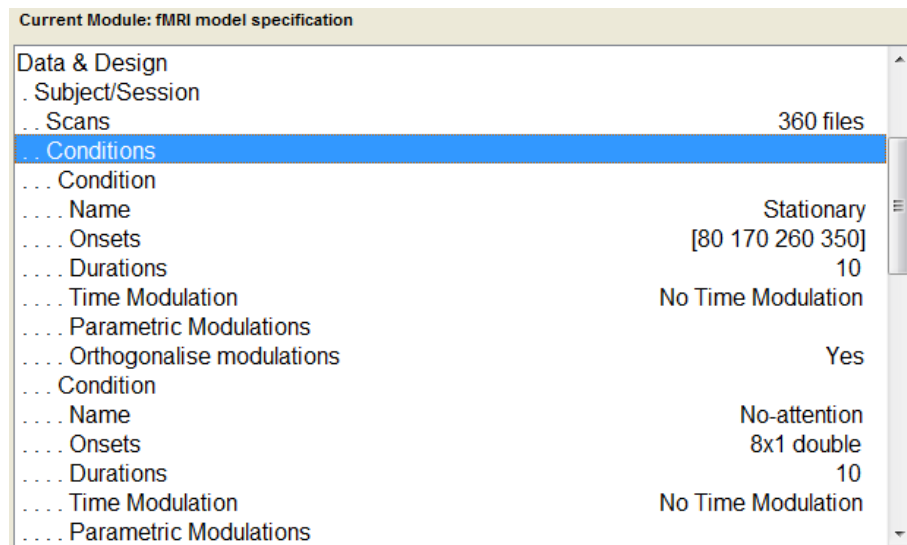


Figure 36.5: CURRENT MODULE section of the BATCH EDITOR showing 3 Conditions to be filled in.

12. Click CONDITIONS then in the CURRENT ITEM box click NEW: CONDITION 3 times, Figure 36.5.
13. Condition 1: Name = Stationary, ONSETS = stat, DURATIONS = 10.
14. Condition 2: Name = No-attention, ONSETS = natt, DURATIONS = 10.
15. Condition 3: Name = Attention, ONSETS = att, DURATIONS = 10.
16. Next you will enter 3 regressors to model block effects. This will account for the fact that the experiment took place over 4 runs that have been concatenated into a single session to make the PPI analysis easier. *Note: Only 3 of the 4 sessions need to be modeled by block regressors because the fourth one is modeled by the mean column of the design matrix.*

First load the regressors:

```
>> load block_regressor.mat
```

17. Click REGRESSORS then click NEW: REGRESSOR 3 times in the CURRENT ITEM box, Figure 36.6.
18. Regressor 1: NAME = Block 1, VALUE = block1
19. Regressor 2: NAME = Block 2, VALUE = block2
20. Regressor 3: NAME = Block 3, VALUE = block3

#### Define conditions using multiple condition and multiple regressor files

21. If you would like to look at the organization of the variables in the multiple condition file, first load it.

```
>> load multi_condition.mat
>> names
>> onsets
>> durations
```

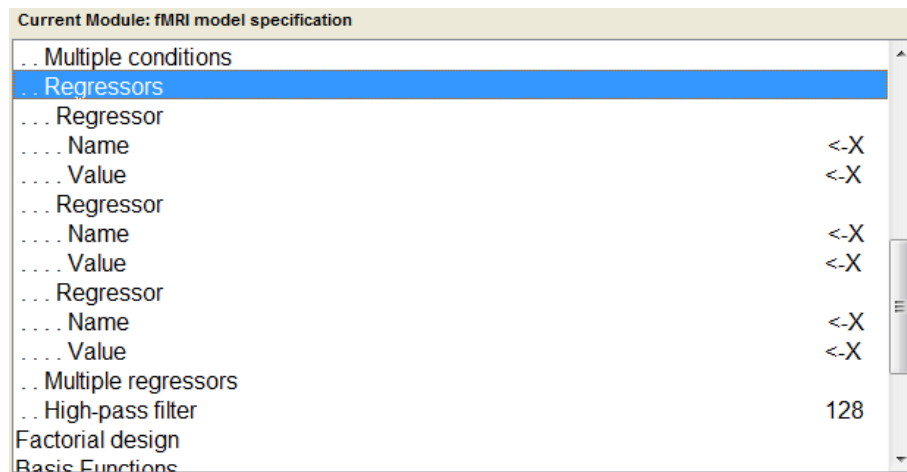


Figure 36.6: CURRENT MODULE section of the BATCH EDITOR showing 3 Regressors to be filled in.

The variables in a multiple condition file must always be named: 'names', 'onsets', and 'durations'. Notice that these three variables are cell arrays. (*Note: You only need to do this step if you want to look at the organization of the variables. In contrast to defining conditions individually, as shown above, when using a multiple condition file you do not have to load the file in order to enter it into the design.*)

22. To use the multiple conditions file in the design, click MULTIPLE CONDITIONS, then SPECIFY FILES in the Options box and choose the `multi_condition.mat` file.
23. Next you will enter the 3 regressors to model block effects by using a multiple regressor file. To look at the organization of the multiple regressor variable, first load it. (*Again you do not have to load the multiple regressor file in order to use it. This step is just to allow you to examine the file and the variables it contains.*)

```
>> load multi_block_regressor.mat
>> R
```

Notice that this file contains a single variable, R, which is a 360 x 3 matrix. The number of rows is equal to the number of scans, and each regressor is in a separate column.

24. To use the multiple regressor file, click MULTIPLE REGRESSORS then select the `multi_block_regressor.mat` file.

### Complete the design setup

25. HIGH-PASS FILTER [192] (*Note: most designs will use a high-pass filter value of 128. However, this dataset requires a longer high-pass filter in order not to lose the low frequency components of the design.*)
26. FACTORIAL DESIGN is not used
27. The BASIS FUNCTION is the CANONICAL HRF as shown and MODEL DERIVATIVES [NO DERIVATIVES]
28. MODEL INTERACTIONS (VOLTERRA): [DO NOT MODEL INTERACTIONS]
29. GLOBAL NORMALISATION [NONE]
30. EXPLICIT MASK [NONE]

## 31. SERIAL CORRELATIONS [AR(1)]

**Model Estimation**

32. Under MODEL ESTIMATION click SELECT `SPM.MAT` then click the DEPENDENCY button and choose FMRI MODEL SPECIFICATION: `SPM.MAT FILE`. The METHOD should be left as Classical.

**Contrast Manager**

33. Under CONTRAST MANAGER click SELECT `SPM.MAT` then click the DEPENDENCY button and choose MODEL ESTIMATION: `SPM.MAT FILE`
34. Click CONTRAST SESSIONS then click NEW: F-CONTRAST once, and NEW: T-CONTRAST twice from the CURRENT ITEM box.
35. Click CONTRAST VECTORS and then NEW: F CONTRAST.
36. The F contrast vector (named “effects of interest”) can be entered as `[eye(3), zeros(3,4)]`, which will produce:

```
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
```

37. For the first T-contrast, NAME is `Attention`, and the T CONTRAST VECTOR is `0 -1 1 0 0 0 0` (Note the order of the conditions in the design matrix is: `Stationary`, `NoAttMot` and `AttMot`).
38. For the second T-contrast NAME is `Motion`, and the T CONTRAST VECTOR is: `-2 1 1 0 0 0 0`.
39. Click the SAVE icon on the toolbar and save the batch file.

**Design estimation**

40. If everything has been entered correctly the RUN button should now be green. Click RUN to estimate the design.
41. The design matrix should look as shown in Figure 36.7, below.

**36.3.2 GLM analysis - Results**

1. Click RESULTS and select the `SPM.mat` file.
2. Choose the `Attention` contrast
3. Apply masking [None]
4. p value adjustment to control [None]
5. threshold T or p value [0.0001]
6. & extent threshold voxels [10]
7. You should see an SPM that looks like the one shown below, Figure 36.8. Note the Superior Parietal and Dorso-Lateral Prefrontal activations, among others. By selecting OVERLAYS → SECTIONS, and selecting the normalised structural image, you should be able to identify the anatomy more accurately.



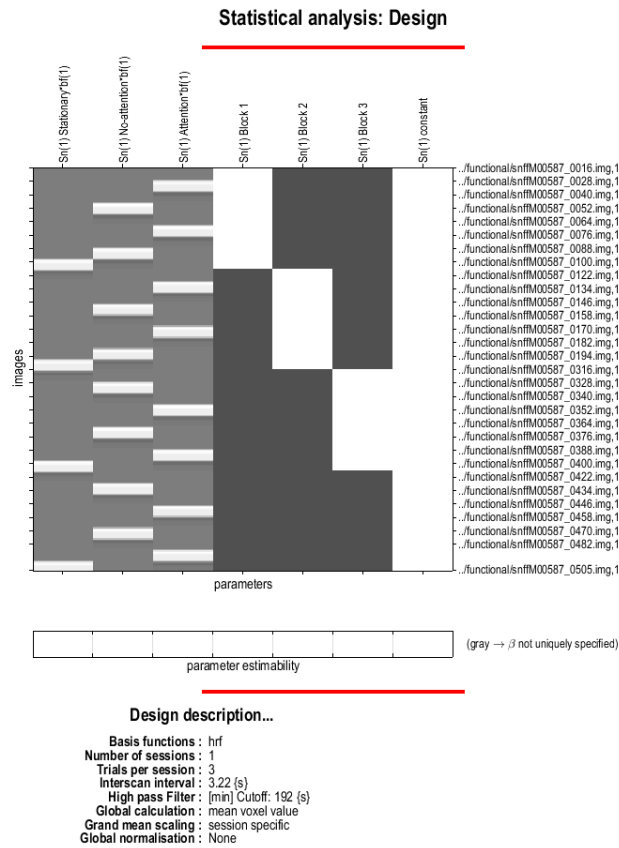


Figure 36.7: Design matrix

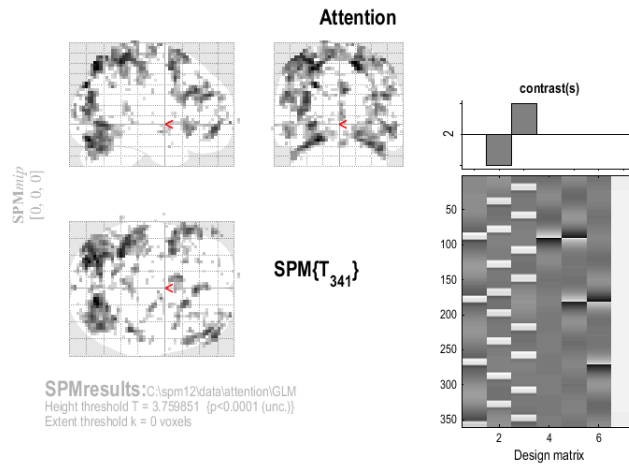


Figure 36.8: Statistical parametric map for the Attention contrast

8. To look at the Motion contrast where Attention is greater than No Attention, click RESULTS, choose the SPM.mat file and choose the Motion contrast.
9. apply masking [Contrast]
10. Select contrast for masking: Choose the Attention contrast
11. Uncorrected mask p-value [0.01]
12. Nature of Mask: [inclusive]

13. p value adjustment to control [FWE]
14. threshold T or p value [0.05]
15. & extent threshold voxels [3]
16. The masked motion contrast on the glass brain is shown below in Figure 36.9.

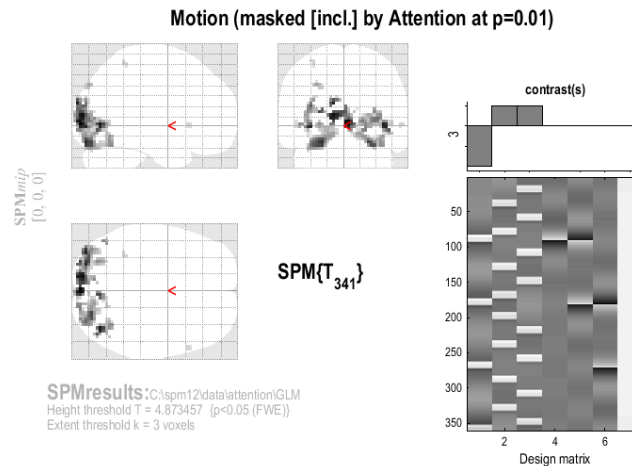


Figure 36.9: *Statistical parametric map for the Motion contrast inclusively masked by the Attention contrast*

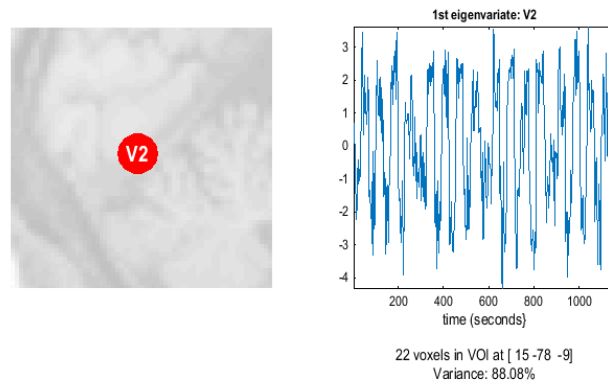
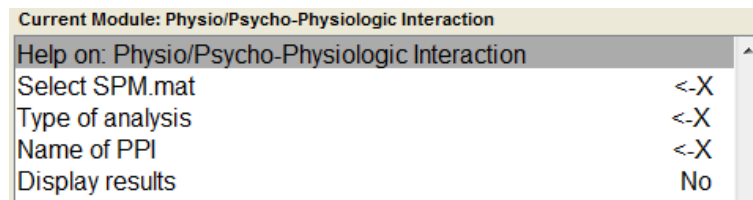
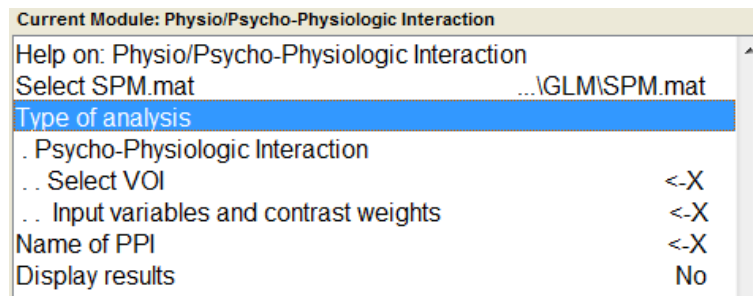
## 36.4 GLM analysis - Extracting VOIs

1. First select the **Motion** contrast, but do not include masking. Use a p-value adjustment of FWE with height threshold of 0.05 and a cluster threshold of 3.
2. Go to point [15 -78 -9]
3. Press **eigenvariate**
4. Name of region [V2]
5. Adjust data for [effects of interest]
6. VOI definition [sphere]
7. VOI radius(mm) [6]

This saves the extracted VOI data in the file `VOI_V2_1.mat` in the working directory, and displays Figure 36.10, below. The left side shows the location on a standard brain. The right side shows the first eigenvariate of the extracted BOLD signal.

## 36.5 PPI analysis - Create PPI variable

1. PPIs can be calculated either by pressing the PPIs button in the SPM MENU window, or by selecting the **PHYSIO/PSYCHO-PHYSIOLOGIC** menu item from the SPM → Stats menu of the **BATCH EDITOR**. This example uses the **BATCH EDITOR**, Figure 36.11.
2. Choose the `SPM.MAT` file in the **GLM** directory.
3. Type of analysis: Choose **PSYCHO-PHYSIOLOGIC INTERACTION**, Figure 36.12.
4. Select VOI: Choose `VOI_V2_1.mat`

Figure 36.10: *First eigenvariate of the extracted BOLD signal in V2*Figure 36.11: *Physio/Psycho-Physiologic module in the Batch Editor*Figure 36.12: *Specifying a Psycho-Physiologic interaction.*

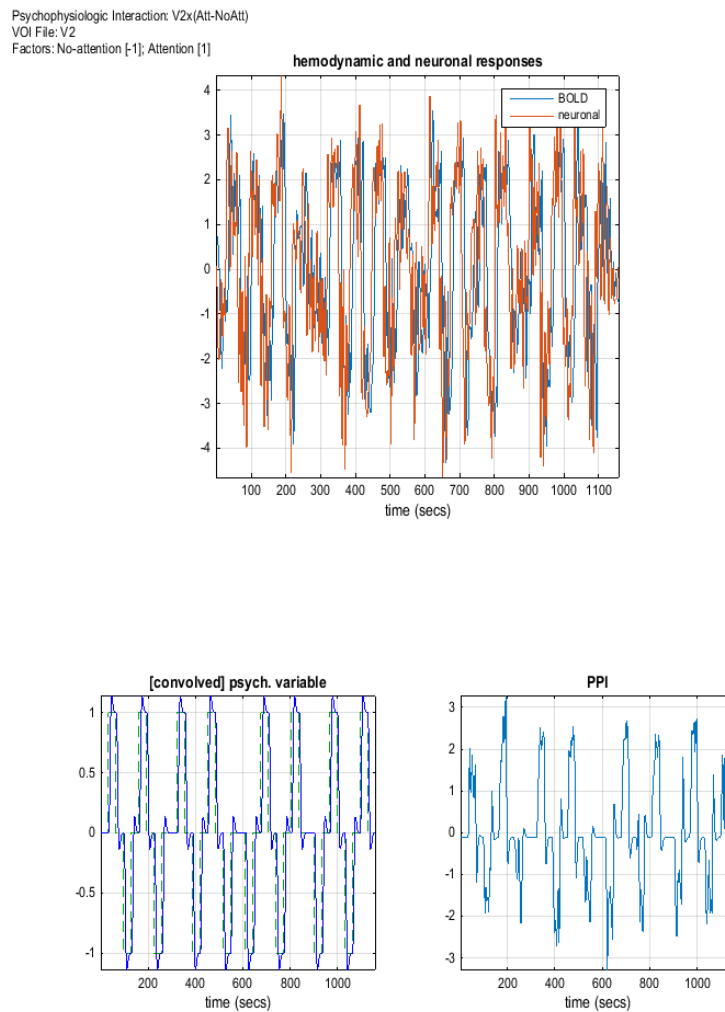
5. Input variables and contrast weights: Must be specified as an  $n \times 3$  matrix, where  $n$  is the number of conditions included in the PPI. The first column of the matrix indexes `SPM.Sess.U(i)`. The second column indexes `SPM.Sess.U(i).name{ii}`. It will generally be a 1 unless there are parametric effects. The third column is the contrast weight. In order to include Attention - No-attention in the PPI, recall that the conditions were entered as: Stationary, No-attention, Attention, therefore the matrix should be.

$$[2 \ 1 \ -1; \ 3 \ 1 \ 1]$$

6. Name of PPI [ V2x(Att-NoAtt) ]

7. Display results: Yes

After a few seconds the PPI will be calculated and a graphical window will appear, Figure 36.13. In the upper left, the details of the PPI setup calculation are given including the name of the PPI, the chosen VOI file, and the included conditions and their contrast weights. The main central graph shows the original BOLD signal (actually the eigenvariate) in blue and the neuronal or deconvolved signal in green. These will look quite similar for block design data. The graph in the lower left shows the task condition plot, dotted green line, and the convolved task conditions (psych variable). In the lower right the PPI interaction term is plotted.

Figure 36.13: *PPI output graphics*

The PPI calculation will create a file `PPI_V2x(Att-NoAtt).mat` in the current working directory. It contains the variable `PPI.ppi` (the interaction term), `PPI.Y` (the original VOI eigenvariate) and `PPI.P` (the `Attention - No Attention` task vector). You will use these vectors in setting up your psychophysiological interaction GLM analysis. See `spm_peb_ppi` for a full description of the PPI data structure.

### 36.5.1 PPI GLM analysis - Design setup and estimation

1. Copy the file `PPI_V2x(Att-NoAtt)` MAT-file to the PPI directory that you created at the start of this exercise.
2. Change directory to the new one, i.e. `cd PPI`
3. At the MATLAB prompt type
 

```
>> load PPI_V2x(Att-NoAtt)
```
4. In the BATCH EDITOR setup another GLM analysis by choosing the modules FMRI MODEL SPECIFICATION, MODEL ESTIMATION and CONTRAST MANAGER as you did above, and fill it in as follows.
5. Directory: Choose the PPI directory

6. Units for design [scans]
7. Interscan interval [3.22]
8. Add a NEW: SUBJECT/SESSION under DATA & DESIGN
9. Click SCANS and choose all the functional scans `snffM00587_00xx.img`. There should be 360 \*.img files.
10. Click NEW: REGRESSOR and add 6 regressors.
11. Regressor 1: NAME = PPI-interaction, VALUE = PPI.ppi
12. Regressor 2: NAME = V2-BOLD, VALUE = PPI.Y
13. Regressor 3: NAME = Psych\_Att-NoAtt, VALUE = PPI.P
14. Regressor 4: NAME = Block 1, VALUE = block1
15. Regressor 5: NAME = Block 2, VALUE = block2
16. Regressor 6: NAME = Block 3, VALUE = block3
17. High Pass Filter [192]

#### Model Estimation

18. Under MODEL ESTIMATION click SELECT SPM.MAT then click the DEPENDENCY button and choose FMRI MODEL SPECIFICATION: SPM.MAT FILE. The METHOD should be left as Classical.

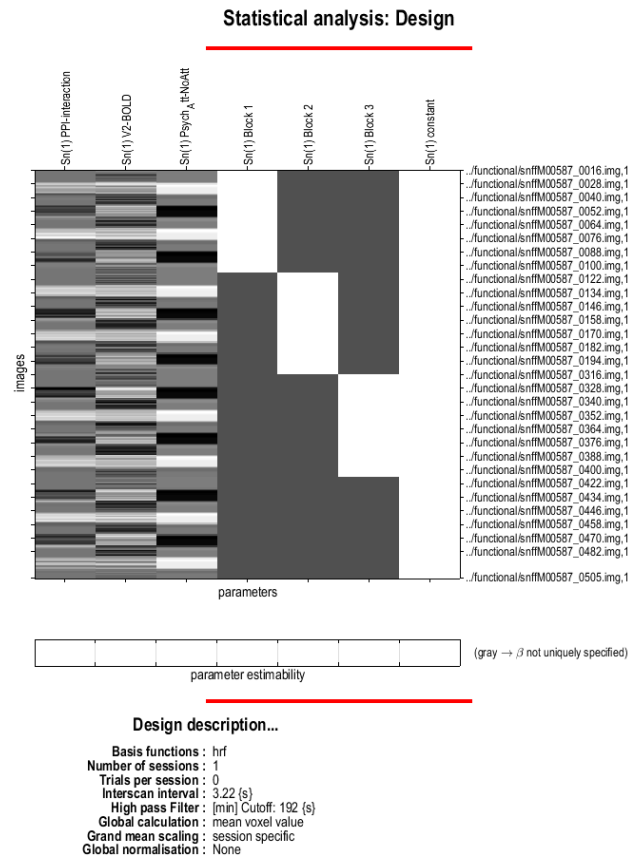
#### Contrast Manager

19. Under CONTRAST MANAGER click SELECT SPM.MAT then click the DEPENDENCY button and choose MODEL ESTIMATION: SPM.MAT FILE
20. Click CONTRAST SESSIONS then click NEW: T-CONTRAST
21. T-contrast, NAME: PPI-Interaction, vector: 1 0 0 0 0 0 0
22. Save the batch file.
23. Run

The design matrix is shown below, Figure 36.14.

### 36.5.2 PPI analysis - Results

1. Press the RESULTS button and select the SPM.mat file in the PPI directory.
2. Choose the PPI-Interaction contrast
3. apply masking [No]
4. p value adjustment to control [None]
5. threshold T or p value [0.01]
6. & extent threshold voxels [10]
7. You should see an SPM that looks the same as the one shown below in the top part of Figure 36.15. The resulting SPM shows areas showing differential connectivity to V2 due to the effect of attention vs. no attention conditions. The effect in this subject is weak.

Figure 36.14: *Design matrix for the PPI analysis*

### 36.5.3 PPI analysis - Plotting

1. One region showing the psychophysiological interaction is the V5 region, which is located at [39 -72 0] in this subject. Move the cursor to this point to view the area of activation, as shown below, in the bottom half of Figure 36.15.
2. In order to plot the PPI graph showing the effect of attention, you need to extract a VOI from the V5 region. To do this, you will return to the original GLM analysis.
3. Click Results, then select the GLM analysis SPM.mat file and the Motion contrast.
4. apply masking [No]
5. p value adjustment to control [None]
6. threshold T or p value [0.001]
7. & extent threshold voxels [3]
8. Go to point [39 -72 0]
9. Press eigenvariate
10. Name of region [V5]
11. Adjust data for [effects of interest]
12. VOI definition [sphere]
13. VOI radius(mm) [6]

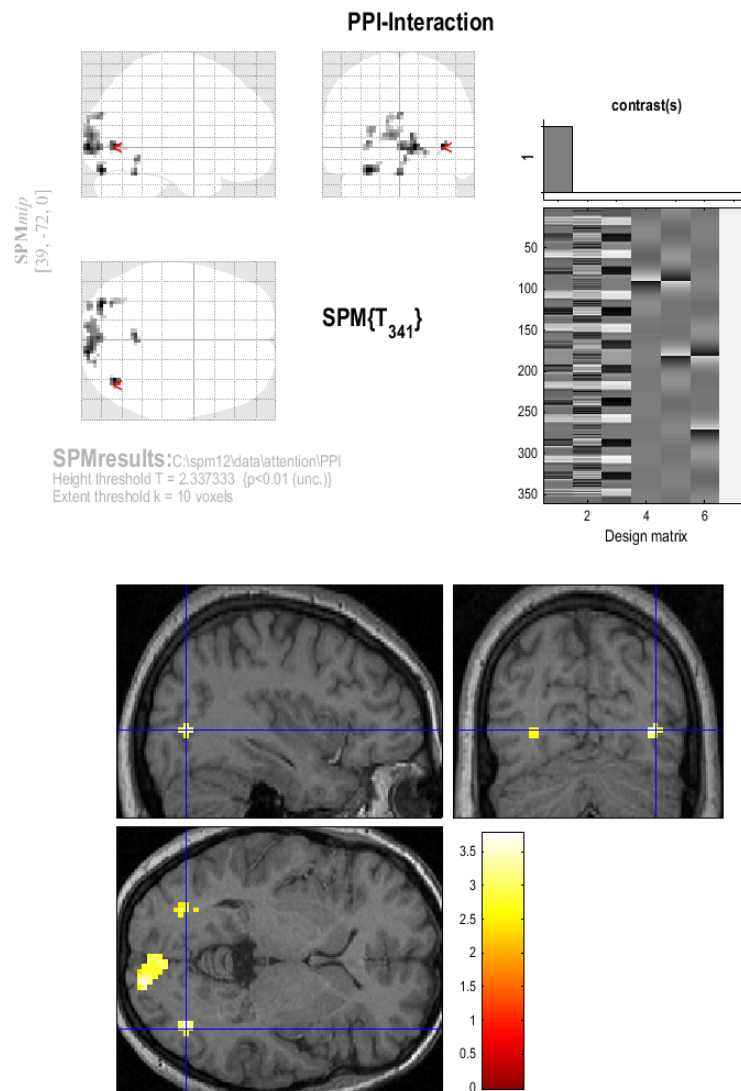


Figure 36.15: PPI results

14. Now you will create 4 PPIs (Follow the steps under section 36.5, Create PPI Variable, above). By using the PPI software machinery to create the interaction vectors, rather than just multiplying the extracted V2 and V5 eigenvariates by the behavioral vectors, the PPI vectors will be formed properly.
15. V2xNoAttention (Use the V2 VOI and include No-Attention with a contrast weight of 1, do not include Stationary, Attention)
16. V2xAttention (Use the V2 VOI and include Attention with a contrast weight of 1, do not include Stationary, No-Attention)
17. V5xNoAttention (Use the V5 VOI and include No-Attention with a contrast weight of 1, do not include Stationary, Attention)
18. V5xAttention (Use the V5 VOI and include Attention with a contrast weight of 1, do not include Stationary, No-Attention)
19. Load the PPIs you just created with the following commands at the MATLAB prompt:

```
>> v2noatt = load('PPI_V2xNoAttention');
```

```
>> v2att = load('PPI_V2xAttention.mat');
>> v5noatt = load('PPI_V5xNoAttention.mat');
>> v5att = load('PPI_V5xAttention.mat');
```

20. Plot the PPI datapoints with the following commands at the MATLAB prompt:

```
>> figure
>> plot(v2noatt.PPI.ppi,v5noatt.PPI.ppi,'k-');
>> hold on
>> plot(v2att.PPI.ppi,v5att.PPI.ppi,'r-');
```

21. To plot the best fit lines type the following first for NoAttention

```
>> x = v2noatt.PPI.ppi(:);
>> x = [x, ones(size(x))];
>> y = v5noatt.PPI.ppi(:);
>> B = x\y;
>> y1 = B(1)*x(:,1)+B(2);
>> plot(x(:,1),y1,'k-');
```

22. Then for Attention

```
>> x = v2att.PPI.ppi(:);
>> x = [x, ones(size(x))];
>> y = v5att.PPI.ppi(:);
>> B = x\y;
>> y1 = B(1)*x(:,1)+B(2);
>> plot(x(:,1),y1,'r-');
>> legend('No Attention','Attention')
>> xlabel('V2 activity')
>> ylabel('V5 response')
>> title('Psychophysilogic Interaction')
```

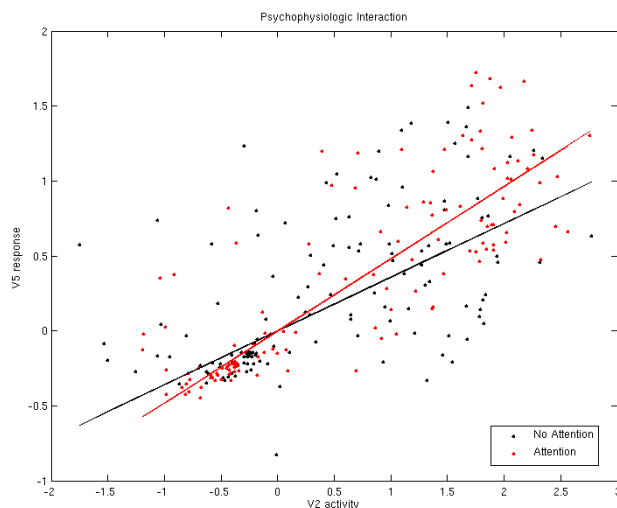


Figure 36.16: Graph demonstrating PPI interaction.



## Chapter 37

# Bayesian Model Inference

This chapter describes the use of SPM's Bayesian Model Inference capabilities. For a fuller background on this topic see [85]. We illustrate the methods using a DCM for fMRI study of the language system.

### 37.1 Background

The neuroimaging data derive from an fMRI study on the cortical dynamics of intelligible speech [67]. This study applied dynamic causal modelling of fMRI responses to investigate activity among three key multimodal regions: the left posterior and anterior superior temporal sulcus (subsequently referred to as regions P and A respectively) and pars orbitalis of the inferior frontal gyrus (region F). The aim of the study was to see how connections among regions depended on whether the auditory input was intelligible speech or time-reversed speech.

The basic DCM, from which all other models were derived, is shown in figure 37.1. Auditory input enters region P and the three areas have full intrinsic connectivity. The modulatory input, encoding whether or not the auditory stimulus was speech or reversed speech, was then allowed to modulate a subset of connections in the model. These are the forward and backward connections between P and F, and the forward and backward connections between P and A. As these are either present or absent this results in  $2^4 = 16$  different DCMs.

### 37.2 Data

An archive containing 16 DCMs for each of 12 subjects can be downloaded from the SPM web page<sup>1</sup>. This archive is called `dcm_bms.zip`. When you extract the data onto your computer a number of subdirectories will be created - one for each of the 12 subjects. The 16 DCMs for each subject are then available in these subject-specific directories. You can load one of these into SPM and examine the information contained therein.

These DCM files contain the usual information eg. the original time series from each region of interest are available in `DCM.xY(1)` for region 1, wherein `DCM.xY(1).name='PSTS_6'` indicates this is the posterior temporal region. The estimated parameter values are available in `DCM.Ep`. You should note that these DCMs were specified and estimated using SPM revision 3894 (from May 2010) and that these DCM structures differ from earlier SPM releases.

Also in the Zip archive is a file called `model_space.mat`. If you load `model_space`, you will see that it contains a data structure called `subj` with subfields 'sess' and then 'model'. If you type eg. `subj(1).sess(1).model(1)` you will see four further subfields containing information about the first DCM for subject 1. This comprises the filename (fname), the free energy approximation to the model evidence (F), posterior mean parameters (Ep), and the posterior covariance of parameters (Cp).

The use of a 'model space' file makes use of SPMs Bayesian model comparison (BMC) routines much simpler. If this file is not specified it will be automatically generated (from the DCM files)

---

<sup>1</sup>Bayesian comparison of Dynamic Causal Models: [http://www.fil.ion.ucl.ac.uk/spm/data/dcm\\_bms/](http://www.fil.ion.ucl.ac.uk/spm/data/dcm_bms/)

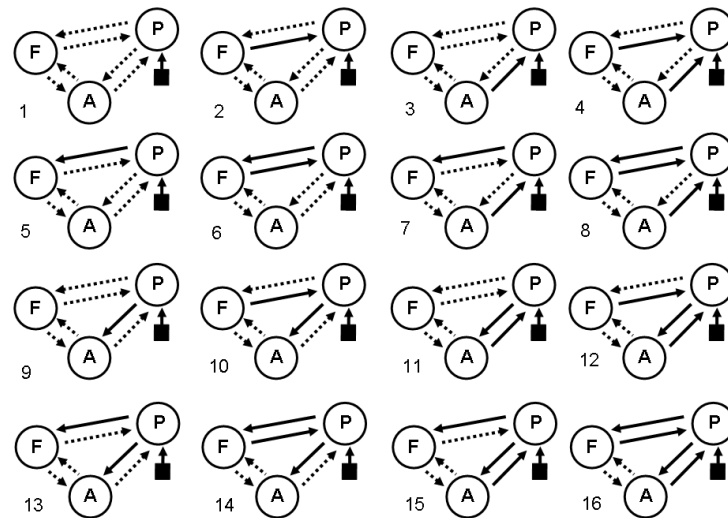


Figure 37.1: All DCMs were fully connected ie. there were endogenous connections between all three regions (dotted lines) (i) left posterior temporal sulcus (region P), (ii) left anterior superior temporal sulcus (region A) and (iii) pars orbitalis of the inferior frontal gyrus (region F). Auditory input enters region P. The sixteen models differ in their modulatory connectivity (solid lines)

the first time you use the BMC routines (see below). Alternatively, you can easily create your own model space file. To get the current file to work on your system you will need to change all of the filenames (fname) so that they correspond to the positions of the DCM files in your filespace. You can do this with the `model_space_filenames` function (also provided in the Zip archive).

### 37.3 Analysis

After unzipping the archive, correct the model space filenames using the command `subj=model_space_filenames(subj,new_base_dir)` where `new_base_dir` is the name of the directory where you have unzipped the archive. This should be something like `'C:\blah\blah\blah\dcm-base-files'`. Then save `subj` back in the model space file using `save model_space subj`.

#### 37.3.1 Single Family

Now open SPM and in the Menu window go to Batch, SPM, DCM, Bayesian Model Selection, Model Inference. This will open SPM's batch editor. Select an appropriate directory (eg. where you unzipped the archive), highlight Load model space and select the `model_space.mat` file. For inference method select 'FFX'. Save the batch job as `ffx_all_models.mat`, then press the green play button to run the job. This will produce the figure 37.2, showing that model 6 is the best model.

We can now go back and load the `ffx_all_models.mat` job in the batch editor (press the Batch button) and change the inference methods to RFX. This will produce something like the results in figure 37.3 (note that the RFX approach uses a sampling procedure with a different random initial seed on each run, so the results can vary slightly from run to run). Again, model 6 is the best model, but not by much. These RFX results will be stored in the same `BMS.mat` file as the FFX results.

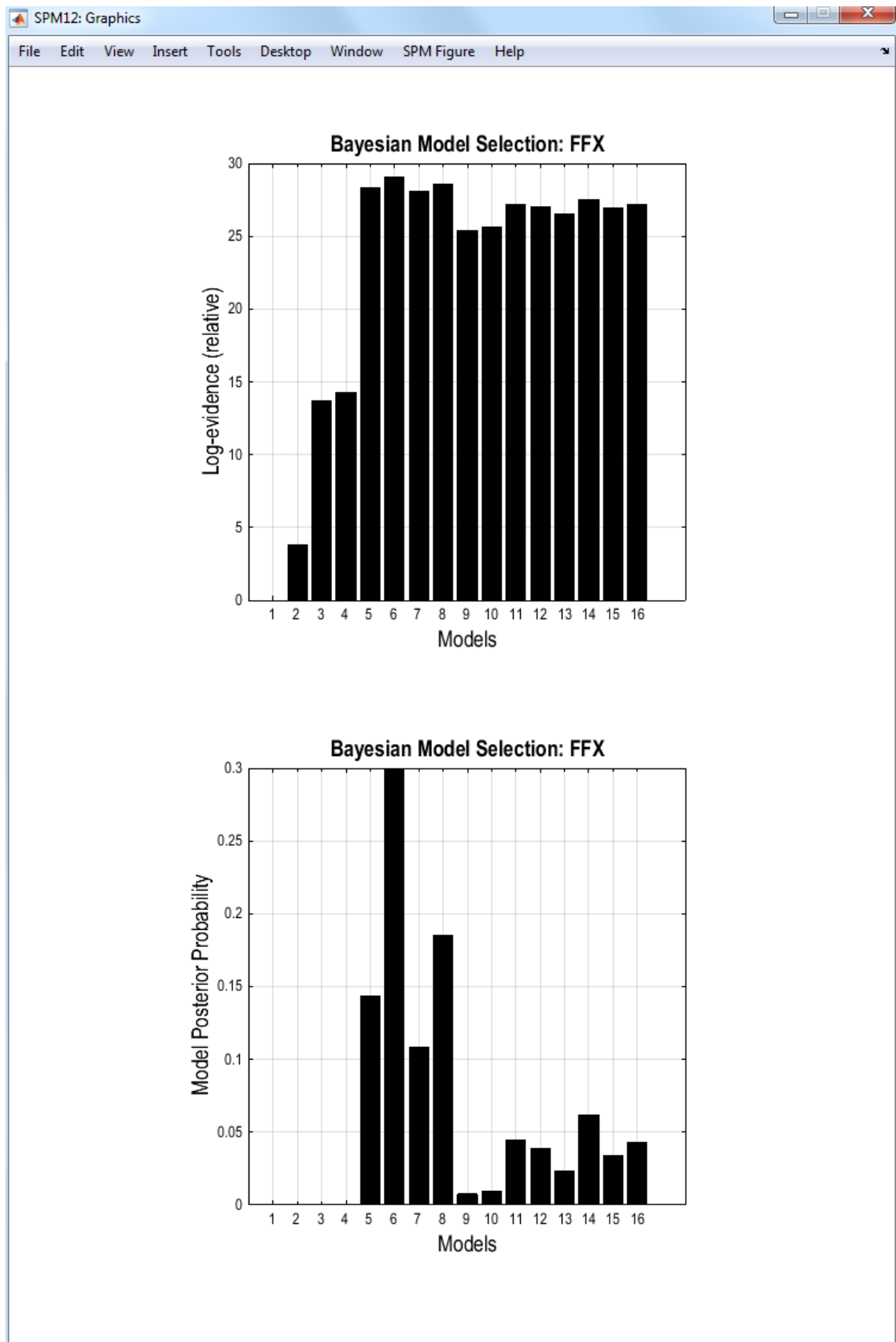


Figure 37.2: Fixed effects model inference

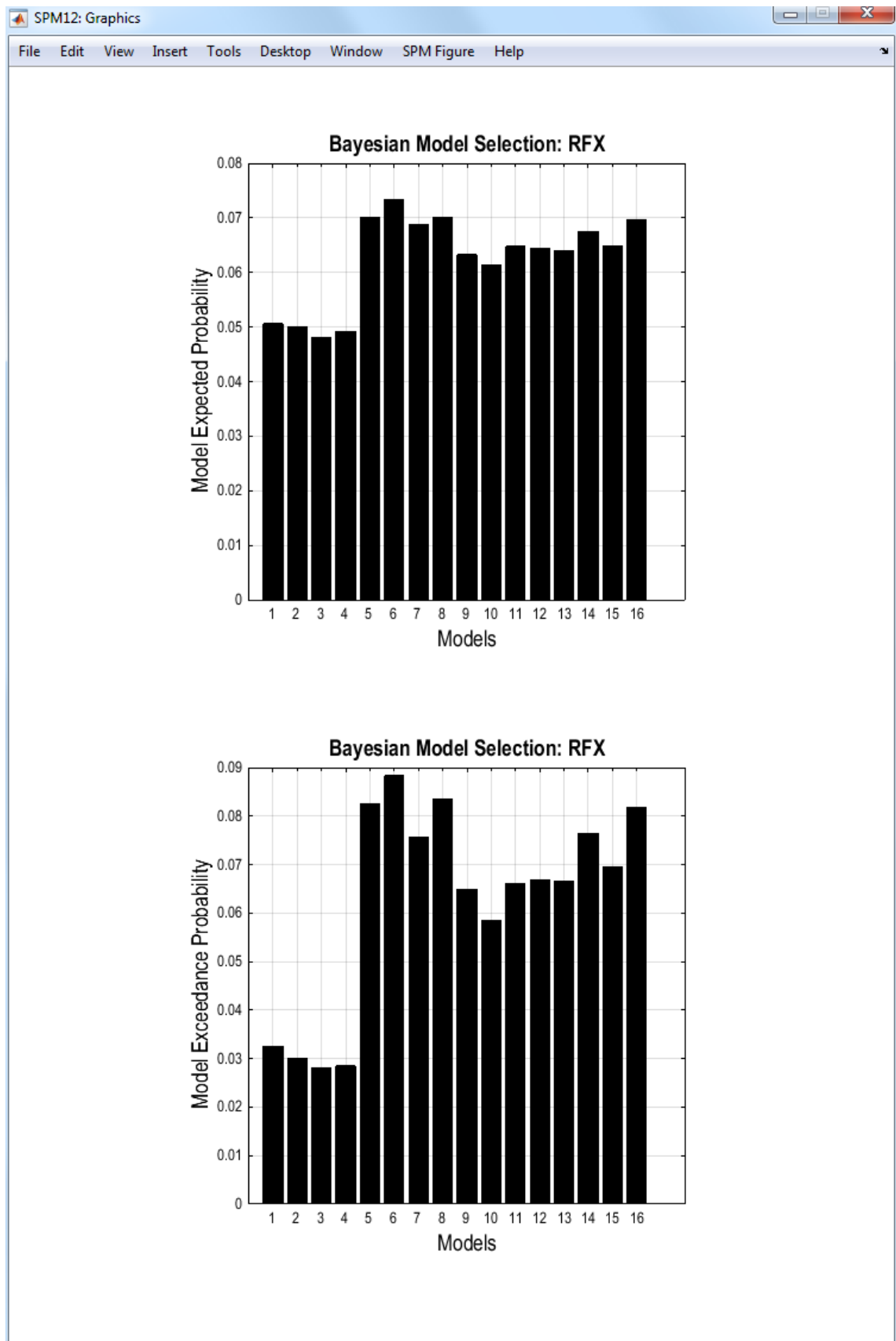


Figure 37.3: Random effects model inference

### 37.3.2 Bayesian Model Averaging

Now go back into the batch editor and reload the `ffx_all_models.mat` job. Highlight BMA, and select Choose family (instead of 'Do not compute'). Accept the 'Winning Family' option. The BMA results will be saved in the same BMS.mat file as the previous analyses. Now go ahead and press the green play button. SPM will do the FFX model inference (again), but will also implement a weighted average of the model parameters where the weights are given by the evidence for each model, as described in [85]. After the averaging is complete, SPM will report the number of models in Occams window. This should be 10 models (models 5,6,7,8,11,12,13,14,15,16).

To look at the BMA results, go to the Menu window and press the Dynamic Causal Modelling button. Then select Average, select BMA, and then the BMS.mat file just created. If you then highlight the tab (top left) to select the modulatory variables you should get the plot shown in figure 37.4.

### 37.3.3 Family level inference

The results so far have made no use of SPM's family inference procedure. Or rather, they have, but have assumed that all models belong to the same family.

Open the `ffx_all_models.mat` batch file again, highlight Family inference and select Load family. Highlight Load family and select the `pf_family.mat` file contained in the Zip archive. This comprises two families (i) those with a forward connection from P to F ('PF'), and (ii) those without it ('No PF'). Set the BMA option to Do not Compute. Select a new directory you have created for this analysis (eg pf-family) and run the job. SPM will create the family level inference plot shown in figure 37.5. This gives a 90% posterior probability to models with the P to F connection.

We will now repeat the analysis but with RFX inference. You should see a result similar to that shown in figure 37.6.

### 37.3.4 Summary Statistics and Group Analyses

The group mean DCM parameters can be easily obtained from the MATLAB command window by loading the BMS.mat file and then typing: `BMS.DCM.ffx.bma.Ep`.

The subject specific mean DCM parameters can be obtained as follows: `BMS.DCM.ffx.bma.Eps(n)`, where  $n$  is the subject number. For random-effects change `ffx` to `rfx`.

If we are interested in the modulatory connection from region 1 to region 3 (that is modulated by the second input), then the mean value of this for Subject 10 is given by `BMS.DCM.ffx.bma.Eps(10).B(3,1,2)` (which should be 0.7475). The mean connection values for all subjects (12) can be gotten with the MATLAB syntax `for i=1:12, b(i) = BMS.DCM.ffx.bma.Eps(i).B(3,1,2); end`.

These subject specific mean parameters can then act as summary statistics for a standard group analysis. For example to look for significant differences between eg. a control group and a patient group in a modulatory parameter one would implement a two-sample t-test on data from the appropriate entries in the `mean_bs` matrices. Similarly, if one has 3 groups one would use a 3-level ANOVA.

## 37.4 BMS.mat file

The BMS structure saved in BMS.mat file contains the following variables<sup>2</sup>:

**BMS.DCM.ffx/rfx** (fixed-effects (FFX) / random-effects (RFX) analysis)

<sup>2</sup> $nm$  = number of models;  $nfam$  = number of families;  $nsub$  = number of subjects;  $nsamp$  = number of samples;  $dima/b/c/d$  = dimensions of a/b/c/d DCM parameters;  $np$  = number of model parameters;  $nsses$  = number of sessions.

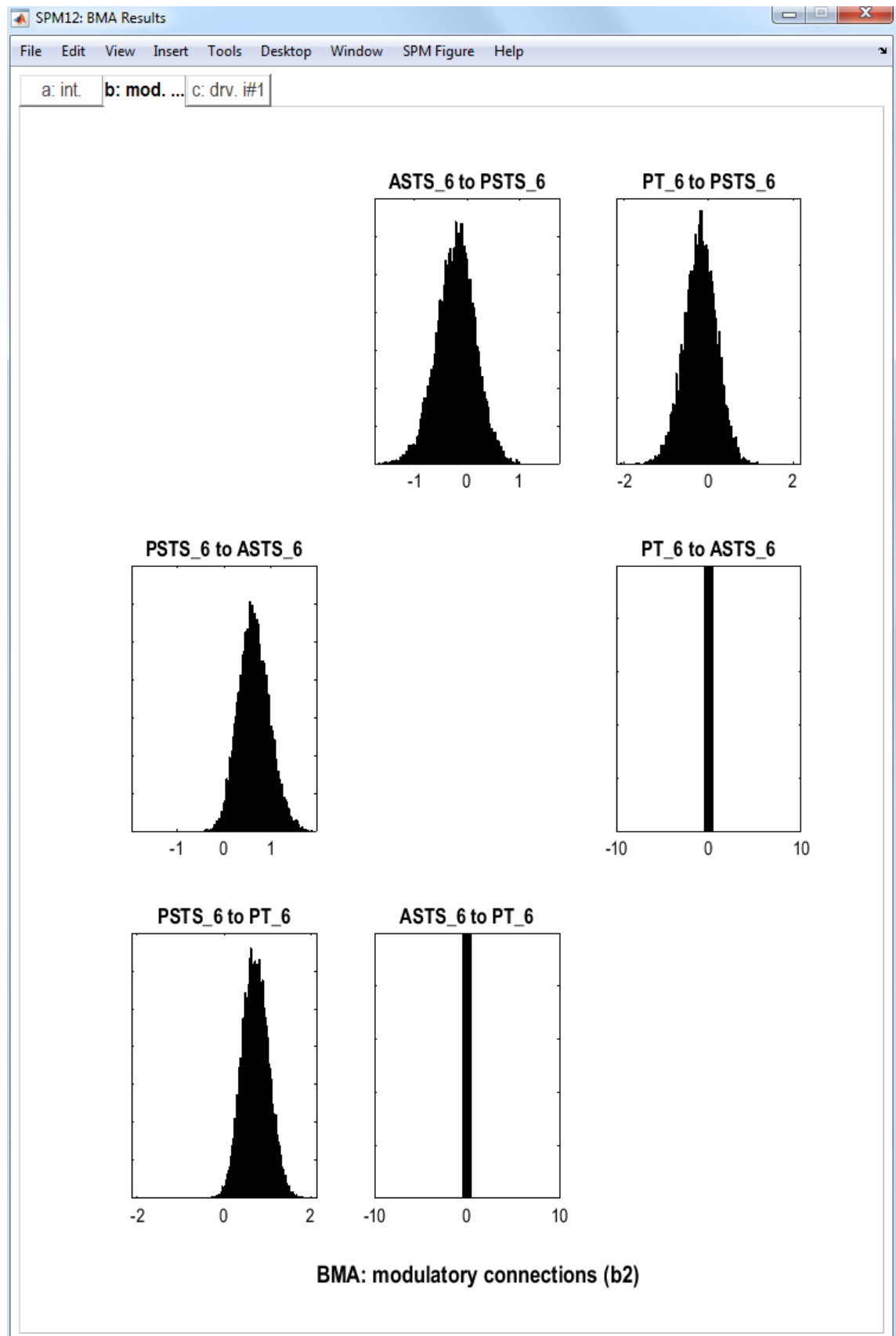


Figure 37.4: Bayesian model averaging over all 16 models

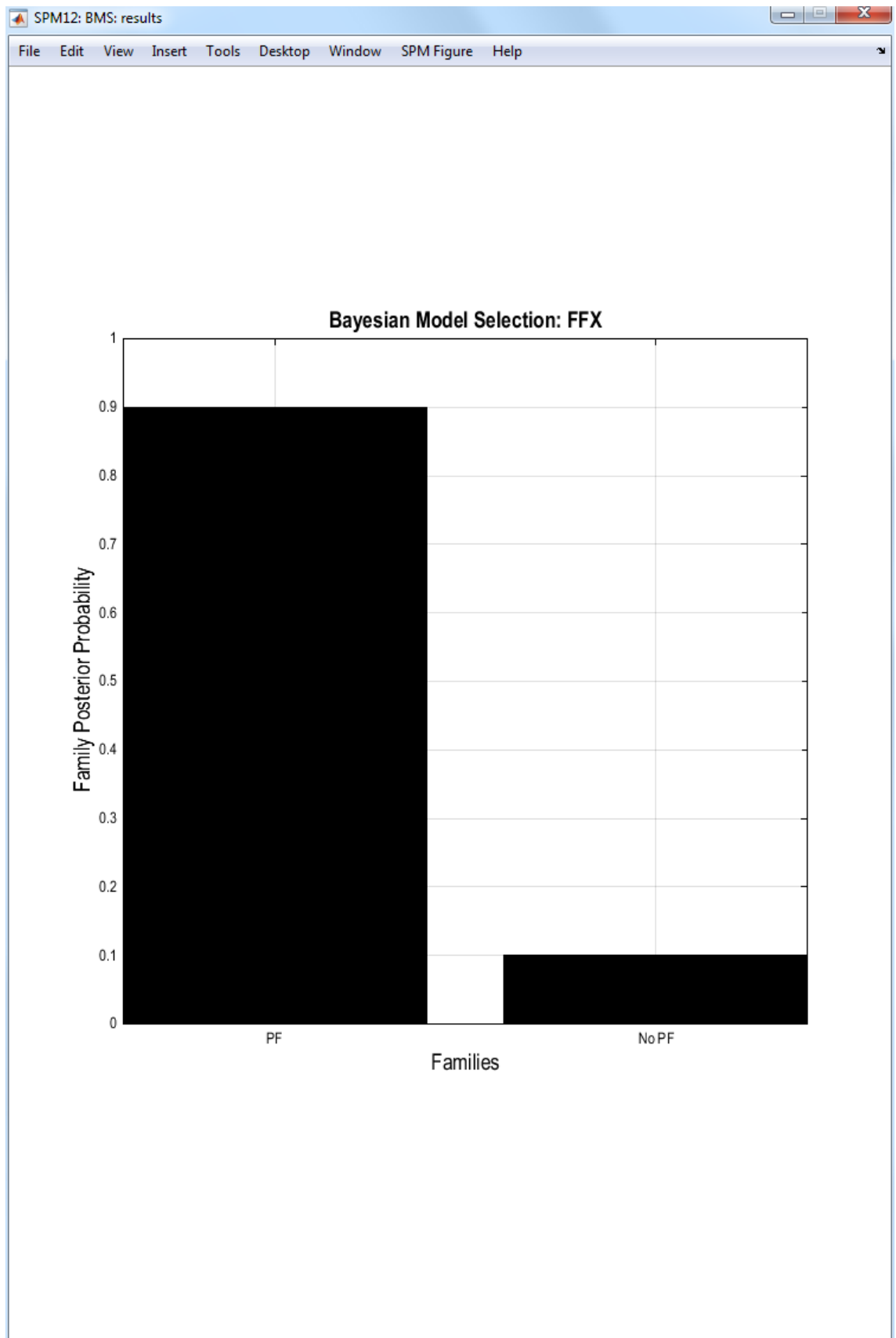


Figure 37.5: FFX family inference

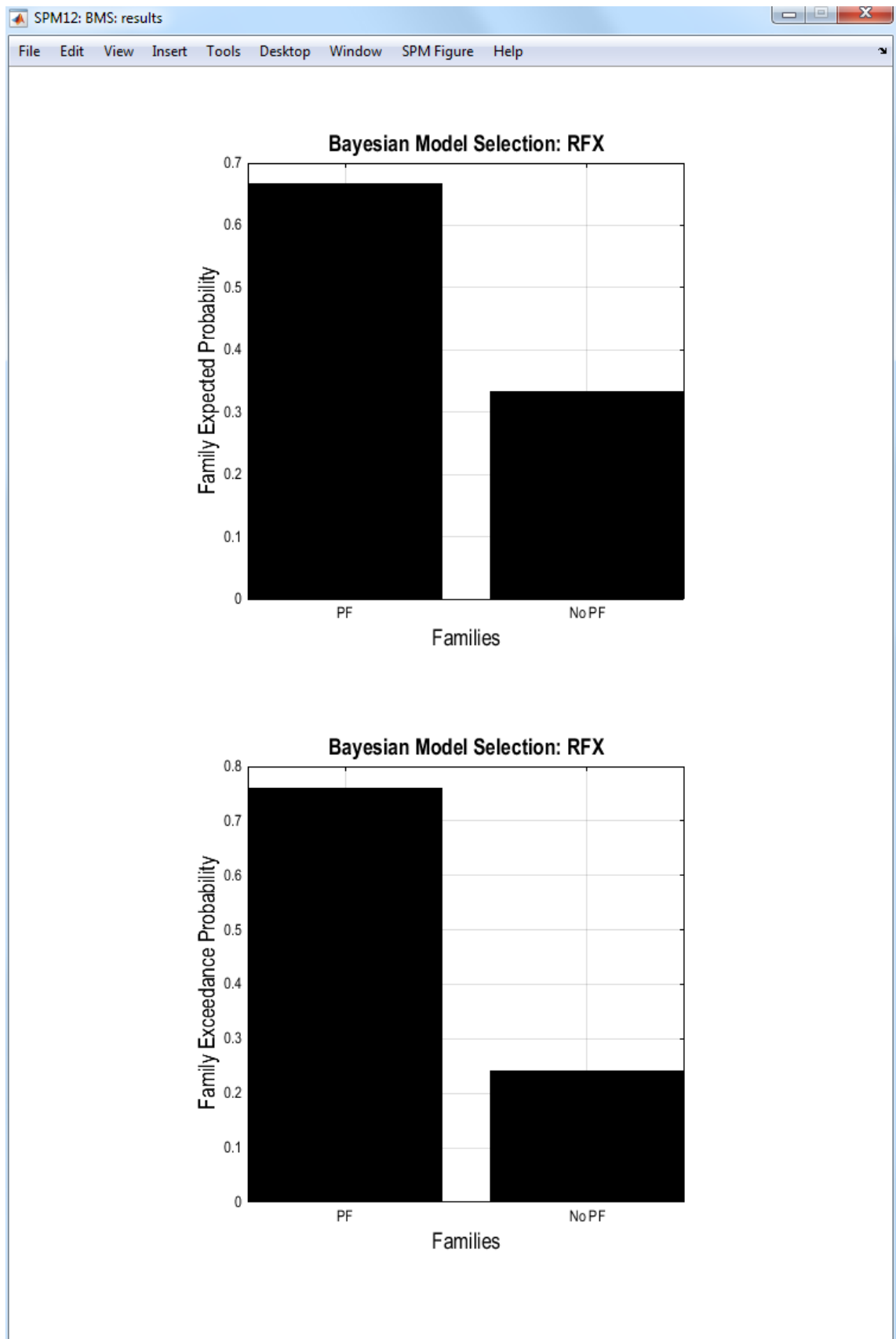


Figure 37.6: RFX family inference



<b>.data</b>	path to model_space.mat file (see below).
<b>.F_fname</b>	path to file containing the log evidence matrix, F, (if this option is specified).
<b>.F</b>	matrix of log model evidences for all subjects and models, [nsub × nm].
<b>.SF</b>	vector of summed log evidences over subjects [1 × nm].
<b>.model</b>	results from model level inference (see below).
<b>.family</b>	results from family level inference (see below).
<b>.bma</b>	results from Bayesian model averaging (see below).

### 37.4.1 Model level results

Fixed-effects:

<b>model</b>	
<b>.prior</b>	model priors, $p(m)$ , [1 × nm].
<b>.subj_lme</b>	log model evidence matrix, [nsub × nm].
<b>.like</b>	model likelihoods, $p(Y m)$ , [1 × nm].
<b>.posts</b>	model posterior probabilities, $p(m Y)$ , [1 × nm].

Random-effects (different from fixed-effects):

<b>model</b>	
<b>.alpha0</b>	initial Dirichlet parameters (prior counts), $\alpha_0$ , [1 × nm].
<b>.exp_r</b>	model posterior means, $\langle r Y \rangle$ , [1 × nm].
<b>.xp</b>	model exceedance probabilities, $\psi_m$ [1 × nm].
<b>.r_samp</b>	samples from the model posterior density, $p(r Y)$ , [nsamp × nm].
<b>.g_post</b>	posterior model probabilities for subject n and model m, $p(m_n Y)$ , [nsub × nm].

### 37.4.2 Family level results

Fixed-effects:

<b>family</b>	
<b>.names</b>	family names, ex: {'F1', 'F2', 'F3'}.
<b>.partition</b>	partition vector assigning each model to a family [1 × nm].
<b>.infer</b>	inference method ('ffx' or 'rfx').
<b>.prior</b>	family priors, $p(f_k)$ , [1 × nfam].
<b>.post</b>	family posterior probabilities, $p(f_k Y)$ , [1 × nfam].
<b>.like</b>	family likelihoods, $p(Y f_k)$ , [1 × nfam].

Random-effects (different from fixed-effects):

<b>family</b>	
<b>.Nsamp</b>	number of samples used in Gibbs $\bar{O}$ sampling (default = 20000).
<b>.prior</b>	family type of priors ('F-unity', $\alpha_0 = 1$ , for each family, is the default; other option, 'M-unity', $\alpha_0 = 1$ , for each model) .
<b>.alpha0</b>	initial values of the Dirichlet parameters (prior counts), $\alpha_{prior}(m)$ , [1 × nfam].
<b>.s_samp</b>	samples from family posterior density, $p(s Y)$ , [nsamp × nfam].
<b>.exp_r</b>	family posterior means, $\langle s_k Y \rangle$ , [1 × nfam].
<b>.xp</b>	family exceedance probabilities, $\psi_k$ , [1 × nfam].

### 37.4.3 Bayesian model averaging (BMA)

Fixed-effects:

**bma**

<b>.nsamp</b>	number of samples used to average parameters (default = 10000).
<b>.oddsr</b>	posterior odds ratio, $\pi_{OCC}$ , (number of models in Occam's window, default = 0).
<b>.Nocc</b>	number of models in Occam's window.
<b>.Mocc</b>	index of models in Occam's window, $[1 \times nm]$ .
<b>.indx</b>	index of models in Occam's window (different for each subject in RFX), $[1 \times nm]$ .
<b>.a</b>	samples from posterior density over DCM.a parameters $[dima \times nsamp]$ .
<b>.b</b>	samples from posterior density over DCM.b parameters $[dimb \times nsamp]$ .
<b>.c</b>	samples from posterior density over DCM.c parameters $[dimc \times nsamp]$ .
<b>.d</b>	samples from posterior density over DCM.d parameters $[dimd \times nsamp]$ .
<b>.mEp</b>	mean DCM parameters $[1 \times 1 \text{ struct}]$ .
<b>.sEp</b>	standard deviation of DCM parameters $[1 \times 1 \text{ struct}]$ .
<b>.mEps</b>	mean DCM parameters per subject $[1 \times nsub \text{ struct}]$ .
<b>.sEps</b>	standard deviation DCM parameters per subject $[1 \times nsub \text{ struct}]$ .

Random-effects - same variables as in fixed-effects.

### 37.5 model\_space.mat file

This structure is created automatically if it doesn't exist in the chosen directory and can be loaded for subsequent analyses as a faster option to reading the DCM.mat files. The model\_space.mat file contains the following structure:

<b>subj(nsub).sess(nsess).model(nm)</b>	
<b>.fname</b>	path to DCM.mat file.
<b>.F</b>	log-evidence (free energy).
<b>.Ep</b>	parameter estimates: conditional expectation, $[np \times 1]$ .
<b>.Cp</b>	parameter estimates: conditional covariance, $[np \times np]$ .

For a detailed description of all the variables and methods please see [85] and [91].

## Chapter 38

# Dynamic Causal Modelling for resting state fMRI

This chapter provides an extension to the framework of Dynamic Causal Modelling (DCM) for modelling intrinsic dynamics of a resting state network [36, 90]. This DCM estimates the effective connectivity among coupled populations of neurons, which subtends the observed functional connectivity in the frequency domain. We refer to this as spectral DCM (spDCM).

### 38.1 Theoretical background

Spectral DCM uses a neuronally plausible power-law model of the coupled dynamics of neuronal populations to generate complex cross spectra among measured responses. Spectral DCM is distinct from stochastic DCM (sDCM) [68] as it eschews the estimation of random fluctuations in (hidden) neural states; rendering spectral DCM essentially deterministic in nature. These models are similar to conventional deterministic DCM for fMRI [34] but model endogenous activity that would reproduce the functional connectivity (correlations) observed in resting state fMRI. DCMs for resting state data are also slightly simpler; given that most resting state designs compare groups of subjects (e.g. patient cohorts vs. controls), spDCMs do not usually require the bilinear term (accounting for condition-specific effects on effective connection strengths). In other words, spectral DCM is intended to simply compare endogenous coupling between groups of subjects (e.g. patients vs. healthy controls).

In modelling resting state activity, it is necessary to augment the ordinary differential equations used in standard DCM, with a stochastic term to model endogenous neuronal fluctuations. This renders the equations of the motion stochastic. The stochastic generative model for the resting state fMRI time series, like any other DCM, comprises of two equations: the Langevin form of evolution equation (motion) is written as:

$$\dot{z} = f(z, u, \theta) + v \quad (38.1)$$

and the observation equation, which is a static nonlinear mapping from the hidden physiological states in Eq. 38.1 to the observed BOLD activity and is written as:

$$y = h(z, u, \phi) + e \quad (38.2)$$

where  $\dot{z}$  is the rate in change of the neuronal states  $z$ ,  $\theta$  are unknown parameters (i.e. the effective connectivity) and  $v$  (resp.  $e$ ) is the stochastic process – called the state noise (resp. the measurement or observation noise) – modelling the random neuronal fluctuations that drive the resting state activity. In the observation equations,  $\phi$  are the unknown parameters of the (haemodynamic) observation function and  $u$  represents any exogenous (or experimental) inputs – that are usually absent in resting state designs. For resting state activity, Eq. 38.1 takes on a very simple linear form:

$$\dot{z} = Az + Cu + v \quad (38.3)$$

where  $A$  is the Jacobian describing the behaviour – i.e. the effective connectivity – of the system near its stationary point ( $f(z_o) = 0$ ) in the absence of the fluctuations  $v$ . It is to be noted that we can still include exogenous (or experimental) inputs,  $u$  in our model. These inputs drive the hidden states – and are usually set to zero in resting state models. It is perfectly possible to have external, (non-modulatory) stimuli, as in the case of conventional functional neuroimaging studies. For example, in [34] we used an attention to visual motion paradigm to illustrate this point.

Inverting the stochastic DCM of the form given by Eq. 38.3 in the time domain, which includes state noise, is rather complicated because such models require estimation of not only the model parameters (and any hyperparameters that parameterise the random fluctuations), but also the hidden states, which become random (probabilistic) variables. Hence the unknown quantities to be estimated under a stochastic DCM are  $\psi = \{z, \phi, \theta, \sigma\}$ , where  $\sigma$  refers to any hyperparameters (precisions or inverse covariances) defining the neuronal fluctuations. In terms of temporal characteristics, the hidden states are time-variant, whereas the model parameters (and hyperparameters) are time-invariant. There are various variational schemes in literature that can invert such models. For example, dynamic expectation maximization (DEM) [44] and generalized filtering (GF) [42].

Although the stochastic models in Eq. 38.1 and their inversion in time domain provide a useful means to estimate effective connectivity they also require us to estimate hidden states. This poses a difficult inverse problem that is computationally demanding; especially when the number of hidden states becomes large. To finesse this problem, we furnish a constrained inversion of the stochastic model by parameterising the neuronal fluctuations. This parameterisation also provides an opportunity to compare parameters encoding the neuronal fluctuations among groups. The parameterisation of endogenous fluctuations means that the states are no longer probabilistic; hence the inversion scheme is significantly simpler, requiring estimation of only the parameters (and hyperparameters) of the model.

Spectral DCM simply estimates the time-invariant parameters of their cross spectra. In other words, while stochastic DCMs model the observed BOLD timeseries of each node, spectral DCMs model the observed functional connectivity between nodes. Effectively, this is achieved by replacing the original timeseries with their second-order statistics (i.e., cross spectra), under stationarity assumptions. This means, instead of estimating time varying hidden states, we are estimating their covariance, which does not change with time. This means we need to estimate the covariance of the random fluctuations; where a scale free (power law) form for the state noise (resp. observation noise) that can be motivated from previous work on neuronal activity:

$$\begin{aligned} g_v(\omega, \theta) &= \alpha_v \omega^{-\beta_v} \\ g_e(\omega, \theta) &= \alpha_e \omega^{-\beta_e} \end{aligned} \quad (38.4)$$

Here,  $\{\alpha, \beta\} \subset \theta$  are the parameters controlling the amplitudes and exponents of the spectral density of the neural fluctuations. This models neuronal noise with a generic  $1/f^\gamma$  spectra, which characterizes fluctuations in systems that are at nonequilibrium steady-state. Using the model parameters,  $\theta \supseteq \{A, C, \alpha, \beta\}$ , we can simply generate the expected cross spectra:

$$\begin{aligned} y &= \kappa * v + e \\ \kappa &= \partial_x h \exp(t \partial_x f) \\ g_y(\omega, \theta) &= |K(\omega)|^2 g_v(\omega, \theta) + g_e(\omega, \theta) \end{aligned} \quad (38.5)$$

where  $K(\omega)$  is the Fourier transform of the system's (first order) Volterra kernels  $\kappa$ , which are a function of the Jacobian or effective connectivity. The unknown quantities  $\psi = \{\phi, \theta, \sigma\}$  of this deterministic model can now be estimated using standard Variational Laplace procedures [37]. Here  $g_y(\omega, \theta)$  represents the predicted cross spectra that can be estimated, for example, using autoregressive (AR) model. Specifically, we use a fourth-order autoregressive model to ensure smooth sample cross spectra of the sort predicted by the generative model. The frequencies usually considered for fMRI range from 1/128 Hz to 0.1 Hz in 32 evenly spaced frequency bins.

## 38.2 Practical example

Data used for this example can be downloaded from the SPM website. This dataset<sup>1</sup> consists of an exemplar subject from the full dataset available from the FC1000 project website<sup>2</sup> and was used in [90] to interrogate the information integration in default mode network (DMN) – a distinct brain system that is activated when an individual engages in introspection like mindwandering or daydreaming. The DMN comprises part of the medial prefrontal cortex (mPFC), posterior cingulate cortex (PCC) and parts of inferior parietal lobe and superior frontal regions.

The archive contains the smoothed, spatially normalised, realigned, slice-time corrected images in the directory `func`. The directory `anat` contains a spatially normalised T1 structural image and the directory `GLM` contains the file `rp_rest0000.txt` containing six head motion parameters. All preprocessing took place using SPM12.

### 38.2.1 Defining the GLM

First, we need to set up the GLM analysis and extract our time series from the results. For resting state fMRI because there is no task so first we need to generate `SPM.mat` so that we can extract the time series. This can be done by following the steps below.

Let's set up a batch that will specify the model and estimate it.

1. The analysis directory you have downloaded should include:
  - (a) A directory named `func`, which includes the preprocessed fMRI volumes.
  - (b) A directory named `anat`, which includes a normalised T1 structural volume.
  - (c) A directory named `GLM`, which include file `rp_rest0000.txt` containing the movement regressors from the realignment step.
2. In MATLAB type
 

```
>> cd GLM
>> spm fmri
```
3. From the main SPM window, click on the BATCH button.
4. From the SPM menu at the top of the Batch Editor, select “Stats > fMRI model specification”.
5. Click DIRECTORY and choose the GLM directory that you made above.
6. UNITS FOR DESIGN [SCANS]
7. INTERSCAN INTERVAL [2]
8. Click DATA & DESIGN, Choose NEW "SUBJECT/SESSION"
9. Click SCANS and choose all the functional scans `swrestxxxx.img`. There should be 175 `*.img` files.
10. From the SPM menu at the top of the Batch Editor, select “Stats > model estimation”.
11. For SELECT SPM.MAT, click on the DEPENDENCY button and choose the proposed item (the output from the previous module).
12. You should now be able to press the RUN green arrow at the top of the Batch Editor window. This will specify and estimate the GLM.

We will also need to extract signal from CSF and white matter (WM) to be used as confound. Here is a step-by-step example for extracting the WM (Pons) time series which we will use as one of the nuisance variable:

<sup>1</sup>Resting state DCM dataset: <http://www.fil.ion.ucl.ac.uk/spm/data/spDCM/>

<sup>2</sup>“1000 Functional Connectomes” Project: [http://fcon\\_1000.projects.nitrc.org/fcpClassic/FcpTable.html](http://fcon_1000.projects.nitrc.org/fcpClassic/FcpTable.html)

1. From the main SPM window, click on the BATCH button.
2. From the SPM menu at the top of the Batch Editor, select “Util > Volume of interest”
3. Select the `SPM.mat` file (generated during the previous section).
4. Adjust data: NaN
5. Which session: 1
6. Name of VOI: WM
7. Select ‘Region(s) of Interest’ > Sphere
8. Centre: [0 -24 -33]
9. VOI radius (mm): e.g.6 mm
10. Select ‘Movement of Centre’ > Fixed
11. Select ‘Region of Interest’ > Mask Image
12. Image file: select `mask.nii` (in GLM folder)
13. Expression: `i1&i2`
14. Now you should be able to press the green arrow button. This would extract the WM time series and save this as `VOI_WM_1.mat` in the working directory.

Do the same thing with to extract CSF (from one of the ventricles) signal with a sphere centred on [0 -40 -5]. This will create files `VOI_CSF_1.mat`. Next we need to adjust the SPM with the covariates. Do the following procedure:

1. From the main SPM window, click on the BATCH button.
2. From the SPM menu at the top of the BATCH EDITOR, select “Stats > fMRI model specification”.
3. Click Directory and choose the GLM directory that you made above.
4. UNITS FOR DESIGN [SCANS]
5. INTERSCAN INTERVAL [2]
6. Click DATA & DESIGN, Choose NEW "SUBJECT/SESSION"
7. Click SCANS and choose all the functional scans `swrestxxxx.img`. There should be 175 `*.img` files.
8. Click on Multiple Regressors. And then select the `VOI_CSF_1.matt`, `VOI_WM_1.mat` and `rp_rest000.txt`. Leave the rest of the fields as default.
9. From the SPM menu at the top of the Batch Editor, select “Stats > model estimation”.
10. For SELECT SPM.MAT, click on the DEPENDENCY button and choose the proposed item (the output from the previous module).
11. You should now be able to press the RUN green arrow at the top of the Batch Editor window (press ‘continue’ when asked for overwriting existing SPM.MAT file). This will specify and estimate the GLM.

### 38.2.2 Extracting time series

Once you have specified and estimated the GLM, here is now a step-by-step example for extracting the PCC time series:

1. From the main SPM window, click on the `BATCH` button.
2. From the SPM menu at the top of the Batch Editor, select “Util > Volume of interest”
3. Select the `SPM.mat` file (generated during the previous section).
4. Adjust data: `NaN`
5. Which session: `1`
6. Name of VOI: `PCC`
7. Select ‘Region(s) of Interest’ > `Sphere`
8. Centre: `[0 -52 26]`
9. VOI radius(mm): e.g. `8 mm`
10. Select ‘Movement of Centre’ > `Fixed`
11. Select ‘Region of Interest’ > `Mask Image`
12. Image file: select `mask.nii` (in GLM folder)
13. Expression: `i1&i2`
14. Now you should be able to press the green arrow button. This would extract the WM time series and save this as `VOI_CSF_1.mat` in the working directory.

SPM now computes the first principal component of the time series from all voxels included in the sphere. The result is stored (together with the original time series) in a file named `VOI_PCC_1.mat` in the working directory (the “1” refers to session 1). Do the same for the rest of the VOIs: `mPFC` (`[3 54 -2]`), `LIPC` (`[-50 -63 32]`) and `RIPC` (`[48 -69 35]`).

### 38.2.3 Specifying and estimating the DCM

Now we have extracted the time series, we are ready to build the DCM. We will look at a simplified version of the model described in [90]. In our example here, we will model a fully connected system comprising `PCC`, `mPFC` and bilateral `IPC`. This DCM is shown schematically in Figure 38.1, and can be made as follows:

1. Press the large `Dynamic Causal Modelling` button.
2. Choose `SPECIFY`.
3. Select the `SPM.mat` file you just created when specifying the GLM.
4. `DCM_???.mat`: e.g. `DMN`.
5. Select all VOIs in order `VOI_PCC_1`, `VOI_mPFC_1`, `VOI_LIPC_1` and `VOI_RIPC_1`.
6. Specify slice timings for each area. The default values are set to the last slice of the data, which was the default in the original DCM version. For sequential (as opposed to interleaved) data, this modelling option allows to use DCM in combination with any TR (slice timing differences). Here, we proceed with the default values.
7. Enter `0.04` for “Echo Time, `TE[s]`”.
8. Define the fully connected model. Your connectivity matrix should look like the one in See Figure 38.2.

A polite “Thank you” completes the model specification process. A file called `DCM_DMN.mat` will have been generated.

You can now estimate the model parameters, either by pressing the DCM button again and choosing ESTIMATE (CROSS-SPECTRA), or by typing

```
>> spm_dcm_estimate('DCM_DMN');
```

from the MATLAB command line.

Once this is completed, you can review the results as follows:

1. Press the DCM button.
2. Choose REVIEW.
3. Select `DCM_DMN.mat`

By clicking “review...” you can now select from multiple options, e.g. you can revisit the fit of the model (“Cross-spectra (BOLD)”), shown in Figure 38.3 or look at the parameter estimates for the endogenous coupling (“Coupling (A)”) as shown in Figure 38.4. Of course, you can also explore the model results at the level of the MATLAB command line by loading the model and inspecting the parameter estimates directly. These can be found in `DCM.Ep.A` (endogenous coupling) and `DCM.Ep.a` (neuronal parameters).

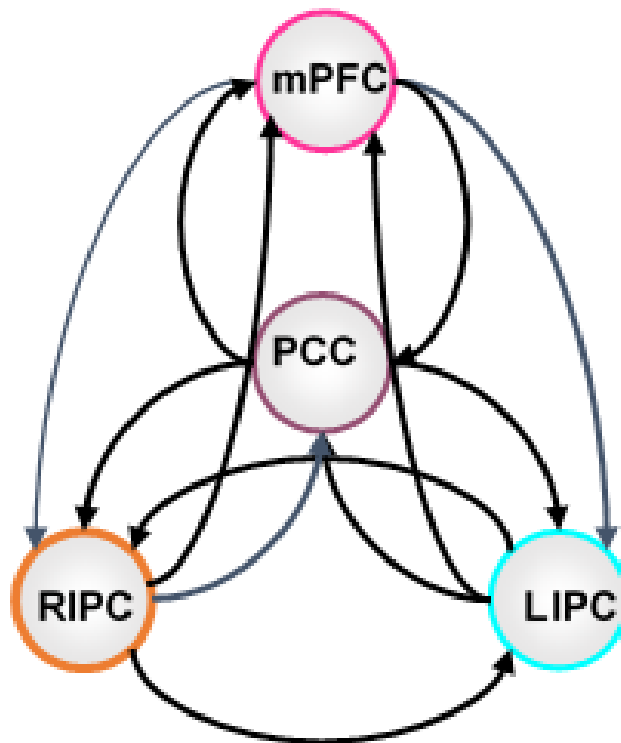


Figure 38.1: *DCM with fully connected model.*



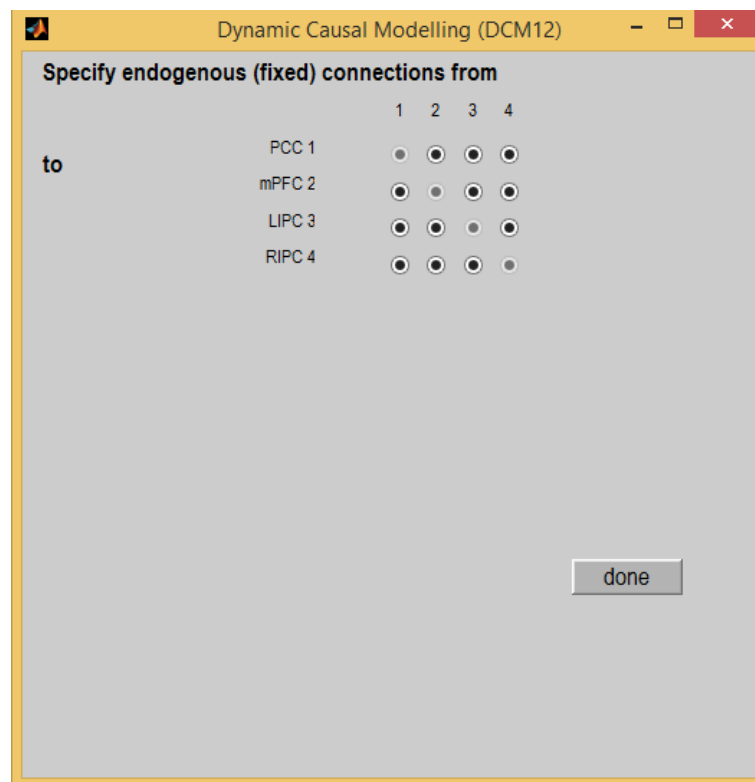


Figure 38.2: Specification of model depicted in Fig 38.2. Filled circles define the structure of the extrinsic connections  $A$  such that all circles are filled since we are using a fully connected model here.

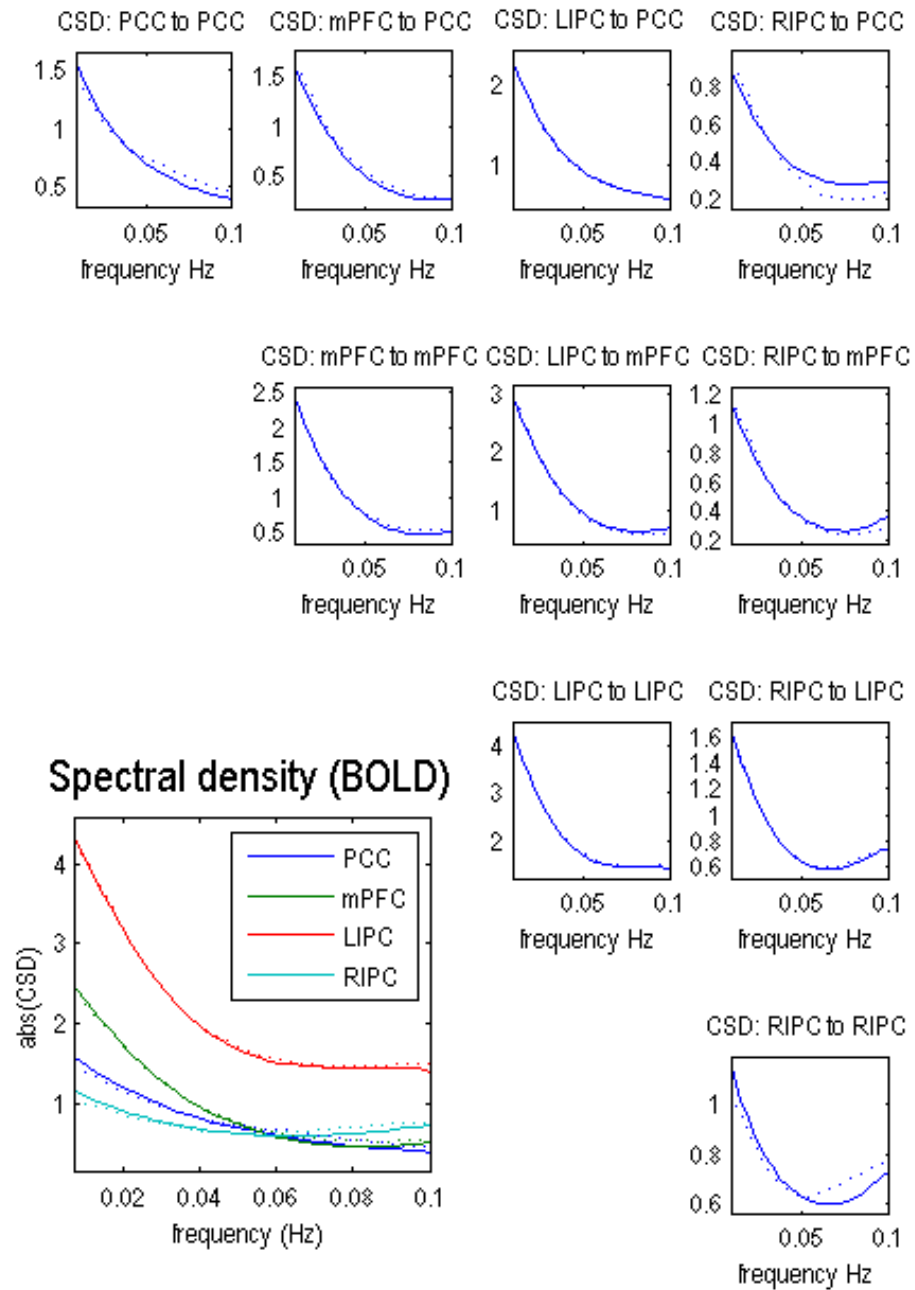


Figure 38.3: Plot of predicted and observed cross spectral densities after convergence in Fig 38.3. The dotted lines are measured cross spectra and solid lines are its predictions. The lower left panel shows the self cross spectra for the four regions. The rest of the graphics show both self and cross spectra for the four regions.

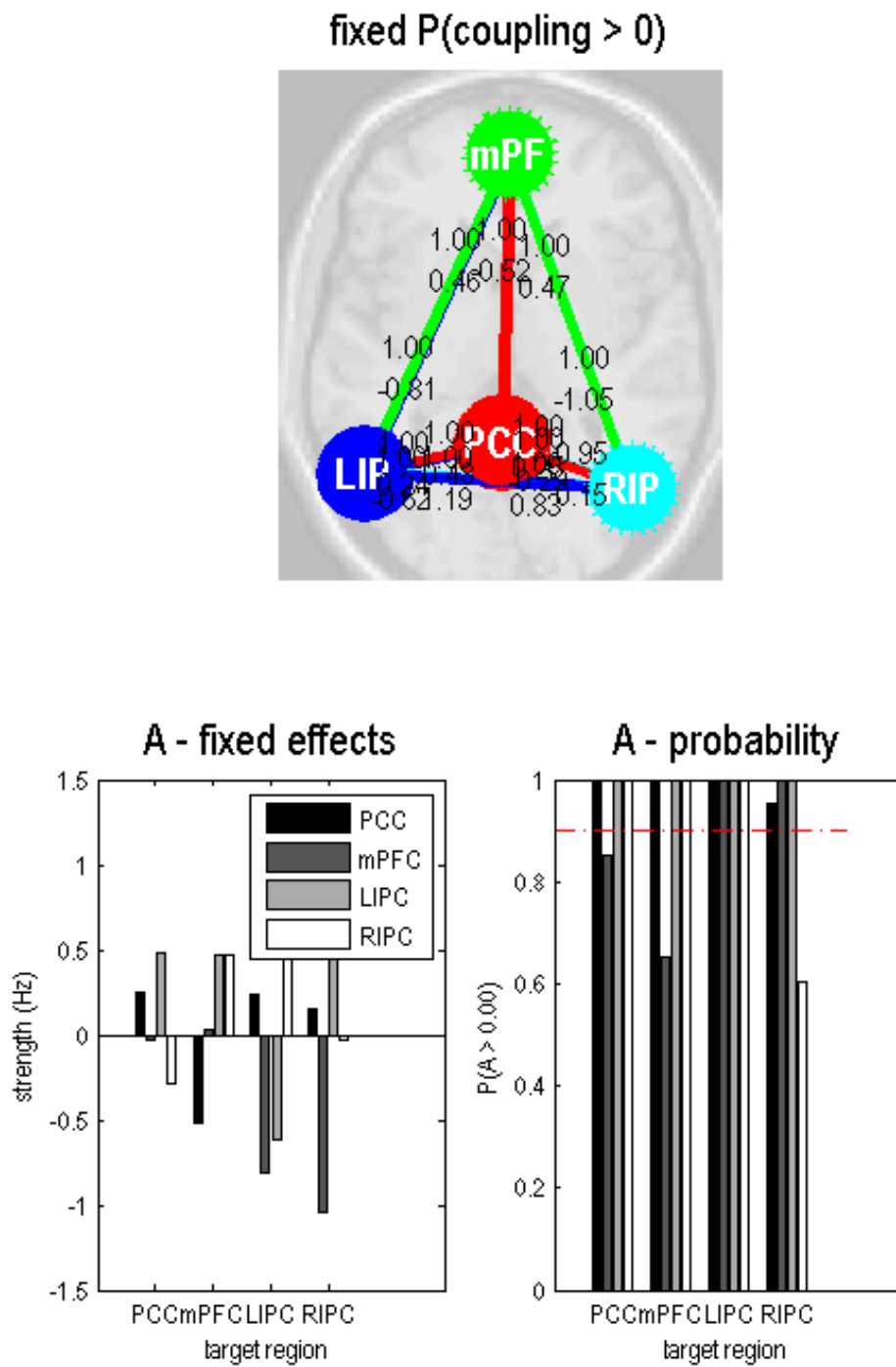


Figure 38.4: This Fig 38.4 shows the estimated fixed  $A$  matrix (top and lower left panels). The posterior probabilities of these effective connectivity parameters are shown on the lower right panel. The red dashed line depicts the 95% threshold.



## Chapter 39

# MEG source localisation

### 39.1 Overview

In this section we will generate some simulated data to show how the inversion algorithms compare when the ground-truth is known.

### 39.2 Simulation

The easiest way to simulate M/EEG data is by replacing data from an existing experimental recording in which sensor locations/head position etc are already defined. You can do this using the batch editor. Start the batch editor (**Batch** button) on main panel. Then from the dropdown menu **SPM**: select **M/EEG**; select **Source reconstruction**; select **Simulation of sources on cortex**. You will see the following menu:

You can use any SPM file you like to provide the basic simulation set up: this file will include information on sensor locations, triggers, head model. As an example we can use the preprocessed multimodal face-evoked MEG dataset<sup>1</sup>. So for M/EEG dataset select

```
cdbespm12_SPM_CTF_MEG_example_faces1_3D.mat
```

**Inversion index** allows you to determine which forward model/ inversion is used to simulate data, leave this at the default value (1) for now. **Output file prefix** allows you to specify the prefix to be added to the new file. Under 'what conditions to include', you can either specify to simulate data in all experimental conditions 'All' or in specific conditions only. Here we want to test between conditions so we will simulate data in only one condition. Select the **Conditions** option and for **Condition label** type

```
faces
```

The next option **Use inversion or define sources** allows you to either re-generate data based on a previous source reconstruction (and vary the SNR) or to set up a number of active sources on the cortical surface. We will use the last option, select **Set sources**. You can use the default options for now which defines two sources at different frequencies in approximately the auditory cortices.

That is, the two dipoles are currently set to be on (at 10 and 20Hz) during the faces condition and off during the scrambled condition.

This file has dipoles at [52, -25, 9] and [-52, -25, 9] in MNI space. The dipoles are energized at 10Hz and 20Hz from 0.1 to 0.4 seconds (Figure 39.1). In each epoch the activation profile is identical, the channel data will be slightly different due to the white noise added. The green arrow in the top left menu bar should light up when all the essential parameters have been input and you can press it to run the simulation.

You can visualise the data trial by trial if you like by using the main menu **Display/MEEG** button.

---

<sup>1</sup>Multimodal face-evoked dataset: <http://www.fil.ion.ucl.ac.uk/spm/data/mmfaces/>

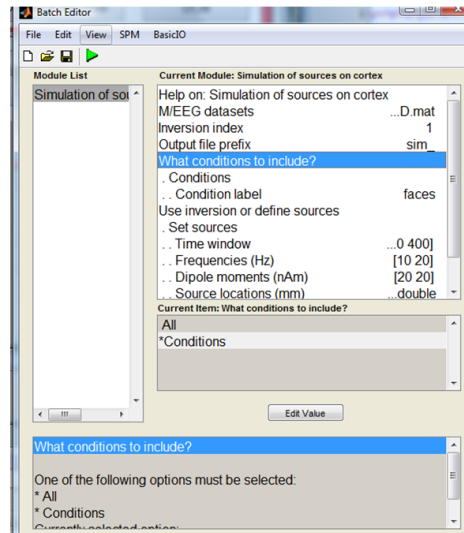


Figure 39.1: *The simulation batch options*

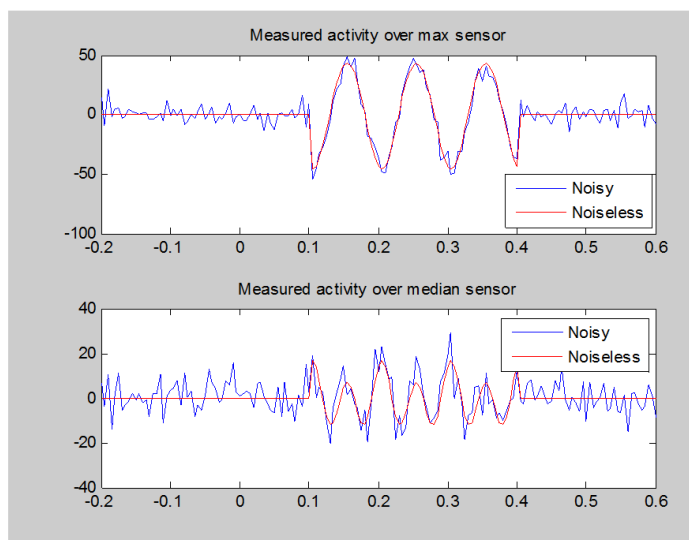


Figure 39.2: *The simulation outputs a glass brain showing maximum and median channel time-series as well as a glass brain showing the locations at which the sources were simulated*

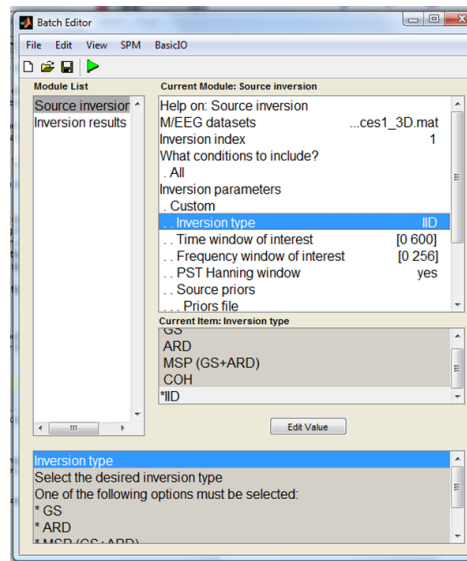


Figure 39.3: *The basic batch settings for the source inversion*

### 39.3 Imaging solutions for evoked or induced responses

There are two pathways you can take to analyse the data. Either with the GUI buttons or with the batch interface.

Firstly lets use the GUI to check that the data is ready for source inversion. On the main menu Click **3D Source Reconstruction**. Press **Load**. Select the new simulated data file `sim_cdbespm12_SPM_CTF_MEG_example_faces1_3D.mat`.

Moving left to right along the bottom panel you will notice that all of the buttons (MRI, Co-register, Forward Model) are active. This means that the preprocessing stages have already been carried out on these data (see multi-modal evoked responses chapter).

The advantage of the batch interface is that you can then build up large and automated analysis pathways, it also is a little more flexible so it has more functionality.

So restart the Batch editor from the main menu **Batch**. Then from the SPM drop-down menu select **M/EEG / source reconstruction / Source inversion**.

Select the new simulated data file `sim_cdbespm12_SPM_CTF_MEG_example_faces1_3D.mat`. Now we wish to invert all conditions using the same assumptions (and then compare between conditions afterwards) so under 'what conditions to include' select 'All'. At this point we can now try out inversion algorithms with different implicit assumptions. Under **Inversion parameters** select **Custom**. We will modify **inversion type** in the subsequent sections. Select **IID** for minimum norm assumptions for now. For the **Time window of interest** select from 0 to 600ms. For the **frequency window of interest** select 0 to 80 Hz (our data were simulated at 10 and 20Hz between 100 and 400ms). All the other settings should remain as default.

#### 39.3.1 IID (minimum norm)

We will start off with the traditional minimum norm solution: the 'IID' inversion type option . This starts by assuming that all source elements contribute something to the measured data. The constraint is that the total power (in the sources) should be minimised. Press **Invert**. Under reconstruction method press **Imaging**. For **All conditions or trials** press **Yes**. For model press **Custom**. Model inversion **IID**. Under **Time-window** "0 600". For **PST Hanning** select **Yes**. For **High-pass (Hz)** select 1 for **Low-pass (Hz)** select 48. For **Source priors**, select **No**. Under **Restrict solutions** select **No**.

We see the anticipated minimum norm result. The current density estimate is diffuse and relatively superficial due to the minimum energy constraint. Note the log-evidence 1987917 (this value depends on the data - so value of the log evidence you see may be different but it is this value

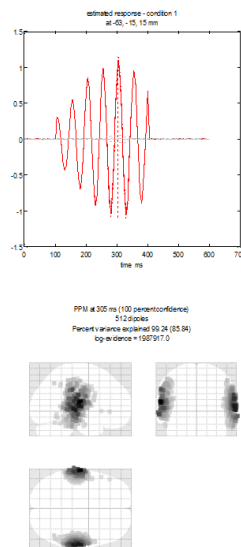


Figure 39.4: *IID imaging source reconstruction*

relative to those following which is important). The top panel shows two time-series extracted from the mesh vertex (location given at the top of the panel) with highest posterior probability. The red line corresponds to the first condition (faces). Note the sinusoidal time-series which should correspond in frequency to the source simulated on that side of the brain. The grey line corresponds to the other condition (scrambled) in which no data were simulated.

### 39.3.2 Smooth priors (COH)

The COH, under *Inversion type*, option allows the mixture of two possible source covariance matrices: the minimum norm prior above and a much smoother source covariance matrix in which adjacent sources are correlated (over the scale of a few mm). Select COH as the custom source reconstruction and run the batch again.

You will see a plot similar to Figure 39.5 appear. The lower panel shows the glass brain in which bilateral sources are apparent. The upper panel shows the time-series of the source with the largest amplitude. In this case the peak activation is identified at location 59,-15, 15mm. The 20Hz time-course (associated with this source) is also clearly visible in the top panel. Log evidence is 2000393 (again this number may be different in your spm version). Note both that the source reconstruction is more compact and that the log evidence has increased over the IID solution.

### 39.3.3 The Multiple sparse priors algorithm

In contrast to IID or COH, the greedy search routine used in MSP builds up successive combinations of source configurations until the model evidence can no longer be improved. Select GS as the inversion type and run the batch again. You will see a plot similar to Figure 39.6 appear. The lower panel shows the glass brain in which bilateral sources are apparent. The upper panel shows the time-series of the source with the largest amplitude. Again the source reconstruction is compact with log evidence is 2150944. Note both that the source reconstruction is more compact and that the log evidence has increased over the IID and COH solutions. There are two more options in the basic MSP implementation- ARD- based on the removal of patches that contribute little to the model evidence; and the use of both schemes 'ARD and GS' in which both methods provide candidate source covariance estimates which are then combined. You can try out these other options for yourself and note the model evidences (which will be directly comparable as long as the data do not change).



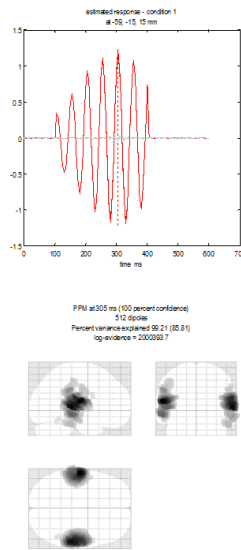


Figure 39.5: *COH imaging source reconstruction.*

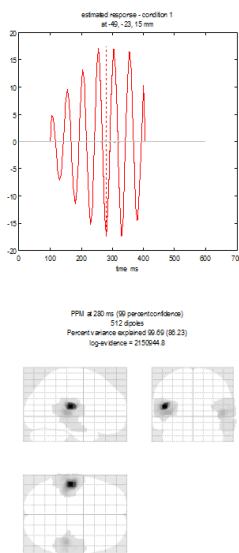


Figure 39.6: *Multiple sparse priors imaging source reconstruction using the Greedy Search (GS) option.*

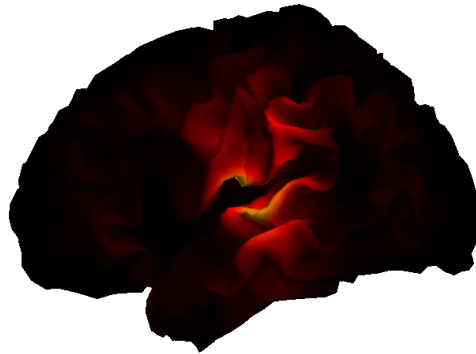


Figure 39.7: *Summary power image from IID source reconstruction on mesh.*

### 39.3.4 Making summary images

Often we will be interested in some summary of conditions over a specific time-frequency window. We can add in an extra module to the batch script to produce such an output. .

From the SPM drop down menu click **M/EEG/ Source reconstruction/ Inversion results**. Now for **M/EEG dataset**, click **Dependency-** and press OK to link the output of the previous function (the inversion) to the input of this one. We can now produce power images per condition based on a 0 to 600ms time window and a 0 to 80Hz frequency window. For **Contrast type** select **Evoked** and for output space and format select **MNI** and **Mesh** .

You should now be able to run the complete batch which re-does the inversion and outputs two surface meshes (one for each condition). You can view these meshes from the main menu : **Render/ Display**. The output image for the face condition (and the IID algorithm) is shown below.

### 39.3.5 Other MSP options

The MSP algorithm is optimized to give the simplest source distribution that explains the most data. However the library of priors (possible patch locations) must be pre-specified in advance. This could potentially cause a problem if the source of interest were not precisely centred on one of the patches in the default library. To this end Jose David Lopez (Conf Proc IEEE Eng Med Biol Soc. 2012;2012:1534-7.) has produced a version of MSP which uses multiple random patch libraries to invert the same data several times. We can make a new batch file for this. So restart the Batch editor from the main menu **Batch**. Then from the **SPM** drop-down menu select **M/EEG / source reconstruction / Source inversion, iterative**.

Select **Classic** as the custom source reconstruction algorithm- this is basically the original version of the MSP algorithm without any re-scaling factors to allow mixing of modalities or group imaging. It is advantageous in many cases as the lack of these scaling factors means that it is a true generative model of the data (and it becomes possible to test between different head positions etc). Note however that these differences in pre-processing mean that at the moment the data entering the inversion (for custom and classic options) are different and so it is not possible to compare between solutions from these two pipelines. The rest of the parameters (time, frequency windows etc) can remain as they were in the last section. The new options are the choice over the number of patches in the library and the number of iterations. You can play with these parameters to adjust the relative burden of computation time. For example- allowing just 2 patches and many iterations will make this something like a (cortically constrained) multiple

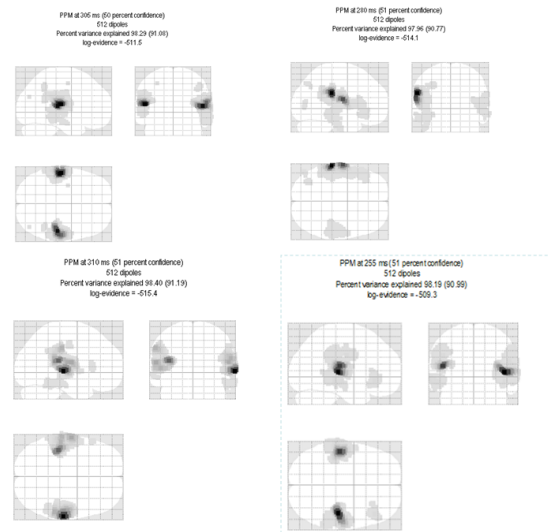


Figure 39.8: *Source inversions based on the same data but using randomly selected sets of 512 spatial priors.*

dipole (patch) fit. Alternatively, having lots of patches initially will mean the computation time is spent on pruning this set (with ARD or GS etc). You also have control over the number of temporal and spatial modes which will be used here (this makes it easier to compare between models where the lead field matrix has changed). The algorithm returns the current distribution based on the patch set with maximum free energy.

An alternative to many spatial priors is to have a single prior that is optimised using functional constraints. This idea was put forward by Belardinelli P et al. PLoS One. 2012;7(12). Here a single candidate source covariance is estimated using beamformer priors and then regularized (in much the same way as IID and COH matrices are) in the Bayesian framework. You can access this by selecting EBB (Empirical Bayes Beamformer) as the inversion type; but you should set the number of iterations here to 1 (as there is only a single prior and it will not change over repetitions).

You can see that the beamformer image is much more focal than any other image type (and it is fast to compute). However there will be many situations in which it is sub-optimal (such as if you were to simulate two correlated sources). In Belardinelli et al. the authors found that this failure was reflected in the free energy; meaning that it is still possible to directly compare this solution with GS, IID etc.

## 39.4 Dipole fitting to the average

Up until this point the analysis we have used could have been applied to either induced or evoked changes in electrical activity. The only difference being that it would not have made much sense to look at the MSPs for specific time-instants in the induced case and we would have proceeded directly to look for changes in a time-frequency window. To examine the dipole fit routine we will however concentrate on the averaged data file which will contain only evoked changes. For this final section we will revert back to the main gui. Press **Average**. Select the simulated data file and leave all of the other options as default.

Press **3D source Reconstruction**.

### 39.4.1 Load/preview the data

In the main menu click on the drop-down **Display** menu. Select **M/EEG**. For the dipole fitting we are going to use averaged MEG data, this is prefixed with an “m” in SPM. You can

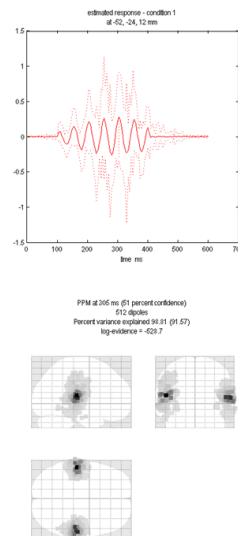


Figure 39.9: *Source inversions based on the same data but using a single beamformer prior.*

generate this file by averaging the epoched file that we have used until now. Select the file `msim_cdbespm12_SPM_CTF_MEG_example_faces1_3D.mat`.

The two sources we simulated were at 10Hz and 20Hz frequency so we can select times when only one or both of them were active. At 235ms there is only one dominant source and at 205ms both sources are clearly visible at the sensor level.

We will now move on to explore Bayesian dipole fitting to these two time instants.

### 39.4.2 Inversion

In the main menu window, select **3D Source Reconstruction**. Click **Load** and select the averaged simulated dataset above. Proceed by pressing the **Invert** button. Select the **VB-ECD** button.

#### Fitting a single dipole with no priors

At the `time_bin` or `average_win` prompt enter “235”. For `Trial type number` choose “1” (we want to model the faces data). At the `Add dipoles to model` click **Single**. For `location prior` click **Non-info**. For `Moment prior` click **Non-info**. At the `Add dipoles to 1 or stop?` prompt click **stop**. At the `Data SNR (amp)` leave as default 5. Leave the default number of iterations at “10”. You will see the 10 successive fits of the same data using a random starting location and moment. At each fit maps of the predicted and simulated data along with free-energy values and percent variance explained are shown. The final plot will be similar to Figure 39.10 where the model (i.e. dipole) which maximised the evidence (the best iteration is shown with a red dot) is displayed. Note down the model evidence (in this case  $-7.508e2$ , but the absolute value in your implementation may be different). The Bayesian dipole fit algorithm will be most useful when one has some prior knowledge of the sources (such as location, orientation or symmetry). Typical dipole fit algorithms fit 3 location parameters per dipole and then estimate the moment through a pseudo-inverse. The VB-ECD algorithm however fits 6 parameters per dipole as the moments are also allowed prior values. That is, if you have no prior knowledge then the Bayesian method will be generally less robust than such fitting methods (as more parameters are being fit). However it is when prior knowledge is supplied that the Bayesian methods become optimal.

#### Fitting a single dipole with reasonable and unreasonable priors

We will now provide some prior knowledge to the dipole fit perhaps led by the literature or a particular hypothesis. In this case we know the answer, but let us specify a location a couple of cm

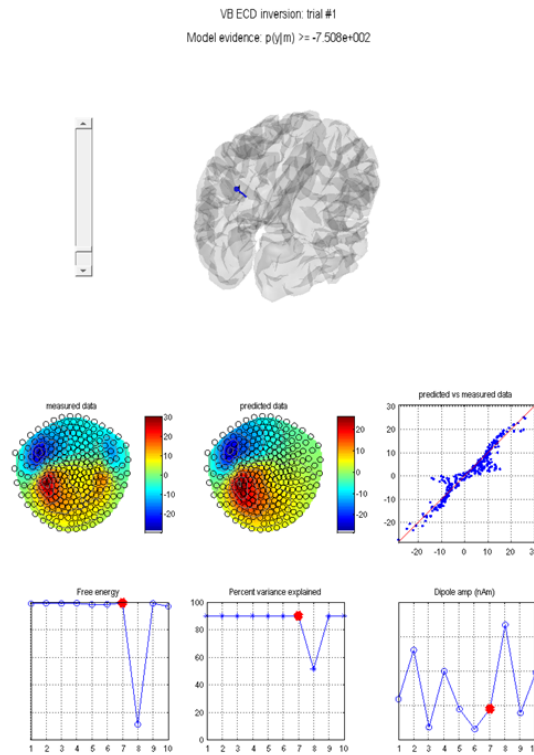


Figure 39.10: *Results of fitting a single dipole with noninformative priors.*

from where we know the source to be and try the fit again. At the `time_bin` or `average_win` prompt enter “235”. For `Trial type` number choose “1” (we want to model the faces data). At the `Add dipoles to model` click `Single`. For `location prior` click `Informative`. For the location enter “-62 -20 10”. For prior location variance leave at “100 100 100” mm<sup>2</sup>. This means that we are not sure about the source location to better than 10mm in each dimension. For `Moment prior` click `Non-info`. At the `Add dipoles to 1 or stop?` prompt click `stop`. Leave the default number of iterations at “10”. Again you will get a final fit location and model evidence (-7.455e2), which should have improved (be more positive) on the evidence above (because in this case our prior was more informative). Now go through exactly the same procedure as above but for the prior location enter “-62 +20 10”, i.e. on the wrong side of the head. You will note that the algorithm finds the correct location but the evidence for this model (with the incorrect prior) is lower (-7.476e2).

### Fitting more dipoles

We will start by examining the time instant at which we can clearly see a two-dipolar field pattern. At the `time_bin` or `average_win` prompt enter “205” (not that we are now changing the data so the subsequent evidence values will not be comparable with those at 235ms). For `Trial type` number choose “1”. At the `Add dipoles to model` click `Single`. For `location prior` click `Informative`. For the location enter “62 -20 10”. For prior location variance enter “400 400 400” mm<sup>2</sup>, that is, the prior standard deviation on the dipole location is 20mm in each direction. For `Moment prior` click `Non-info`. At the `Add dipoles to 1 or stop?` prompt click `Single`. For `location prior` click `Informative`. For the location enter “-62 -20 10”. For prior location variance enter “400 400 400” mm<sup>2</sup>. At the `Add dipoles to 1 or stop?` prompt click `stop`. Leave the default number of iterations at “10”. Note down the final model evidence (-2.548e2).

Alternatively we can exploit the fact that we have prior knowledge that the dipoles will be approximately left-right symmetric in location and orientation (this means we have fewer free parameters or a simpler model). At the `time_bin` or `average_win` prompt enter “205”. For

Trial type number choose "1". At the Add dipoles to model click Symmetric Pair. For location prior click Informative. For the205 location enter 62 -20 10. For prior location variance enter "400 400 400" mm<sup>2</sup>. For Moment prior click Non-info. At the Add dipoles to 2 or stop? prompt click stop. Leave the default number of iterations at "10". Note that the final locations are approximately correct, but importantly the model evidence (-5.235e2) is lower than previously. Given this information one would accept the (more complex) two distinct dipole model over the symmetric pair model.

## Chapter 40

# EEG Mismatch negativity data

This chapter describes the analysis of a 128-channel single subject EEG data set acquired from a study of mismatch negativity in the auditory system [45]. We thank Marta Garrido for providing us with these data. The experiment comprised an auditory oddball paradigm in which subjects heard standard (500Hz) and deviant (550Hz) tones, occurring 80% (480 trials) and 20% (120 trials) of the time, respectively, in a pseudo-random sequence subject to the constraint that two deviant tones did not occur together.

EEG data were recorded with a Biosemi<sup>1</sup> system at 128 scalp electrodes and a sampling rate of 512Hz. Vertical and horizontal eye movements were monitored using EOG electrodes. See [45] for full details of experimental stimuli and recording. To proceed with the data analysis, first download the data set from the SPM website<sup>2</sup>. The data comprises a file called `subject1.bdf` whose size is roughly 200MB. We will refer to the directory in which you have placed it as `DATA_DIR`. This chapter takes you through different stages of analysis:

- Preprocessing
- Sensor space analysis
- Source reconstruction
- Dynamic Causal Modelling

### 40.1 Preprocessing

Unlike preprocessing in SPM8 which consisted of a sequence of steps applied to a dataset using the GUI, preprocessing in SPM12 is most conveniently done by building and configuring a batch pipeline and then running it. The advantage of that is that a batch pipeline can immediately be applied to another dataset with perhaps minor adjustments. It is also possible to call preprocessing functions from more low-level scripts. These scripts can be generated from the history feature of SPM, saved in preprocessed datasets and then modified. There is also an example MATLAB script under `man\example_scripts\history_subject1.m` in the SPM distribution which repeats the preprocessing route we take here.

#### 40.1.1 Simple conversion and reviewing

At the MATLAB prompt type `spm eeg`, from the CONVERT dropdown menu select CONVERT. Select the `subject1.bdf` file and press "Done". At the prompt "Define settings?" select "just read". SPM will now read the original Biosemi format file and create an SPM compatible data file, called `spmeeg_subject1.mat` and `spmeeg_subject1.dat` in the directory containing the original data file (`DATA_DIR`). From "Display" dropdown menu select "M/EEG". In the file selection dialogue that comes up select `spmeeg_subject1.mat` and press "Done". The file will now be opened in the SPM reviewing tool. By default the tool will be opened on the 'history' tab that

<sup>1</sup>BioSemi: <http://www.biosemi.com/>

<sup>2</sup>EEG MMN dataset: [http://www.fil.ion.ucl.ac.uk/spm/data/eeg\\_mmn/](http://www.fil.ion.ucl.ac.uk/spm/data/eeg_mmn/)

shows the sequence of preprocessing steps applied to the dataset until now (in this case there is just a single line for 'Conversion'). You can switch to 'channels' and 'trials' tabs to see the list of channels and events in the dataset. Then switch to 'EEG' tab to review the continuous data. Note that due to large DC shifts in this particular dataset all the channels initially look flat and it is necessary to zoom in to actually see the EEG traces.

### 40.1.2 Preparing batch inputs

Press the 'Prepare SPM file' button in the top right corner of the reviewing tool window. A menu will appear in the SPM interactive window (the small window usually located at the bottom left). Note that on the latest version of MacOS the menu might appear in the menu bar at the top when clicking on the interactive window and not in the window itself. Similar functionality can also be accessed without the reviewing tool by choosing PREPARE from the CONVERT dropdown menu. Here we will prepare in advance some inputs that will be necessary for the subsequent preprocessing stages. The reason this needs to be done is that all inputs to the batch tool must be provided in advance before hitting the 'Run' button. Some processing steps (e.g. channel selection) are much easier to do interactively, first SPM will read some information from the dataset (e.g. list of all channels) and then ask the user to make a choice using GUI. For these steps interactive GUI tools have been added to the 'Prepare' interface under 'Batch inputs' menu. These tools save the result of the user's choice in a MAT-file that can be provided as input to batch. Usually it should be sufficient to perform the interactive steps for one dataset and the results can be used for all datasets recorded in the same way. So although preparing something in advance and only using it later may seem cumbersome at first, the idea is to facilitate batch processing of multiple datasets which is what most people routinely do. As the first step we will make a list of channels to be read. The original files contains some unused channels that do not need to be converted and we can exclude them from the beginning. From 'Batch inputs' menu select 'Channel selection'. A popup window will appear with the list of all channels in the dataset. A subset of channels can be selected (use Shift- and Ctrl- clicks if necessary to select multiple items). Select all EEG channels (these are the channels from A1 to D32). In addition select 3 channels that were used in the experiment to record EOG: EXG1, EXG2 and EXG3. Press 'OK' and save the selection as a MAT-file named e.g 'channelselection.mat'.

Next we will prepare an input for the Montage step. This step will derive vertical and horizontal EOG channels by subtracting pairs of channels located above and below the eye and below and lateral to the eye respectively. It will also change the recording reference to be the average of all EEG channels. There are several ways to create montage specifications: using script, GUI, batch and even copying and pasting from Excel. Here we will show the steps to create the montage solely using the GUI, which for this particular dataset requires several steps. First we will change the channel type of the channels containing EOG. Select 'EOG' from the 'Channel types' menu and in the channel list that comes up select EXG1, EXG2 and EXG3 channels and press OK. Next from 'Batch inputs' menu select 'Montage' and there select 'Re-reference'. In the channel list that comes up press 'Select all', press OK and save the montage e.g. as 'avref.mat'. This montage subtracts the average of all channels from each channel - this is called 'conversion to average reference'. We would now like to add to the montage two extra lines deriving the EOG channels. To do that from the same 'Montage' submenu select 'Custom montage' and in the window that appears press 'Load file' and load the previously saved 'avref.mat'. Click on the button at the top left of the montage table to select the whole table and press Ctrl-C (or equivalent on your system). Now press 'OK' , open 'Custom montage' tool again press the top left button again and press Ctrl-V (or equivalent). On some systems instead of pressing Ctrl-C and Ctrl-V it is better to select 'Copy' and 'Paste' respectively after right-clicking the corner button. Now scroll to the bottom of the table. On line 129 write 'HEOG' instead of 'EXG1'. On line 130 write 'VEOG' instead of 'EXG2'. Click to select the whole of line 131 and press 'Delete' (or equivalent). Finally scroll to the bottom right of the table and add '-1' in line 129 in the column for 'EXG2' . Also add '-1' in line 130 in the column for 'EXG3'. This defines the HEOG channel as the difference of EXG1 and EXG2 and VEOG as the difference of EXG2 and EXG3. Save the montage by pressing the 'Save as' button and naming the file 'avref\_eog.mat'. Press 'OK' to close the 'Custom montage' tool.

An alternative way for creating a montage specification is using a short script available in the



`example_scripts` folder: `montage_subject1.m`. You are welcome to try it by copying this script into `DATA_DIR` and running it. This will generate a file named `MONT_EXP.mat`.

The final input we should prepare in advance is trial definition for epoching. Select 'Trial definition' from 'Batch inputs' menu. Choose the peri-stimulus time window as -100 400. Choose 2 conditions. You can call the first condition "standard". A GUI pops up which gives you a complete list of all events in the EEG file. The standard condition had 480 trials, so select the type with value 1 and press OK. Leave 'Shift triggers' at default 0. The second condition can be called "rare". The rare stimulus was given 120 times and has value 3 in the list. Select this trial type and press OK. Leave 'Shift triggers' at default 0. Answer "no" to the question "review individual trials", and save the trial definition as `trialdef.mat`.

### 40.1.3 Preprocessing step by step

We will now build a complete batch pipeline for preprocessing and statistical analysis of the MMN dataset. We will start with running different steps one-by-one and then show how to chain these steps to run them all together.

#### Convert

As the first step we will repeat the data conversion, this time using batch. This makes it possible to refine the conversion settings and make conversion part of the complete preprocessing pipeline that can be reused later. Select `CONVERT` from the `CONVERT` dropdown menu, select the `subject1.bdf` file and this time answer 'yes' to 'Define settings?'. The conversion batch tool will appear. If you are not familiar with SPM batch, the tool presents the configuration structure as a tree where you can enter values and select different options from a list. For many options there are default settings. Options where user input must be provided are marked by the  $\leftarrow X$  sign on the right of the batch window. All these entries must be specified to enable the 'Run' button (green triangle at the toolbar) and run the batch. In our case we will provide the raw dataset name and channel selection. Click on 'Channel selection' and in the list of option appearing below the configuration tree display click on 'Delete: All(1)'. This will remove the default setting of selecting all channels. Then click on 'New: Channel file'. An entry for 'Channel file' will appear under 'Channel selection'. Double-click on it and select the previously saved `channelselection.mat` file. The 'Run' button is now enabled. Press on it to run the batch and convert the dataset.

#### Montage

Select `MONTAGE` from the `PREPROCESSING` dropdown menu. Montage batch tool will appear. Double-click on 'File name' and select the `spmeeeg_subject1.mat` file created by conversion. Click on 'Montage file name' and select the `avref_eog.mat` (or `MONT_EXP.mat`) file generated as described above. Run the batch. The progress bar appears and SPM will generate two new files `Mspmeeeg_subject1.mat` and `Mspmeeeg_subject1.dat`.

#### Prepare

The previous step also assigned default locations to the sensors, as this information is not contained in the original Biosemi \*.bdf file. It is usually the responsibility of the user to link the data to sensors which are located in a coordinate system. In our experience this is a critical step. Here we will perform this step using 'Prepare' batch. Select `PREPARE (BATCH)` from the `CONVERT` dropdown menu. Select `Mspmeeeg_subject1.mat` dataset as input. Click on 'Select task(s)' and from the options list select 'New: Load EEG sensors'. Under 'Select EEG sensors file' select `sensors.pol` file provided with the example dataset and run the batch. At this step no new files will be generated but the same dataset will be updated.

#### High-pass filter

Filtering the data in time removes unwanted frequency bands from the data. Usually, for evoked response analysis, the low frequencies are kept, while the high frequencies are assumed to carry

noise. Here, we will use a highpass filter to remove ultra-low frequencies close to DC, and a lowpass filter to remove high frequencies. We filter prior to downsampling because otherwise high-amplitude baseline shifts present in the data will generate filtering artefacts at the edges of the file. Select FILTER from the PREPROCESSING dropdown menu. Select `Mspmeeeg_subject1.mat` dataset as input. Click on 'Band' and choose 'Highpass'. Double-click on 'Cutoff(s)' and enter 0.1 as the cutoff frequency. Run the batch. The progress bar will appear and the resulting filtered data will be saved in files `fMspmeeeg_subject1.mat` and `fMspmeeeg_subject1.dat`.

### Downsample

Here, we will downsample the data in time. This is useful when the data were acquired like ours with a high sampling rate of 512 Hz. This is an unnecessarily high sampling rate for a simple evoked response analysis, and we will now decrease the sampling rate to 200 Hz, thereby reducing the file size by more than half. Select DOWNSAMPLE from the PREPROCESSING dropdown menu and select the `fMspmeeeg_subject1.mat` file. Choose a new sampling rate of 200 (Hz). The progress bar will appear and the resulting data will be saved to files `dfMspmeeeg_subject1.mat` and `dfMspmeeeg_subject1.dat`.

### Low-pass filter

Select FILTER from the PREPROCESSING dropdown menu. Select `dfMspmeeeg_subject1.mat` dataset as input. Keep the band at default 'Lowpass'. Double-click on 'Cutoff(s)' and enter 30 as the cutoff frequency. Run the batch. The progress bar will appear and the resulting filtered data will be saved in files `fdfMspmeeeg_subject1.mat` and `fdfMspmeeeg_subject1.dat`.

### Epoch

Here we will epoch the data using the previously created trial definition file. Note that it is possible to create a trial definition file based on one dataset and use it on a different dataset as long as events are coded the same way in both datasets. Select EPOCH from the PREPROCESSING dropdown menu. Select the `fdfMspmeeeg_subject1.mat` file as input. For 'How to define trials' select 'Trial definition file' and choose the previously saved 'trialdef.mat'. The progress bar will appear and the epoched data will be saved to files `efdfMspmeeeg_subject1.mat` and `efdfMspmeeeg_subject1.dat`.

### Artefacts

A number of different methods of artefact removal are implemented in SPM. Here, we will demonstrate a simple thresholding method. However, before doing so, we will look at the data in the display:

- Choose "M/EEG" from the "Display" dropdown menu.
- Select the `efdfMspmeeeg_subject1.mat` file.
- Click on the "EEG" tab.
- Press the "scalp" radio button.

The time-series for the first trial will then appear in the topographical layout shown in Figure 40.1.

You will see that Channel 14, second-row from bottom, left hemisphere, contains (slightly) higher variability data than the others. Right-click on the channel; this tells you that this channel is "A14". You will also see as an entry in this menu "bad: 0". Select this entry, and click the left button. This will make the menu disappear, but the channel now has a grey background. You have marked this channel as bad. Click on "save" in the top-right corner. This channel will then be ignored in subsequent processing. In fact this channel probably doesn't need removing, but we do so for teaching purposes only.

Select DETECT ARTEFACTS from the PREPROCESSING dropdown menu. Select the `efdfMspmeeeg_subject1.mat` file as input. Double click "How to look for artefacts" and a new branch will appear. It is possible to define several sets of channels to scan and several different methods for artefact detection. We

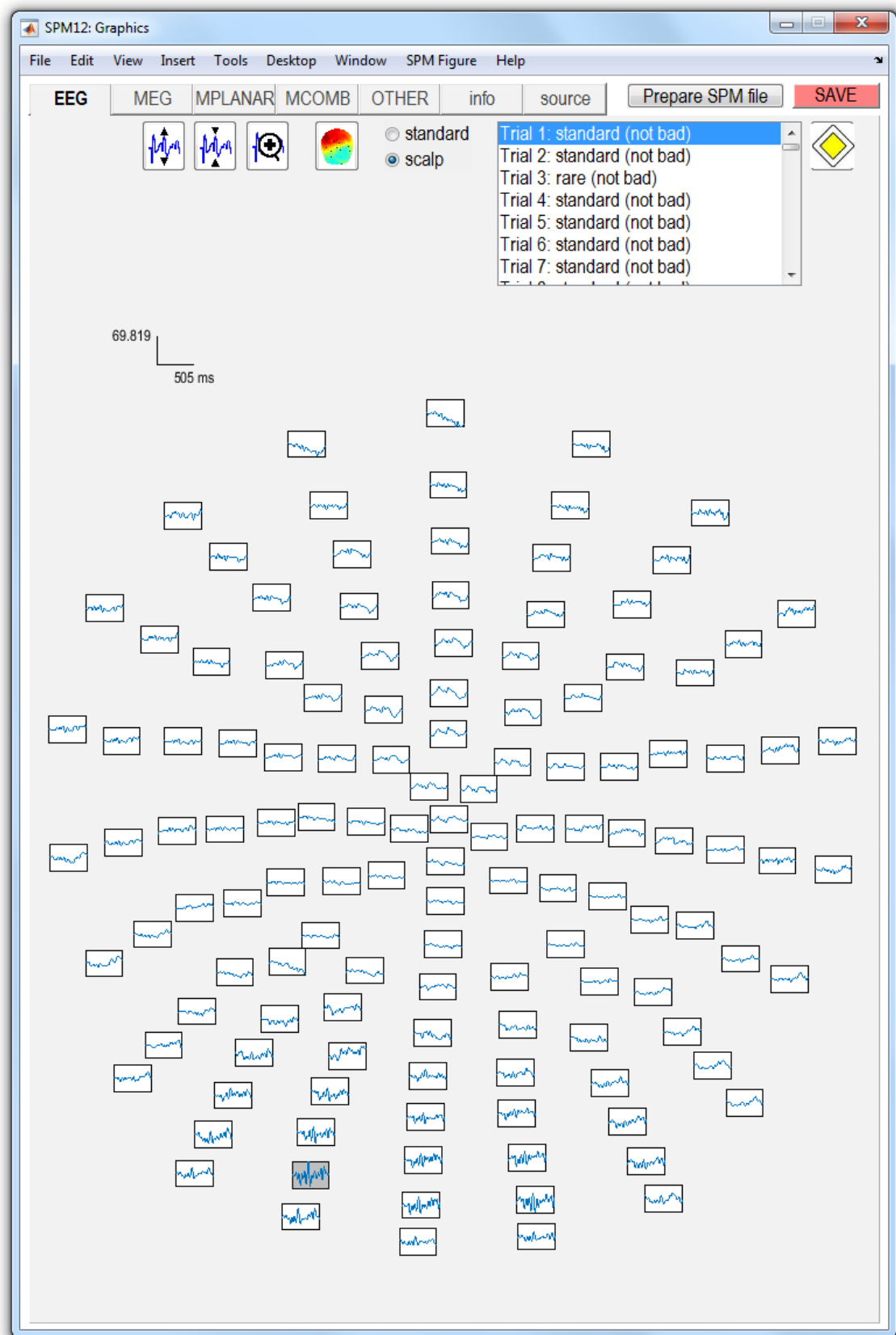


Figure 40.1: Scalp topography of single trial MMN EEG data. Channel 14, second-row from bottom, left hemisphere contains (slightly) higher variability data than the others. This channel is to be marked as artefactual (ie. 'bad').

will use simple thresholding applied to all channels. Click on “Detection algorithm” and select “Threshold channels” in the small window below. Double click on “Threshold” and enter 80 (in this case  $\mu V$ ). The batch is now fully configured. Run it by pressing the green button at the top of the batch window.

This will detect trials in which the signal recorded at any of the channels exceeds 80 microvolts (relative to pre-stimulus baseline). These trials will be marked as artefacts. Most of these artefacts occur on the VEOG channel, and reflect blinks during the critical time window. The procedure will also detect channels in which there are a large number of artefacts (which may reflect problems specific to those electrodes, allowing them to be removed from subsequent analyses).

For our dataset, the MATLAB window will show:

```
81 rejected trials: 3    4    5    7    26    27    28    [...]
1 bad channels: A14
Done      'M/EEG Artefact detection'
```

A new file will also be created, `aefdfMspmeeeg_subject1.mat`.

As an alternative to the automatic artefact rejection tool, you can also look at the interactive artefact removal routines available from `Toolbox` → `MEEG tools` → `Fieldtrip visual artifact rejection`.

### Averaging

To produce an ERP, select `AVERAGE` from the `AVERAGE` dropdown menu and select the `aefdfMspmeeeg_subject1` file as input. At this point you can perform either ordinary averaging or “robust averaging”. Robust averaging makes it possible to suppress artefacts automatically without rejecting trials or channels completely, but just the contaminated parts. For robust averaging select ‘Robust’ under ‘Averaging type’. Also select “yes” for “Compute weights by condition”<sup>3</sup>. After running the batch a new dataset will be generated `maefdfMspmeeeg_subject1.mat`. This completes the preprocessing steps.

Open the averaged dataset in the reviewing tool. To look at the ERP, click on the `EEG` tab, and press the “scalp” radio button. Now hold the `Shift` button down on the keyboard whilst selecting trial 2 with the left mouse button in the upper right corner of the graphics window. This will overlay responses to standard and rare trials on the same figure axes.

Now press the “plus” icon at the top of this graphics window and select channel `C23` (seventh central channel down from the top) with a left mouse click. This will plot the ERPs shown in Figure 40.2.

#### 40.1.4 Automatisation of preprocessing

The preprocessing steps we performed separately for instructive purposes can all be run together as part of a pipeline. Once specified such a pipeline can be reused on a different input dataset from a different session or subject. There are two ways to build a pipeline in SPM. One way is to use the batch tool as we did above but instead of configuring and running each step separately to configure all the steps as a chain and run them together. The second way is to use a script calling the low-level preprocessing functions. Such a script can be generated semi-automatically from the history of a pre-processed file. For more complicated pipelines there can be scripts that configure and run batch pipelines for some steps with some processing with the user’s own code in between. For different tools available in SPM the ‘division of labour’ between the code that is part of the batch tools and more low-level code can be different. For M/EEG preprocessing running low-level functions without the batch is quite simple whereas for statistics or source reconstruction it is much easier to use the batch code. Some examples will be provided below.

#### Building a batch pipeline

Press the ‘Batch’ button at the bottom right of the SPM menu window. An empty batch tool will be open. Processing steps can now be added via the menu. From the SPM menu in the

<sup>3</sup>In this case we do not want to pool both conditions together because the number of standard and rare trials are quite different.

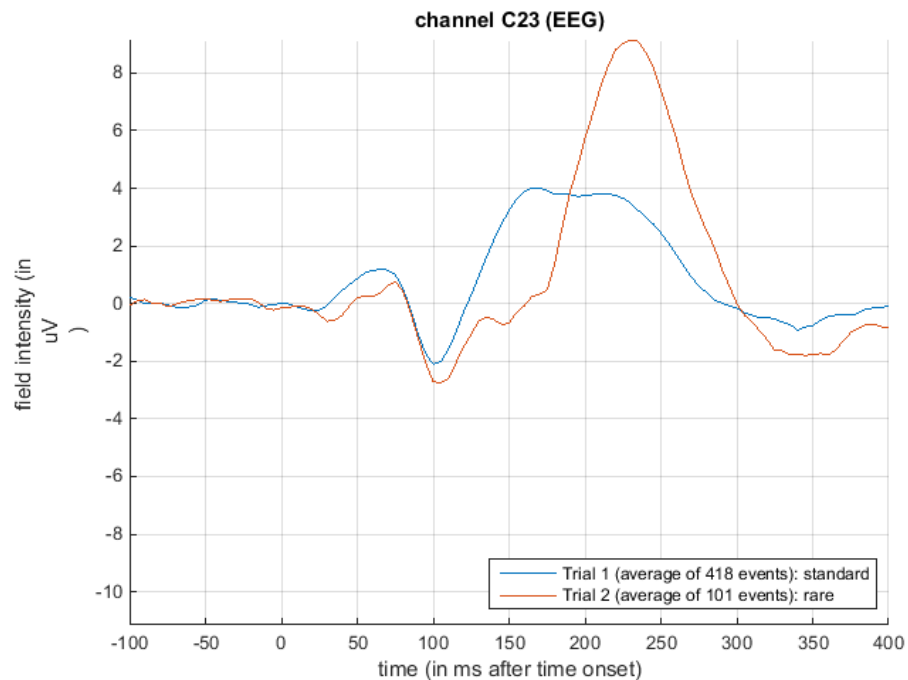


Figure 40.2: ERPs at channel C23 (CZ) for standard and rare tones. The ERP curve for rare tones lies underneath that for standard tones between 100 and 180ms. This corresponds to the mismatch negativity signal in this subject.

batch window select M/EEG submenu and in the submenu select 'Conversion'. The conversion configuration tree will appear. Configure it as described above (the 'Conversion' section). Now without running the batch or closing the batch window go back to the M/EEG submenu and select the 'Preprocessing' sub-submenu and from there 'Montage'. On the left of the batch window 'Montage' will appear in module list. Click on it to switch to the montage configuration interface. Now comes the critical difference from the previously described step-by-step processing. Single-click on 'File name'. The 'Dependency' button will appear at the bottom right part of the batch window. Press this button. A list will appear with the outputs of all the previous batch modules. In our case there is only one item in the list - the output of conversion. Select it and press 'OK'. The rest of montage configuration is as described in the 'Montage' section above. Now continue with 'Prepare' and the other modules in similar fashion. Once batch is fully configured it can be run by pressing the green triangle button.

Note that you can use "Delete" batch tool from the "Other" submenu to delete the intermediate datasets that you don't need once the final processing step has been completed. Finally, you can save the full batch pipeline as MATLAB code. To do that select "Save batch" from the "File" menu of the batch tool. At the bottom of the dialogue box that appears, select "MATLAB .m script file". Save the batch as e.g. `mmnbatch.m`. You can then open the batch .m file either in the batch tool to reproduce the configured pipeline or in the MATLAB editor. The generated MATLAB code is quite straightforward to interpret. It basically creates a data structure closely corresponding to the structure of the configuration tree in the batch tool. The structure can be easily modified e.g. by replacing one input dataset name with another. Things one should be careful about include not changing the kind of brackets around different variables, and also paying attention to whether a list is a row or column cell array of strings. Once the `matlabbatch` structure is created by the code in a batch m-file (or loaded from a batch mat-file) it can immediately be run from a script without the need to load it in a batch GUI. Assuming that the batch structure is called `matlabbatch`, the command for running it is `spm_jobman('run', matlabbatch)`.

### Generating scripts from history

The evoked response file (and every other SPM MEEG data file) contains a history-entry which stores all of the above preprocessing steps. You can take this history and produce a script that will re-run the same analysis which you entered using the GUI. See the “history” tab in the “info” section when displaying the data. Chapter 12 provides more details on this.

## 40.2 Sensor space analysis

A useful feature of SPM is the ability to use Random Field Theory to correct for multiple statistical comparisons across N-dimensional spaces. For example, a 2D space representing the scalp data can be constructed by flattening the sensor locations and interpolating between them to create an image of  $M \times M$  pixels (when  $M$  is user-specified, eg  $M=32$ ). This would allow one to identify locations where, for example, the ERP amplitude in two conditions at a given timepoint differed reliably across subjects, having corrected for the multiple t-tests performed across pixels. That correction uses Random Field Theory, which takes into account the spatial correlation across pixels (i.e. that the tests are not independent). Here, we will consider a 3D example, where the third dimension is time, and test across trials within this single subject. We first create a 3D image for each trial of the two types, with dimensions  $M \times M \times S$ , where  $S=101$  is the number of samples (time points). We then take these images into an unpaired t-test across trials (in a 2nd-level model) to compare “standard” and “rare” events. We can then use classical SPM to identify locations in space and time in which a reliable difference occurs, correcting across the multiple comparisons entailed. This would be appropriate if, for example, we had no a priori knowledge where or when the difference between standard and rare trials would emerge. The appropriate images are created as follows.

Select ‘Convert to images’ from the ‘Images’ dropdown menu. In the batch tool that will appear select the `aefdfmspmeeg_subject1.mat` as input. For the ‘Mode’ option select ‘scalp x time’. In the ‘Channel selection’ option delete the default choice (‘All’) and choose ‘Select channels by type’ with ‘EEG’ as the type selection. You can now run the batch.

SPM will take some time as it writes out a NIfTI image for each condition in a new directory called `aefdfmspmeeg_subject1`. In our case there will be two files, called `condition_rare` and `condition_standard`. These are 4D files, meaning that each file contains multiple 3D scalp x time images, corresponding to non-rejected trials. You can press “Display: images” to view one of these images. Change the number in the ‘Frames’ box to select a particular trial (first trial is the default). The image will have dimensions  $32 \times 32 \times 101$ .

To perform statistical inference on these images:

- Create a new directory, eg. `mkdir XYTstats`.
- Press the “Specify 2nd level” button.
- Select “two-sample t-test” (unpaired t-test)
- Define the images for “Group 1” as all those in the file `condition_standard`. To do that write ‘standard’ in the ‘Filter’ box and ‘Inf’ in the ‘Frames’ box of the file selector. All the frames will be shown. Right click on any of the frames in the list and choose ‘Select all’. Similarly for “Group 2” select the images from `condition_rare` file.
- Finally, specify the new `XYTstats` directory as the output directory.
- Press the “save” icon, top left, and save this design specification as `mmn_design.mat` and press “save”.
- Press the green “Run” button to execute the job<sup>4</sup> This will produce the design matrix for a two-sample t-test.
- Now press “Estimate” in SPMs main window, and select the `SPM.mat` file from the `XYTstats` directory.

<sup>4</sup>Note that we can use the default “nonsphericity” selections, i.e. that the two trial-types may have different variances, but are uncorrelated.

Now press “Results” and define a new F-contrast as [1 -1] (for help with these basic SPM functions, see eg. chapter 30). Keep the default contrast options, but threshold at  $p < .05$  FWE corrected for the whole search volume and select “Scalp-Time” for the “Data Type”. Then press “whole brain”, and the Graphics window should now look like that in Figure 40.3. This reveals a large fronto-central region within the 2D sensor space and within the time epoch in which standard and rare trials differ reliably, having corrected for multiple F-tests across pixels/time. An F-test is used because the sign of the difference reflects the polarity of the ERP difference, which is not of primary interest.

The cursor in Figure 40.3 has been positioned by selecting the second cluster in the results table. This occurs at time point 160ms post stimulus.

Now:

- Press the right mouse button in the MIP
- Select “display/hide channels”
- Select the `maefdfMspmeeeg_subject1.mat` file.

This links the `SPM.mat` file with the M/EEG file from which the EEG images were created. It is now possible to superimpose the channel labels onto the spatial SPM, and also to “goto the nearest channel” (using options provided after a right mouse click, when navigating the MIP).

We have demonstrated sensor space analysis for single-subject data. More frequently, one would compute ERP images for each subject, smooth them, and then perform paired t-tests over subjects to look for condition differences. See [45] for a group analysis of MMN data.

Finally, if one had more constrained a priori knowledge about where and when the differences would appear, one could perform a Small Volume Correction (SVC) based on, for example, a box around fronto-central channels and between 100 and 200ms poststimulus. We also refer the reader to chapter 13 for further details on sensor space analysis.

### 40.2.1 Batching statistics

The above described steps can be automatized using the batch tool. The relevant modules can be found in the ‘Stats’ submenu of the ‘SPM’ menu. They are ‘Factorial design specification’, ‘Model estimation’, ‘Contrast manager’ and ‘Results report’. We will not go over the batch steps in detail but you should be able to build the batch now based on previously described principles. One point worth mentioning is that preprocessing and statistics can be done in a single batch with dependencies. For that the ‘Convert2Images’ module can be added to the batch twice and the ‘Conditions’ option in this module can be used to convert the ‘standard’ and ‘rare’ conditions separately in the two instances of the module to match the two separate dependencies from ‘Factorial design specification’.

## 40.3 Source reconstruction

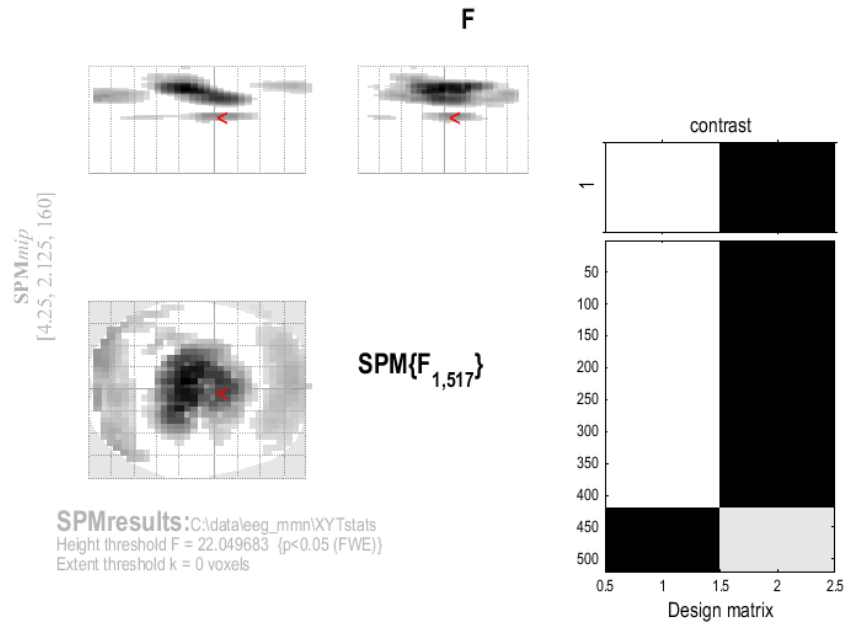
Source reconstruction comprises forward modeling and inverse modeling steps and is implemented by pressing the 3D source reconstruction button in SPM’s top-left window. This brings up the source localisation GUI shown in Figure 40.4. The following subsections detail each of the steps in a source reconstruction analysis. We also advise the reader to consult the reference material in chapter 14.

### 40.3.1 Mesh

The first step is to load the data and create a cortical mesh upon which M/EEG data will be projected:

- Press the “Load” button in the source localisation GUI and select the file `maefdfMspmeeeg_subject1.mat`.
- Enter “Standard” under “Comment/Label for this analysis” and press OK.
- Now press the “template” button.





Statistics: p-values adjusted for search volume

set-level		cluster-level				peak-level					mm mm ms	
p	c	p <sub>FWE-corr</sub>	q <sub>FDR-corr</sub>	k <sub>E</sub>	p <sub>uncorr</sub>	p <sub>FWE-corr</sub>	q <sub>FDR-corr</sub>	F	(Z <sub>≡</sub> )	p <sub>uncorr</sub>		
0.000	8	0.000	0.000	3157	0.000	0.000	0.000	120.59	Inf	0.000	-4	-25 285
						0.000	0.000	111.89	Inf	0.000	4	-25 300
						0.000	0.000	110.09	Inf	0.000	-13	-19 280
		0.000	0.000	492	0.000	0.000	0.000	60.62	7.48	0.000	4	2 160
						0.000	0.000	54.81	7.12	0.000	0	13 160
						0.001	0.014	32.37	5.48	0.000	13	-19 160
		0.000	0.000	906	0.000	0.000	0.000	54.27	7.09	0.000	-34	-89 255
						0.000	0.000	53.28	7.02	0.000	-21	-95 250
						0.000	0.000	50.27	6.82	0.000	9	-95 240
		0.000	0.000	1018	0.000	0.000	0.000	43.26	6.34	0.000	-4	61 310
						0.000	0.002	36.94	5.86	0.000	-26	56 305
						0.000	0.003	35.88	5.77	0.000	26	45 310
		0.001	0.018	43	0.011	0.000	0.013	32.75	5.51	0.000	-55	-52 155
						0.001	0.016	32.07	5.45	0.000	-51	-62 155
						0.009	0.199	26.14	4.91	0.000	-60	-36 160
		0.007	0.194	12	0.146	0.009	0.208	25.95	4.89	0.000	-55	40 300
						0.039	0.811	22.60	4.56	0.000	-60	29 300
		0.020	0.453	4	0.396	0.016	0.337	24.69	4.77	0.000	-26	-25 395
		0.035	0.692	1	0.692	0.050	0.990	22.07	4.50	0.000	30	-19 150

table shows 3 local maxima more than 8.0mm apart

Height threshold: F = 22.05, p = 0.000 (0.050)  
 Extent threshold: k = 0 voxels  
 Expected voxels per cluster, <k> = 5.963  
 Expected number of clusters, <c> = 0.05  
 FWEp: 22.050, FDRp: 31.630, FWEc: 1, FDRc: 43

Degrees of freedom = [1.0, 517.0]  
 FWHM = 23.3 28.9 30.5 mm mm ms; 5.5 5.4 6.1 (voxels)  
 Volume: 9528813 = 83426 voxels = 424.5 resels  
 Voxel size: 4.3 5.4 5.0 mm mm ms; (resel = 179.72 voxels)

Figure 40.3: In this SPM the time axis is reflected in the two MIP windows in the top row, with time proceeding from the bottom to the top of the page. The cursor has been positioned by selecting the third cluster in the results table. This occurs at time point 160ms post stimulus. The design matrix on the right hand side comprises two columns, the first for standard trials and the second for rare ones.



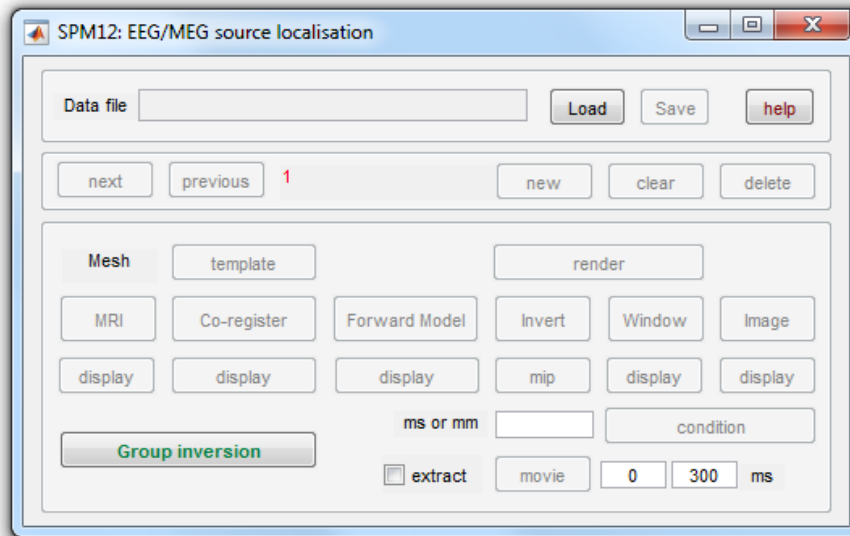


Figure 40.4: Graphical user interface for 3D source localisation. A complete localisation comprises the following steps (i) creation of a cortical mesh, (ii) co-registration of the mesh with M/EEG data, (iii) creation of a forward model, and (iv) results interrogation. As each of these steps is completed the relevant part of the GUI becomes highlighted (text appears more solid).

- For “Cortical mesh”, select “normal”.

SPM will then form the “standard” or “canonical” cortical mesh shown in the Graphics window which, after rotation, should look like Figure 40.5

### 40.3.2 Coregister

Now press the “Co-register” button. This will create further output in the Graphics window, the upper panel of which should look like Figure 40.6.

In this coregister step we were not required to enter any further parameters. However, if you are not using the template (or “canonical” mesh) or if at the “prepare” stage above you loaded your own (non-standard) sensor positions then you will be asked for the locations in MNI coordinates of the fiducial positions.

### 40.3.3 Forward model

Now press the “Forward model” button. Then select “EEG-BEM” in response to the “Which EEG head model?” question. SPM will then use a Boundary Element Method (BEM) which will take a few minutes to run. Upon completion SPM will write the `single_subj_T1_EEG_BEM.mat` file into the canonical subdirectory of your SPM distribution. The Graphics window should now appear as in Figure 40.7. The next time you wish to use an EEG-BEM solution based on the template mesh, SPM will simply use the date from the `single_subj_T1_EEG_BEM.mat` file (so this step will be much quicker the next time you do it). The same principle applies to EEG-BEM solutions computed from meshes based on subjects individual MRIs.

### 40.3.4 Invert

Now press the Invert button and

- Select an “Imaging” reconstruction.
- Select “Yes” for “All conditions or trials”.

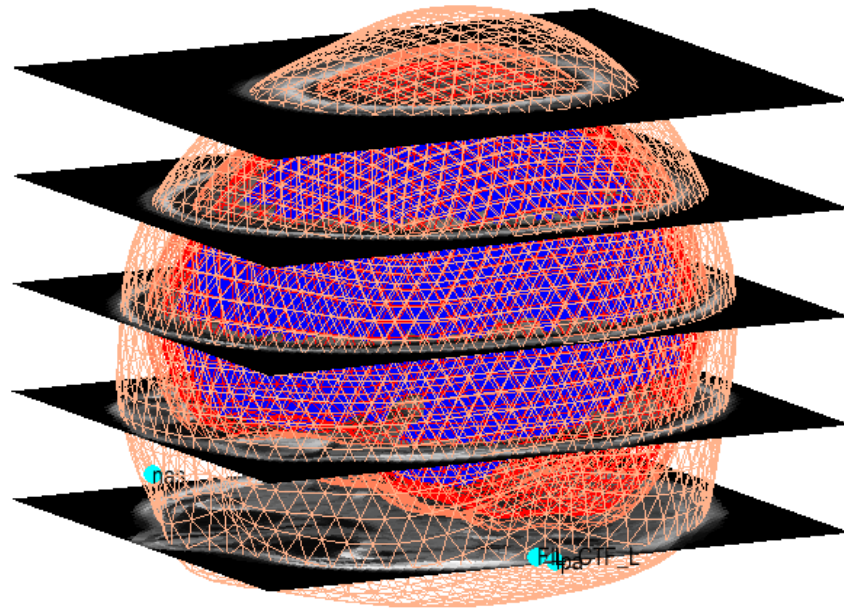


Figure 40.5: The figure shows the canonical cortical mesh (blue), inner skull surface (red) and scalp surface (light brown). The hardwired fiducials are shown in light blue. Transverse slices of canonical MRI are also shown in black, with gray scale inlays showing anatomical detail.

- Select “Standard” for Model.

SPM will now compute a leadfield matrix and save it in the file `SPMgainmatrix_maefdfMspmeeeg_subject1.mat` placed in `DATA_DIR`. This file can be replaced with one computed using other methods for computing the lead field (e.g. methods external to SPM). The forward model will then be inverted using the Multiple Sparse Priors (MSP) algorithm (the progress of which is outputted to the MATLAB command window). SPM will produce, in the Graphics window, (i) a Maximum Intensity Projection (MIP) of activity in source space (lower panel) and (ii) a time series of activity for (upper panel) each condition.

The “ms or mm” window has three functionalities (i) if you enter a single number this will be interpreted as ms, (ii) if you enter two numbers this will be interpreted as a time window for plotting movies (see below), (iii) if you enter 3 numbers this will be interpreted as MNI coordinates for a time series plot.

Now enter “160” for “ms or mm” and press the MIP button, to see a MIP of activity in source space at 160ms post-stimulus, and the time series of activities (top panel) at the position with largest magnitude signal. The corresponding graphic is shown in Figure 40.8. By toggling the “Condition” button, and pressing MIP each time, you can view the spatial distribution of activity for the different conditions (at the selected time point).

### 40.3.5 Batching source reconstruction

All the functionality of source reconstruction can be batched, using the tools from ‘Source reconstruction’ submenu of ‘M/EEG’. ‘Head model specification tool’ performs mesh generation, coregistration and forward model specification. ‘Source inversion’ tool computes the inverse solution and ‘Inversion results’ tool summarises the inversion results as images. One tip when incorporating source reconstruction batch in a script is one should be aware that the batch reads the `@meeg` object from disk and saves the results to disk but does not update the `@meeg` object in the workspace. Thus, it is advised to save any changes to the object before running the batch (`D.save`) and to reload the object after running the batch (`D = D.reload`).

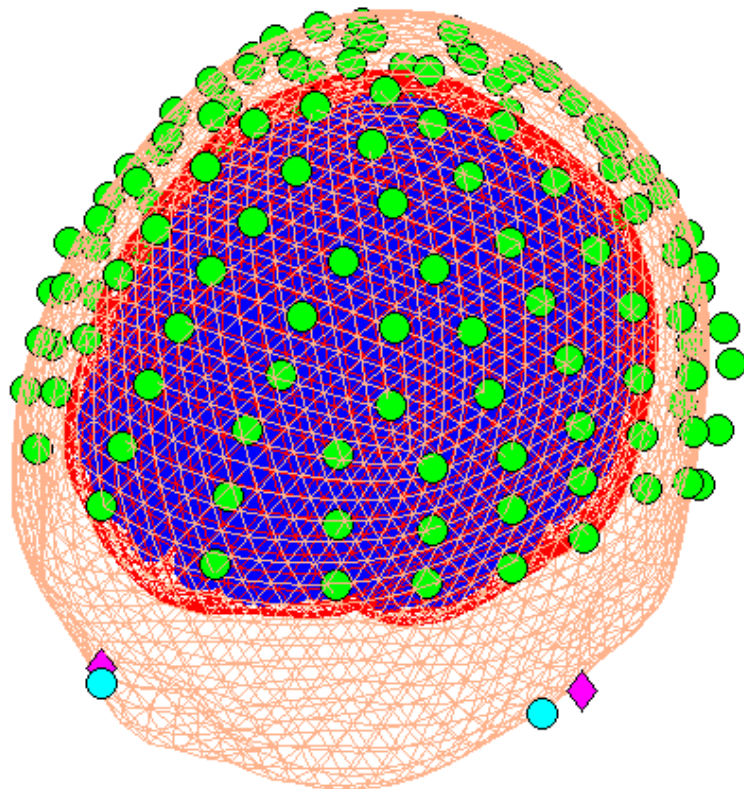


Figure 40.6: *The figure shows the MRI fiducials (pink), the sensor fiducials (blue) and the locations of sensors (green) in addition the the canonical cortical mesh (blue), inner skull surface (red) and scalp surface (light brown).*

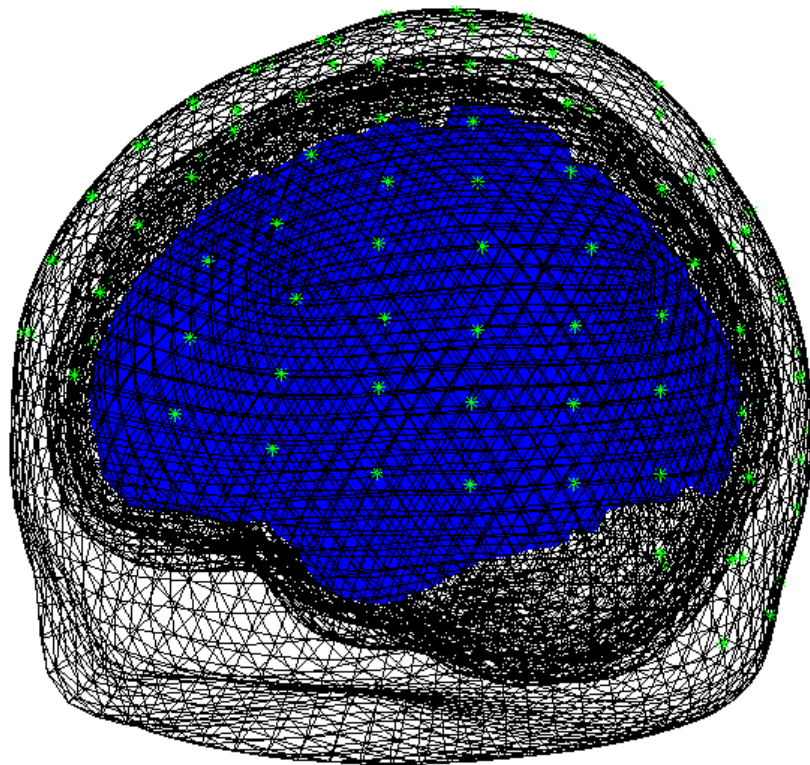
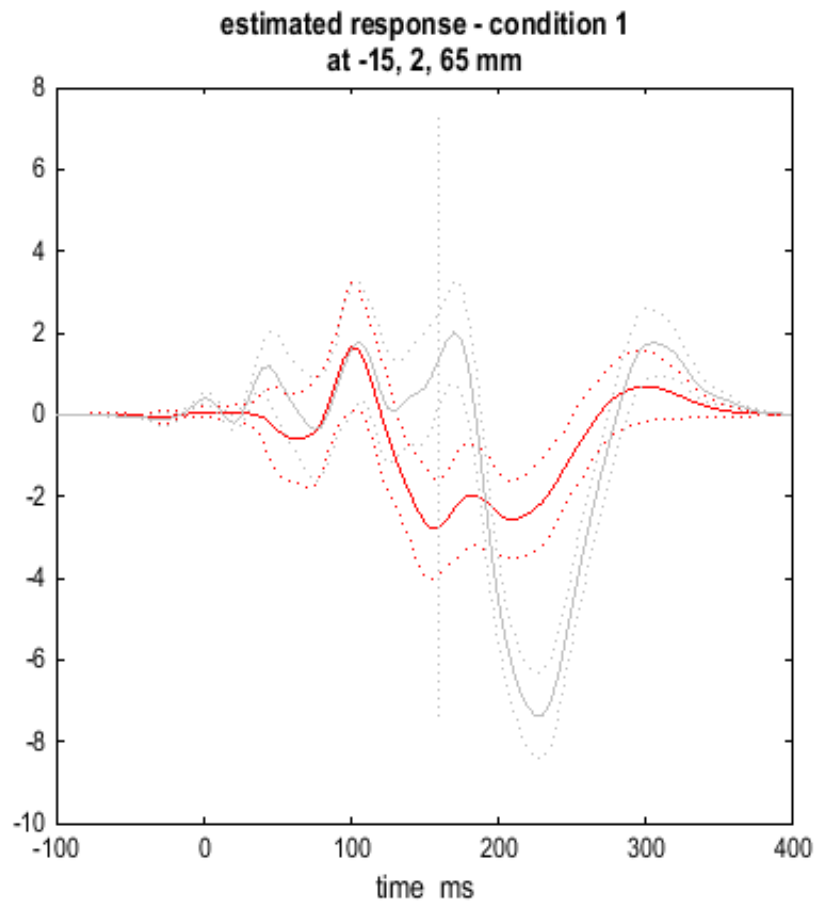
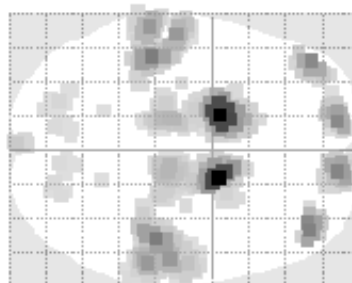
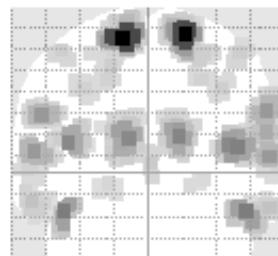
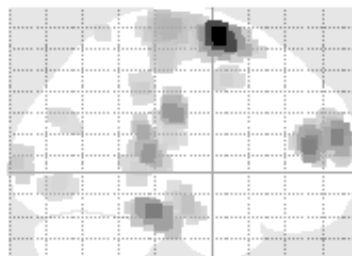


Figure 40.7: *The figure shows the cortical mesh (blue), brain, skull and scalp surfaces. Electrode positions are marked with asterisks.*



**PPM at 160 ms (59 percent confidence)  
512 dipoles  
Percent variance explained 98.10 (97.37)  
log-evidence = -2397.1**



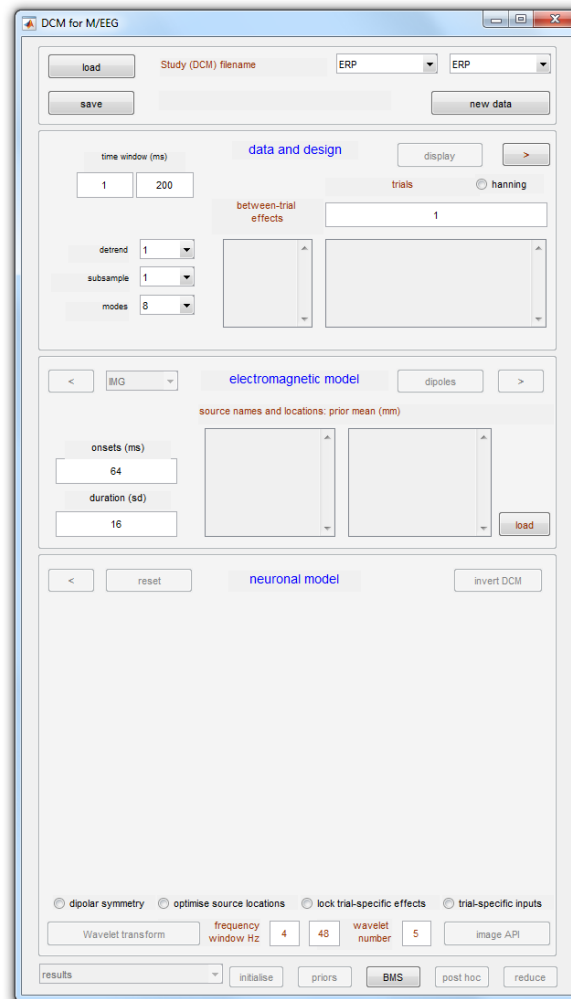


Figure 40.9: *The Dynamic Causal Modeling GUI splits model specification into three reversible phases (i) data and design, (ii) electromagnetic model and (iii) neuronal model. One can move forwards and backwards in the model specification using the left and right arrow buttons (these become highlighted when sufficient information has been entered to proceed to the next step).*

## 40.4 Dynamic Causal Modeling

Many of the functionalities of DCM for M/EEG are described in more detail in the reference chapter 16. In this chapter we demonstrate only the “DCM for ERP” model. Users are strongly encouraged to read the accompanying theoretical papers [23, 63]. Briefly, DCM for ERP fits a neural network model to M/EEG data, in which activity in source regions is described using differential equations based on neural mass models. Activity in each region comprises three populations of cells; pyramidal, local excitatory and local inhibitory. Fitting the model will then allow you to plot estimated activity in each cell population in each region. It will also provide estimates of the long range connections between regions, and show how these values are changed by experimental manipulation (eg. rare versus standard trial types).

In the `example_scripts` folder of the SPM distribution, we also provide an example script that will run a DCM-for-ERP analysis of this data. This can be edited to implement your own analysis.

Pressing the “DCM” button will open up the DCM GUI shown in Figure 40.9. We will now complete the three model specification entries shown in Figure 40.10:

- Press the “new data” button and select the `maefdfMspmeeeg_subject1.mat` file.

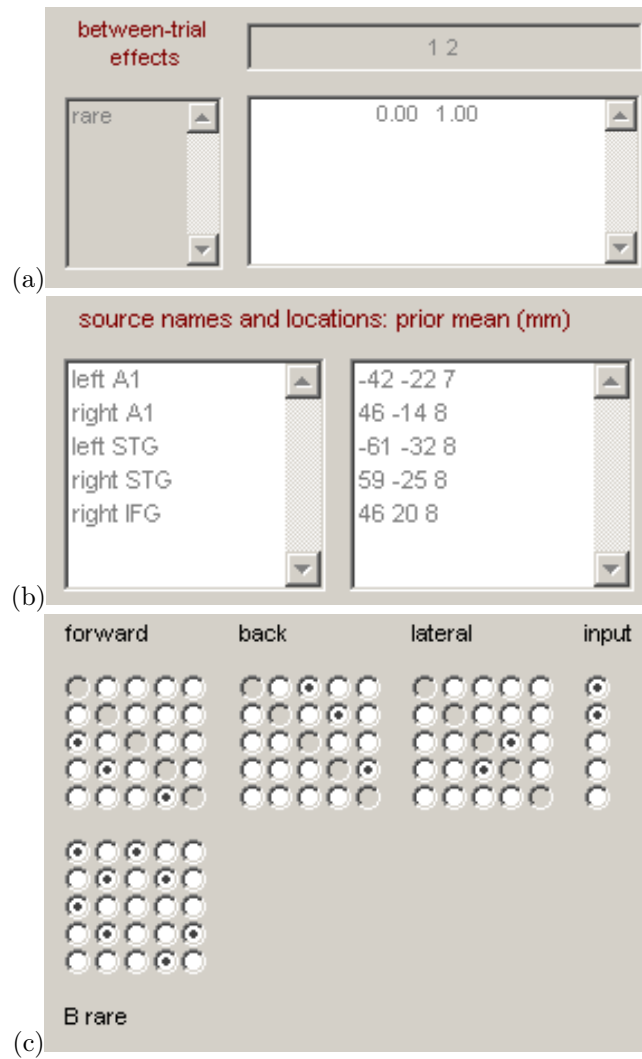


Figure 40.10: *Specification of DCM for ERP model (a) Data and design, (b) electromagnetic model and (c) neuronal model.*



- Enter the “between-trial effects” and design matrix information shown in Figure 40.10(a).
- Press the “Display” button.

This completes the data specification stage. Now:

- Press the right hand arrow to move on to the specification of the electromagnetic model.
- Instead of “IMG” select “ECD” for the spatial characteristics of the sources.
- Now enter the names and (prior mean) locations of the sources shown in Figure 40.10(b).
- Pressing the “dipoles” button will create an interactive display in the graphics window showing the prior source positions.

This completes the specification of the electromagnetic model. Now:

- Press the right hand arrow (next to the dipoles button) to move on to specification of the neuronal model.
- Highlight the connectivity matrix radio buttons so that they correspond to those shown in Figure 40.10(c).
- Press the (top left) ‘save’ button and accept the default file name.
- Press ‘Invert DCM’

SPM will plot the progress of the model estimation in the MATLAB command window. Plots of data and the progressing model fit will be shown in SPM’s graphics window. The algorithm should converge after five to ten minutes (in 64 iterations). Now select the “ERPs (sources)” option from the pull down menu to the right of the “Estimated” button. This will produce the plot shown in Figure 40.11. The values of the connections between areas can be outputted by selecting eg “Coupling(A)” from the pull-down menu in the DCM GUI. This will allow you to interrogate the posterior distribution of model parameters. It is also possible to fit multiple models, eg. with different numbers of regions and different structures, and to compare them using Bayesian Model Comparison. This is implemented by pressing the BMS button (bottom right hand corner of the DCM window).



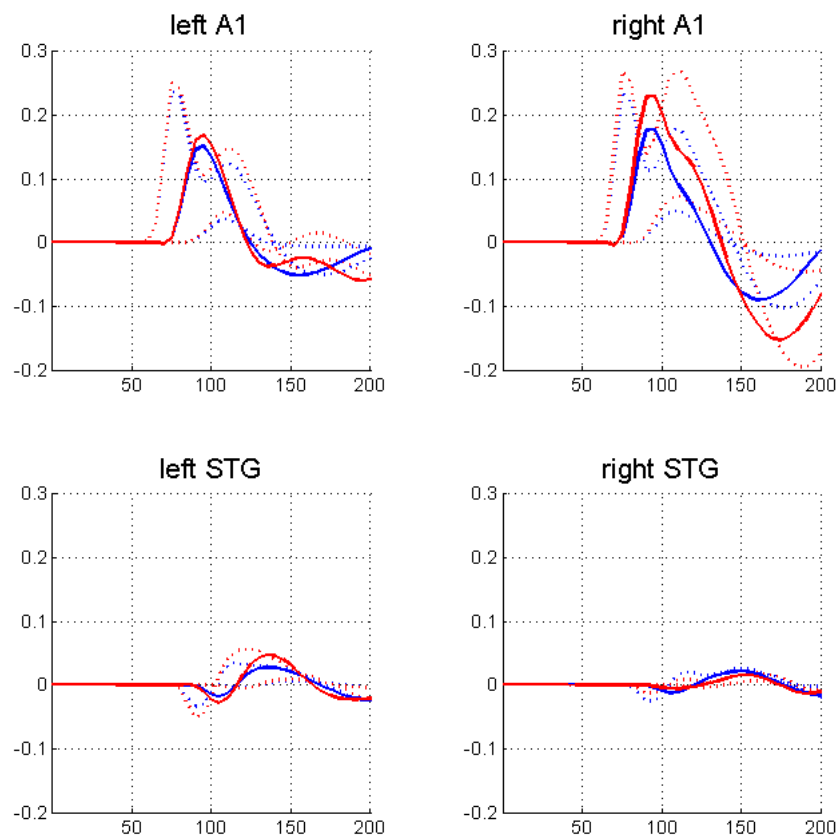


Figure 40.11: Activity plots for three neuronal populations (solid lines for pyramidal cells, dotted lines for others), in four areas (fifth not shown in this figure), for standard (blue) and rare (red) trial types.



## Chapter 41

# Advanced topics in M/EEG artefact removal

In the “Mismatch Negativity” tutorial chapter we presented two simple techniques for dealing with artefacts in M/EEG data: trial rejection based on channel thresholding and robust averaging. These methods do a good job in the majority of practical cases, especially for evoked responses. However, in some situations more sophisticated methods are required.

This chapter will demonstrate several of these advanced techniques. We will continue using the single subject mismatch negativity EEG data as an example.

### 41.1 Artefact marking

In some cases it is convenient to separate detection of the artefacts and using the information about them in subsequent processing. An example of such a situation is when one wants to detect flat segments or jumps in the data. In unprocessed data flat segments will be exactly flat with zero difference between adjacent samples and jumps will be sharp, making their detection straightforward. However, after filtering these clear features are likely to be distorted. On the other hand we might want to apply filtering to continuous data and only reject trials after epoching. In this case the ‘Mark’ mode in the ‘Detect artefacts’ tool can be used. The idea is that in ‘Mark’ mode artefacts are only detected and stored in the header as a special kind of events. With epoching, these events will be assigned their respective trials and then trials containing artefact events can be rejected by running the ‘Detect artefacts’ tool again in ‘Reject’ mode.

Let us use as an example the converted, high-pass filtered and downsampled EEG dataset from the MMN tutorial chapter (`dfMspmeeg_subject1.mat`). Our concern is that the next step which is low-pass filtering might distort high-frequency features useful e.g. for detection of muscle activity. Thus, we’ll run artefact detection at this stage.

Select ‘Detect artefacts’ from the ‘Preprocessing’ drop-down menu. Change ‘Mode’ from ‘Reject’ to ‘Mark’. Select the `dfMspmeeg_subject1.mat` file as input (if you don’t have this file, return to the MMN tutorial chapter and perform the steps described there to obtain it). In ‘How to look for artefacts’ make a new ‘Method’. In ‘Channel selection’ delete ‘All(1)’, add ‘Select channels by type’ and choose ‘EEG’ for the type. Under ‘Detection algorithm’ choose ‘Threshold z-scored difference data’. This tool computes the difference time series for each channels, z-scores the result and threshold the absolute z-score at the user-specified threshold (5 by default). ‘Excision window’ parameter which is also present in many other detection tools makes it possible to mark as bad a time segment around the detected artefact. For instance if we detect a jump at the data we would usually like to exclude also the adjacent data points from the analysis because they could be affected by e.g. hardware filter ringing. The default value of 100 means that the segment  $[-50\ 50]$  ms relative to the artefact will be excluded. If artefacts occur at distances closer than the excision window their bad segments will be merged and if the fraction of data marked as bad for a channel exceeds the ‘Bad channel threshold’ parameter, the whole channel will be marked as bad. Set ‘Bad channel threshold’ to 1 to prevent marking channels as bad at this stage.

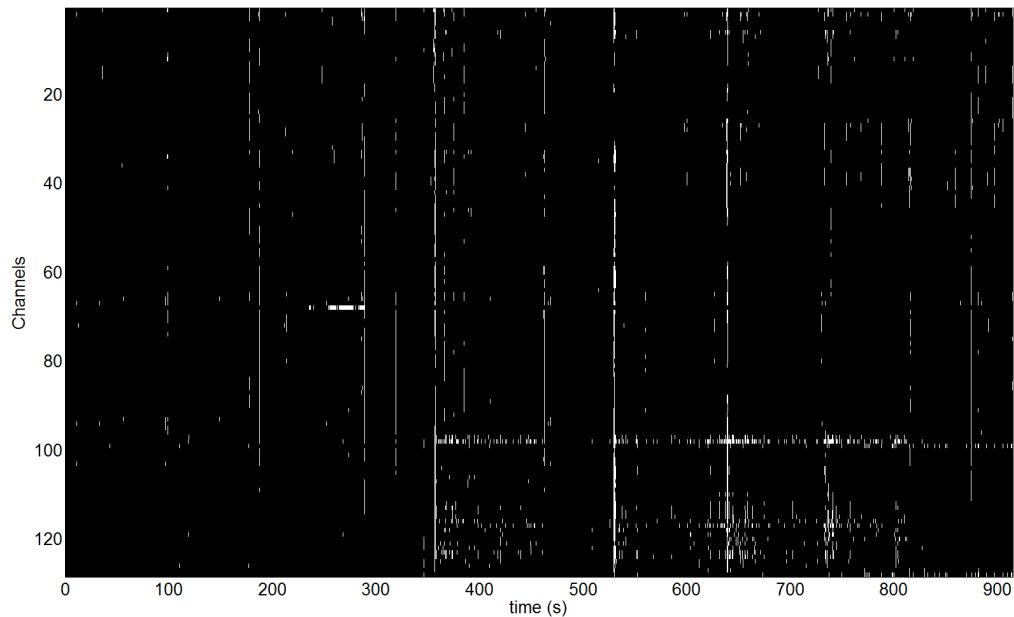


Figure 41.1: *Reviewing marked artefacts*

## 41.2 Reviewing marked artefacts

The artefact marking step produces a new dataset (`adfMspmeeeg_subject1`). Load the data set in the ‘Prepare’ tool (Convert→Prepare). In ‘Batch inputs’ menu select ‘Event list’. You can now see that in addition to the original events defined at acquisition there are multiple new events with type ‘`artefact_zscorediff`’ and values corresponding to names of the channels where artefacts were found. These artefact events can be used for epoching in the same way as other events and we will see the usefulness of that below.

We will now see another way of reviewing the artefacts using ‘`badsamples`’ method of `@meeg` object. Type `D = spm_eeg_load` in the command line and choose `adfMspmeeeg_subject1.mat` file in the dialogue that comes up. Type:

```
figure;
imagesc(D.time, [], badsamples(D, D.indchantype('EEG'), ':', 1));
colormap grey
```

A figure similar to the one in Figure 41.1 will be displayed. The ‘`badsamples`’ method translated the artefact events into a matrix with the same dimensions as our EEG channels x time data. In this matrix we can see for each data sample whether it is bad or not.

We can see in the figure that there were several times when some event affected almost all the channels and there is also a group of channels where these events occur more frequently than in other channels especially in the second half of the recording. This is probably due to muscle activity. Reviewing this kind of plot can be useful to identify problems with your recording and preprocessing pipeline. In the next sections we will see how this information can be used for subsequent automatic processing.

## 41.3 Trial rejection based on marked artefacts

Proceed with the low-pass filtering and epoching steps as described in the MMN tutorial chapter. We will assume that you now have the epoched dataset `efadfMspmeeeg_subject1`. Note that artefact marking can be integrated into the batch pipeline with the other steps. We will not describe this in detail but will leave as an exercise to the reader.

Open the ‘Detect artefacts’ tool again and choose `efadfMspmeeeg_subject1.mat` file as input. In ‘How to look for artefacts’ make a new ‘Method’. You can keep the ‘All’ for ‘Channel selection’ as it will not make a difference in this case. Under ‘Detection algorithm’ choose ‘Reject trials based on events’. By default all artefact events are used for rejection here but you could also load a custom event list that can be created and saved in the ‘Prepare’ GUI. Now you can run the tool. In the command window it will be reported that there are 319 rejected trials and 1 bad channels: D2. In reality it’s probably unnecessary to reject all these trials but we selected a rather low detection threshold for demonstration purposes. If you now proceed to averaging the generated output file (`aefadfMspmeeeg_subject1`), trials with artefacts will not be included in the average.

## 41.4 Explicit artefact exclusion in robust averaging

Rejecting any trial with an artefact in any of the channels sacrifices a lot of good data. This can be justified when we want to be conservative and rule out any contribution of the artefacts to our result. An alternative approach is to use robust averaging but this time to explicitly exclude the data marked as bad from the average. This can help, for example, with eye blinks consistently locked to some event in the trial. If there are many such eye blinks, they will not be suppressed by conventional robust averaging. But if we detect them first and remove them from the average, the average can be made eyeblink-free provided that there is sufficient number of trials without eye blinks. To demonstrate this approach, open the ‘Average’ tool. Choose the `efadfMspmeeeg_subject1.mat` file as input because we do not want to reject trials here. In ‘Averaging type’ switch to ‘Robust’. Set ‘Remove bad data’ to ‘yes’. If you run the tool now data marked as bad will be excluded from the average. It might be a good idea to low-pass filter the average again as robust averaging might introduce high frequency noise back in the data.

There are many possible ways to combine marking of different artefact types with trial rejection and robust averaging and we hope that our demonstration made the principles sufficiently clear to you to be able to explore them independently.

## 41.5 Topography-based artefact correction

If an eye blink occurs almost in every trial, then trial rejection will lead to discarding most of the data. Robust averaging could still work in this situation but only if the eye blinks do not consistently overlap in peri-stimulus time and for every sample eye blinks are only present in small minority of trials. But if, for instance, your subject blinks every time after making a decision in a gambling task, neither method would work. In this case a topography-based artefact correction would help. Note that depending on the experimental question, one could be more or less cautious about this. If you are interested in the activity of orbitofrontal cortex, it would probably be better to make every effort to ensure that eye-blinks are minimised at the recording stage and the remaining ones are completely excluded as any software correction will always leave residuals and it will be difficult to convince your reviewers that these residuals do not confound your experimental effects.

Topography-based correction implemented in SPM is based on principles quite similar to Independent Component Analysis (ICA). However, unlike for ICA there is no need to in the time consuming procedure of estimating physiological and artefactual components based on data statistics. But rather these are provided to the algorithm explicitly by the user. This makes the method faster and easier to use. We have not performed a systematic comparison with ICA but it is likely that for the purposes of eye blink removal the two methods yield very similar results.

Applying the topography based correction consists of several steps which can be implemented in one batch pipeline. We will again use the MMN dataset as the example and start from filtered and downsampled dataset `fdfMspmeeeg_subject1`.

For reasons that will become clear below we might need a forward model for some of the steps we will demonstrate. In principle, however, depending on the settings topography-based artefact correction is also possible without a forward model. Let us start with defining a forward model using the ‘Head model specification’ batch tool (SPM->M/EEG->Source reconstruction menu

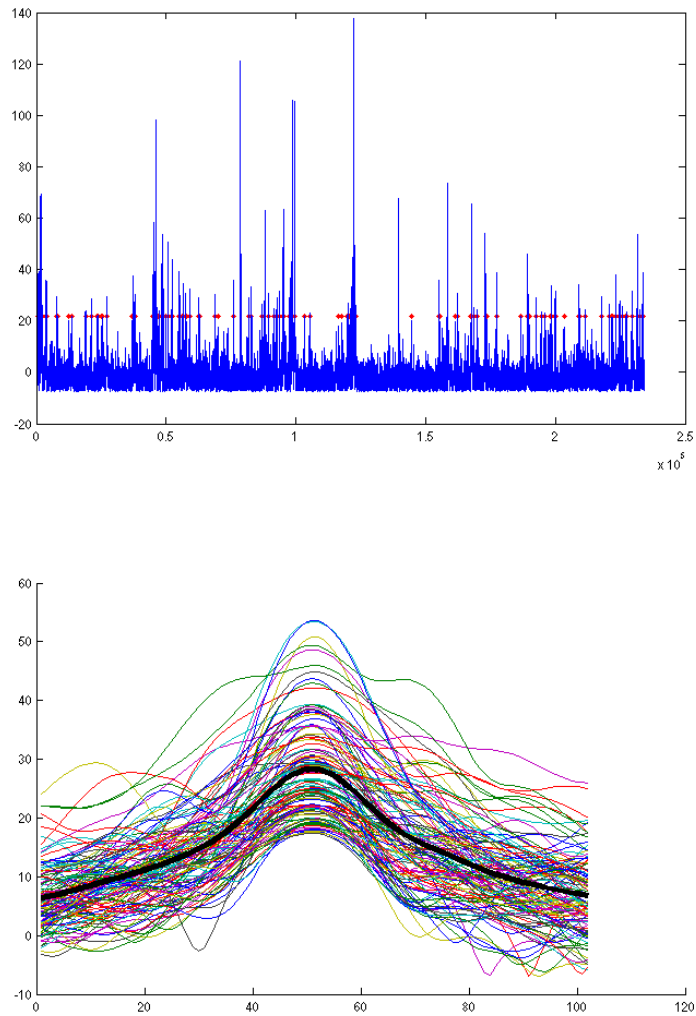


Figure 41.2: *Time points where eye blinks were detected (top) and aligned blink time courses (bottom).*

in batch). Select `fdfMspmeeeg_subject1` file as input, under ‘Coregistration’ switch to ‘Sensor locations are already in MNI space’ and run the tool. Now we are ready to proceed.

Select ‘Detect artefacts’ from the ‘Preprocessing’ drop-down menu. Change ‘Mode’ from ‘Reject’ to ‘Mark’. Select the `fdfMspmeeeg_subject1.mat` file as input (if you don’t have this file, return to the MMN tutorial chapter and perform the steps described there to obtain it). In ‘How to look for artefacts’ make a new ‘Method’. In ‘Channel selection’ delete ‘All(1)’, add ‘Custom channel’ and enter ‘VEOG’ for the channel name. If in your own data you do not have an EOG channel any scalp EEG channel or MEG channel with clear eyeblinks can be used for the same purpose. Under ‘Detection algorithm’ choose ‘Eyeblinks’. Keep the default settings and run the tool.

A plot similar to the one from Figure 41.2 will appear in SPM graphics window. It shows the points where blinks were detected (top) and aligned blink time courses (bottom). For topography-based correction it is not critical to detect all the blinks, just a representative sample will suffice. If you think too many non-eyeblinks or too few eyeblinks were detected you can adjust the ‘Threshold’ parameter.

At the next step we will epoch the data around detected eye blinks. Open the ‘Epoch’ tool in ‘Preprocessing’ menu. Choose the `afdfMspmeeeg_subject1.mat` produced by the previous step

as input. In ‘How to define trials’ choose ‘Define trial’. Set ‘Time window’ to  $[-500\ 500]$  and under ‘Trial definitions’ create a new trial with label ‘Eyeblink’, event type ‘artefact\_eyeblink’ and event value ‘VEOG’ (enter this with single quotes). If you remember the artefact events we reviewed in one of the previous sections, the principle is the same here and you would be able to see the eyeblink events in the ‘Prepare’ tool as well (note that ‘badsamples’ will only show them if you look at the VEOG channel as they are specific to that channel). Set ‘Baseline correction’ to ‘no’. It would also be a good idea to change the default out prefix ‘e’ to ‘eyeblink’ because we might epoch the same file later around stimuli and we would like to avoid a name clash. Now you can run the tool. The output dataset `eyeblinkafdfMspmeeeg_subject1` will contain epochs with eye blinks. You can review it in the reviewing tool and also average to get an average eye blink. Either epoched or averaged eyeblink file can be used to define eye blink topography. We will use the epoched file as this might enable to also better capture the variability between eyeblinks.

Select ‘Define spatial confounds’ from the ‘Preprocessing’ menu. Use the `eyeblinkafdfMspmeeeg_subject1` as input. Choose ‘SVD’ under ‘Mode’. The tool performs singular value decomposition of the trial data to find the spatial patterns that explain most of the variance in the data. What we would like to do now is to keep for artefact template the minimal number of components that are clearly eye-blink related. Since we do not know what that number is we could start with a large number e.g. 4 (set in ‘Number of components’) and run the tool.

Plot such as the one in Figure 41.3 will appear in the Graphics window. For averaged-referenced EEG eye-blink related activity appears at the frontal sensors. Thus only the first of the four components is clearly eye-blink related. We could, therefore, only keep that one for our correction. It is possible that with more accurate data preprocessing and removal of other artefacts additional eyeblink components could be extracted. Also for MEG data where head movement change the blink topography over time one component will usually not suffice. For now we will change the ‘Mode’ to ‘Clear’, run the tool again and then return ‘Mode’ to ‘SVD’, set ‘Number of components’ to 1 and run once again. The tool does not produce a separate output file and appends the confound topographies to the same file. Thus clearing is necessary to remove the results of the first run. For your own analysis you might want to explore the typical numbers of eye-blink components for different subjects and runs and decide whether it is safe to always use the same number of check for each file separately.

The next step is to use the artefact topography we defined using SVD to correct the data we are actually interested in. For this we will need to use ‘Define spatial confounds’ tool once again, but this time our data of interest will be the input, in this case the continuous data file `afdfMspmeeeg_subject1.mat`. Under ‘Mode’ switch to ‘SPM M/EEG Dataset’ and choose the `eyeblinkafdfMspmeeeg_subject1` for which we defined confounds above. Run the tool and the confound definition will be copied from `eyeblinkafdfMspmeeeg_subject1` to `eyeblinkafdfMspmeeeg_subject1`.

Another way to define spatial confounds is to use the ‘Eyes’ options under ‘Mode’. The idea there is that three orthogonal dipoles are placed at each eye and their lead-fields are computed using the forward model (that’s one place where you would need one) and used as artefact topographies. If you want to try this option do not forget to clear the previously defined components first. A plot like the one in Figure 41.4 will appear.

You can see here that all the 6 components are expressed at the frontal sensors. This method can also work for removing eye-blinks. Its advantage is that also other activities coming from the eyes can possibly be captured (such as eye movements). However, you will have to sacrifice 6 dimensions of your data which is effectively like removing 6 channels. If you do not have many channels to begin with this can distort your sensor waveforms quite substantially (which might or might not matter depending on the aim of your analysis). Also if the forward model is imprecise it can also happen that some eye-blink related activity will not be removed. Thus where possible the data-driven (SVD) approach is advised.

We are now ready to correct our data. Choose ‘Correct sensor data’ from the ‘Preprocessing’ menu. Choose `afdfMspmeeeg_subject1.mat` as input. There are two options for correction mode. ‘SSP’ (default) removes everything that can be linearly matched by the artefact topographies from the data, basically making the data orthogonal to the artefact. This method does not require a forward model so if you use SVD in combination with SSP setting you do not have to define a forward model for your data. ‘Berg’ method uses the forward model to define ‘representative’ cortical topographies and keeps the part of the variance that is shared between

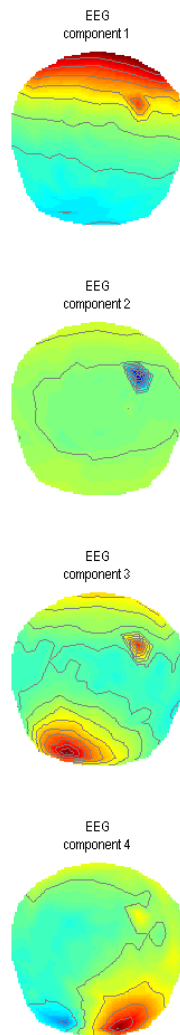


Figure 41.3: *Spatial confounds derived from SVD*





Figure 41.4: *Spatial confounds derived using the EYES option*

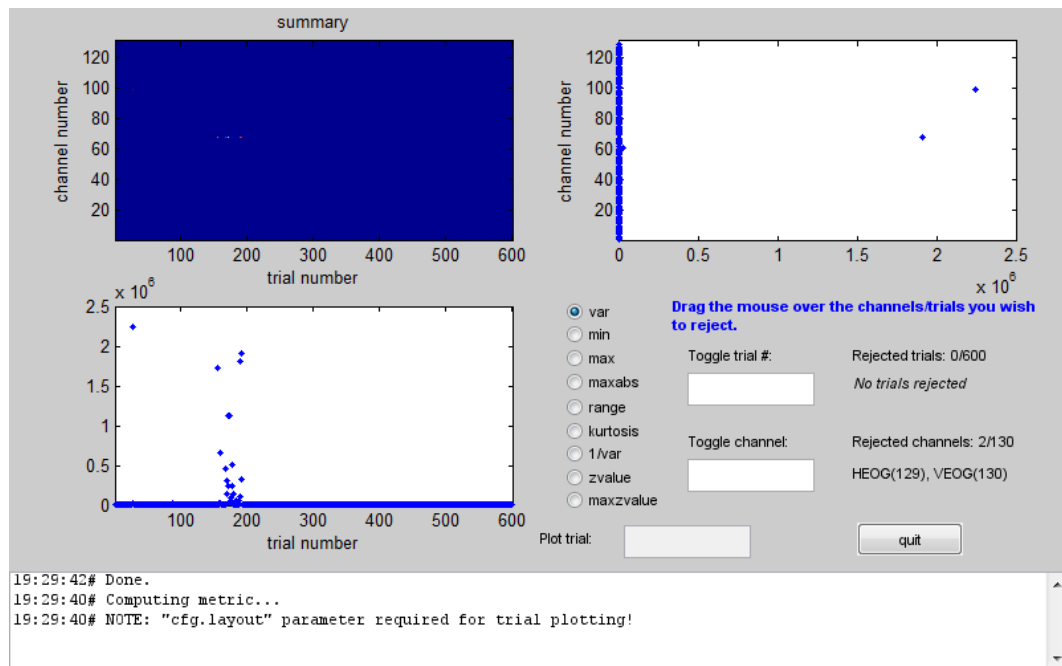


Figure 41.5: *Fieldtrip visual artefact rejection GUI before rejecting trials*

cortical and artefact topographies, thereby only removing the part that is unlikely to come from the cortex. This method is analogous (though slightly differently implemented) to the one in BESA software (Berg P, Scherg M. *Electroencephalogr Clin Neurophysiol.* 1994 Mar;90(3):229-41). It requires forward model to be defined.

Artefact correction will produce a new dataset with T prefix: `TafdfMspmeeeg_subject1.mat`. You might want to run both ‘SSP’ and ‘Berg’ correction changing the default prefix to generate two separate files and compare them later. Review the continuous datasets in the reviewing tool and compare with the uncorrected file to make sure the eyeblinks are indeed gone. You can also epoch and average the corrected files around the eyeblink events and compare the average of the eyeblink dataset we created before (use ‘Merge’ tool to combined different eyeblinks in the same dataset and plot them together). Equivalently, you could apply topography-based correction to the average eyeblink. Since it is a linear operation it does not matter whether you do it on continuous, epoched or averaged data. As a final exercise, you can test the effect of increasing the number of SVD components and compare with the ‘Eyes’ method.

## 41.6 Fieldtrip visual artefact rejection

The last tool we are describing here is a compromise between the automatic methods implement in ‘Detect artefacts’ tool and the approach of some very meticulous researchers who examine all their trials by eye. This tool comes from the FieldTrip toolbox (see <sup>1</sup>) and SPM only provides an easy interface to it. It can be run by choosing ‘MEEGTools’ from the ‘Toolbox’ menu and then choosing ‘Fieldtrip visual artefact rejection’. You will first need to choose the input file. For our demonstration here we will use the `efdfMspmeeeg_subject1` from the MMN tutorial. The tool has three modes that you can choose. We’ll only describe the ‘Summary’ mode here. You can read about the other two at the FieldTrip website (see the link above). You will also be asked to specify the time window you want to look at. This is useful if you have a stimulus artefact and only want to look at part of the peristimulus time. Accept the default which is the whole epoch. A window such as the one in Figure 41.5 will appear.

In the centre of the window there is a column of radio buttons that define different measures

<sup>1</sup>FieldTrip: [http://fieldtrip.fcdonders.nl/tutorial/visual\\_artifact\\_rejection](http://fieldtrip.fcdonders.nl/tutorial/visual_artifact_rejection)

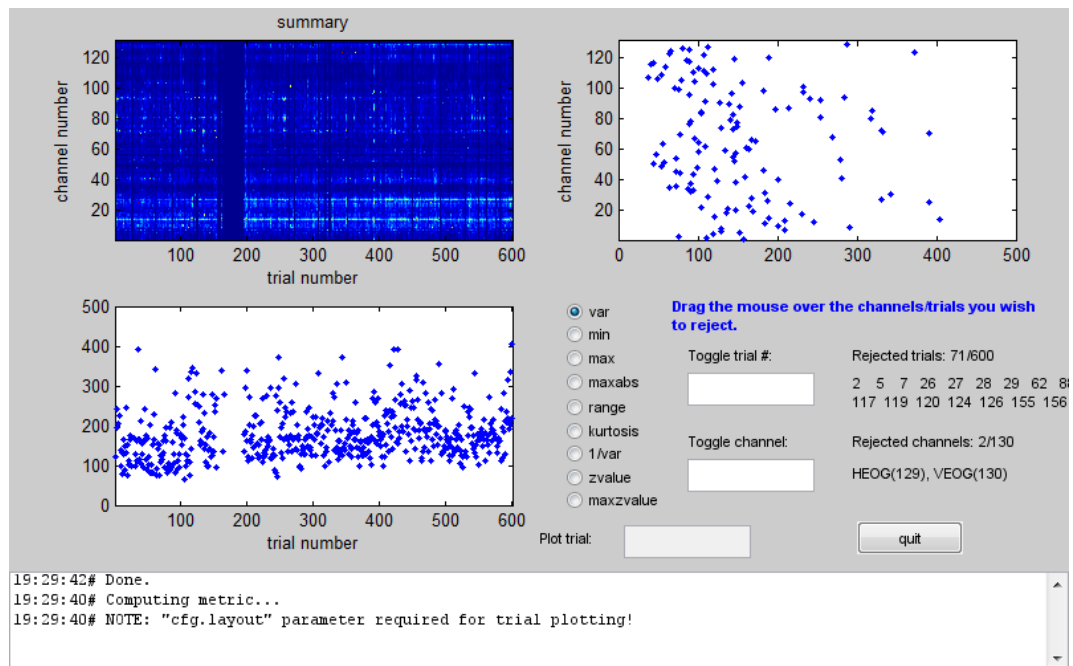


Figure 41.6: *Fieldtrip visual artefact rejection GUI after rejecting outliers*

that can be computed for each channel and trial. When you choose one of these measures the tool computes it for all trials and channels and presents it a channel x trial image at the top left. The plots below and to the right of the image show the maximal values of the measure across channels and across trials respectively. In the above example you can see that between 150th and 200th trial there was a series of trials with unusually high variance. An additional trial with high variance occurred earlier in the recording. If you look at the channel plot you see that there are two channels with higher variance than the others. It is possible that the artefacts are only on those channels and by excluding them you will be able to retain all the trials. Conversely, you could exclude all the bad trials and then the channels will probably not be outliers any more. The choice is in your hands depending on the data and the details of your analysis. Here we will exclude trials. This can be done by clicking on the trials plot and dragging the mouse over the points corresponding to the trials we want to exclude while holding the mouse button. After the initial exclusion the plot will be updated and you will see that there are more outliers that are closer to the group than the ones we initially saw but still quite far away. You can exclude them as well in the same manner. By doing this iteratively you can eventually get to a plot like the one in Figure 41.6.

Here there are no clear outliers in either channels or trials. You could then switch to a different measure which might give you a different picture and keep going until there are no outliers for any of the measures. If you then press the ‘quit’ button your SPM dataset will be updated and the trials and/or channels you chose will be marked as bad. Note that this tool does not create a new dataset so all the changes are done to the dataset you used as input. Make a copy of your dataset first if you want to play with different options. Alternatively you can remove all the bad flags by loading your dataset into the workspace using

```
D = spm_eeg_load
```

and then typing in the commands

```
D = badtrials(D, ':', 0);
D = badchannels(D, ':', 0);
save(D);
```



## Chapter 42

# Multimodal, Multisubject data fusion

### 42.1 Overview

This dataset contains EEG, MEG, functional MRI and structural MRI data from 16 subjects who undertook multiple runs of a simple task performed on a large number of Famous, Unfamiliar and Scrambled faces. It will be used to demonstrate:

1. batching and scripting of preprocessing of multiple subjects/runs of combined MEG and EEG data,
2. creation of trial-averaged evoked responses,
3. 3D scalp-time statistical mapping of evoked responses across trials within one subject,
4. 2D time-frequency statistical mapping of time-frequency data across subjects,
5. preprocessing and group analysis of fMRI data from the same subjects and paradigm,
6. source-reconstruction of the “N/M170” face component (using structural MRI for forward modelling),
7. individual and group-based fusion of EEG and MEG during source reconstruction,
8. statistical mapping across subjects of cortical power in a time-frequency window, using the functional MRI results as spatial priors.

For the formal basis of these steps, see SPM publications, most specifically [Henson et al. (2011)]<sup>1</sup>.

The raw data can be found here (see `README.txt` there for more details):

[ftp://ftp.mrc-cbu.cam.ac.uk/personal/rik.henson/wakemandg\\_hensonrn/](ftp://ftp.mrc-cbu.cam.ac.uk/personal/rik.henson/wakemandg_hensonrn/)

The MEG data consist of 102 magnetometers and 204 planar gradiometers from an Elekta VectorView system. The same system was used to simultaneously record EEG data from 70 electrodes (using a nose reference), which are stored in the same “FIF” format file. The above FTP site includes a raw FIF file for each run/subject, but also a second FIF file in which the MEG data have been “cleaned” using Signal-Space Separation as implemented in MaxFilter 2.1. We use the latter here. A Polhemus digitizer was used to digitise three fiducial points and a large number of other points across the scalp, which can be used to coregister the M/EEG data with the structural MRI image. Six runs (sessions) of approximately 10mins were acquired for each subject, while they judged the left-right symmetry of each stimulus (face or scrambled), leading to nearly 300 trials in total for each of the 3 conditions.

<sup>1</sup>[http://www.fil.ion.ucl.ac.uk/spm/doc/papers/HensonEtAl\\_FiHN\\_11\\_PEB\\_MEEG.pdf](http://www.fil.ion.ucl.ac.uk/spm/doc/papers/HensonEtAl_FiHN_11_PEB_MEEG.pdf)

The MRI data were acquired on a 3T Siemens TIM Trio, and include a  $1 \times 1 \times 1$ mm  $T_1$ -weighted structural MRI (sMRI) as well as a large number of  $3 \times 3 \times \sim 4$ mm  $T_2^*$ -weighted functional MRI (fMRI) EPI volumes acquired during 9 runs of the same task (performed by same subjects with different set of stimuli on a separate visit). (The FTP site also contains ME-FLASH data from the same subjects, plus DWI data from a subset, which could be used for improved head modelling for example, but these are not used here.) For full description of the data and paradigm, see [Wakeman and Henson (2015)]<sup>2</sup>.

Versions of the SPM12 batch job files and scripts used in this chapter can be found here:

<ftp://ftp.mrc-cbu.cam.ac.uk/personal/rik.henson/SPMScripts/>

It should be noted that the pipeline described below is just one possible sequence of processing steps, designed to illustrate some of the options available in SPM12. It is not necessarily the optimal preprocessing sequence, which really depends on the question being asked of the data.

## 42.2 Getting Started

Download the data from above FTP site. There are over 100GB of data in total, so you can start by just downloading one subject (e.g, Subject 15 that is used in the first demo below), and perhaps just their MEEG sub-directory initially (though you will need the T1 and BOLD sub-directories later)<sup>3</sup>. Within the MEEG sub-directory, you will need all the MaxFiltered files (run\_0[1-6]\_sss.fif), the bad\_channels.mat file and the Trials sub-directory. It will be much easier if you maintain this directory structure for your copy of the data.

Open SPM12 and ensure it is set to the EEG modality. To do this, type `spm eeg` into the MATLAB command window. For this to work, SPM12 root folder must be in your MATLAB path.

Open the batch editor window by pressing “Batch” from the SPM Menu window. This opens the window shown in Figure 42.1.

## 42.3 Preprocessing M/EEG data

We will start by creating pipelines (using SPM’s batch interface) for preprocessing the M/EEG data for a single subject, and then scripting these pipelines to repeat over multiple subjects. For no particular reason, we will start with Subject 15.

### 42.3.1 Convert (and epoch)

The first step is to convert raw M/EEG data from its native format (which depends on the acquisition system) to the format used by SPM. In the present example, the raw data are continuous. They can be converted to continuous SPM data, but to save disk space and time, we can “cut out” time windows (epochs) around the trials during the conversion step, so that the resulting SPM file contains epoched data.

In the batch editor, select SPM on the top toolbar, and from the dropdown menu select M/EEG. At the top of the new dropdown menu, select “Conversion”. Once selected, the Module List on the left of the batch editor window will now list “Conversion” as the first (and only) step. Within the main, Current Module window will be listed several variables. The first variable listed is “File Name”. On the right hand of this pane, you will see “<-X”, this indicates that you need to update this field. To do so, click on “File Name”, this will then open up your current working directory. For this example we will be using the M/EEG data from subject number 15 (in the Sub15 directory). Select the file named `run_01_sss.fif` and press “Done”.

<sup>2</sup><http://www.nature.com/articles/sdata20151>

<sup>3</sup> On Linux, from the command line, you can type the following to download all of the relevant data from Subject 15:

```
wget -r -nH -cut-dirs=3 ftp://ftp.mrc-cbu.cam.ac.uk/personal/rik.henson/wakemandg_hensonrn/Sub15/
-X "/*/*/*/*DWI/*/*/*/*FLASH/" -reject raw.fif
```

On Windows, you can access the FTP server from the File Explorer and copy the entire folder on your hard disk. Alternatively, you can use dedicated software such as [FileZilla](#) or [WinSCP](#).

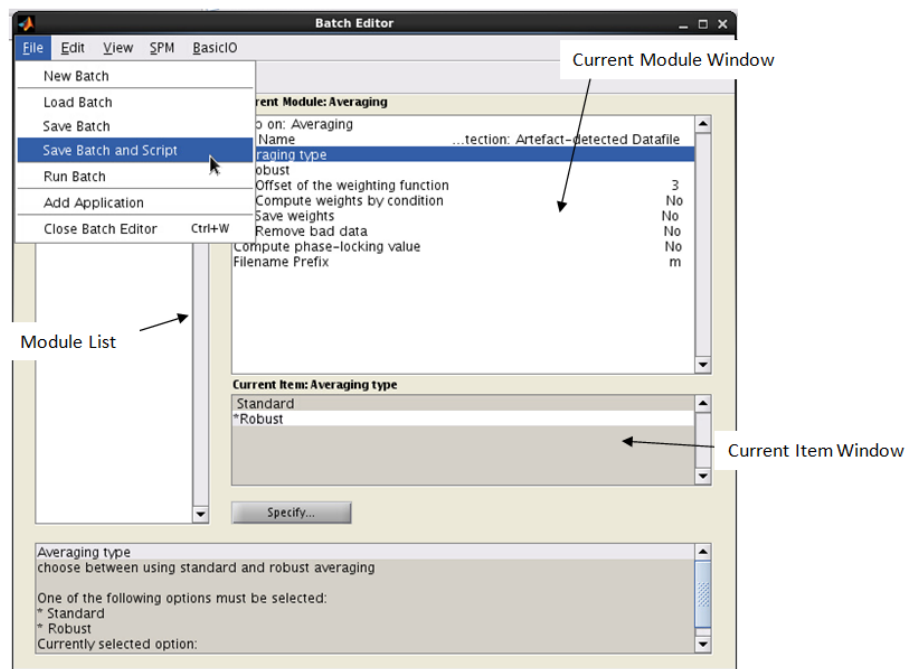


Figure 42.1: Screenshot of the Batch Editor. The Module List, Current Module Window and Current Item Windows are identified. The cursor highlights how to save the pipeline as a batch and script.

Many variables in the Current Module window have default values, but we need to change some of them. For example, we want to epoch during conversion, so highlight the “reading mode” and change from “continuous” to “epoched”. To epoch, we need to define the start and end of each trial. This information normally comes from a trigger channel containing pulses that encode the onset and type of trial. You can ask SPM to read this channel and select the relevant trial codes by hand. However, here we will use “trial definition” files that have been created in advance (given the complexity of the trigger codes in this dataset; contact authors for more information on trial definitions). So select “Trial definition file” under the “epoched” option, click on “Specify” and select the file `run_01_tr1def.mat`. (The contents of this file conform to the FieldTrip format, with one row per trial, each with three columns, which represent the sample numbers from the start of the continuous recording for 1) onset of the epoch, 2) offset of the epoch and 3) pre-stimulus baseline period.) Each trial runs from -500ms to +1200ms relative to the stimulus onset. (In fact, we only really care about the period -100ms to +800ms in the analysis below, and will later “crop” the trials to this period, but we first need the extra 400ms at the start and end to provide “padding” to avoid “end-effects” in the wavelet analysis done later.)

Another change to the defaults is that we do not want to convert all channels in the original file (since many are extraneous), so will select a subset by their type. We first need to delete the default option to convert all channels. To do this, click “channel selection”, and scroll down until you can select the “Delete All(1)” option. Then click the “channel selection” again, but this time choose the option “New: Select channels by type”. This will add “Select channels by type” to the Current Module, and you will see “<-X” on the right hand side of this, indicating the need for user input. Click the “<-X” and then select “EEG” from the “Current Item” section. Repeat this process to additionally include the “MEGMAG” and “MEGPLANAR” channels.

The remaining options for conversion can be left with their default values (which includes the output filename, which defaults to the input filename, simply prepended with `spmeeeg_`). Once all variables are specified, the “play” button on the top toolbar will turn green and the batch could be run. However, for this example, we will continue to use the current batch editor window, so do not press the “play” button yet.

### 42.3.2 Prepare

The next step in the current pipeline is to update some other properties of the data using the “Prepare” module. This is a general-purpose “housekeeping” module that includes options like re-defining channel names, types, locations, etc. as specific to the particular laboratory set-up. In our case, some of channels currently labelled EEG were in fact used to record EOG.

Select “Prepare”, from the preprocessing menu. Highlight “Prepare” in the Module list, this will open up the variables in the current module window. Again we need to complete those variables indicated by the “<-X”. If we had already run the previous conversion stage, we could select the new `spmeeeg_run_01_sss.mat` file produced by that stage as the input for this stage. Here however, we will create a pipeline in which all stages are run in one go, in which case we need to tell SPM that the output of the conversion step, even though not yet created, will be the input of this preparation step. You can do this by selecting the “Dependency” button located further down the window. This will open up a new window, listing all the processing steps up to this point. So far this is just one, the conversion step. Highlight this step and select “OK”.

The next variable to define is the “task(s)”. Clicking this variable will display a variety of options in the “current item” box. Within this, select “New: Set channel type”, then return to the current module window. In here, highlight “Channel selection”, which displays further variables within the current item window. Select “New: Custom channel”. Now return to the current module box and select the variable with an “<-X”. This should be “Custom channel”. Selecting this and clicking on “Specify” will open up a small text window. Within this, type `EEG061`. Create a new Custom channel and type `EEG062`; then select “new channel type” and highlight “EOG”. This is because channels `EEG061` and `EEG062` in fact represented bipolar horizontal (HEOG) and vertical (VEOG) electroculagrams respectively, which can be used to detect ocular artifacts.

This process then needs to be repeated for two other new channel types. First create a new channel type like above, label it `EEG063` and set its channel type as “ECG”. This channel represents the electrocardiogram, which can be used to detect cardiac artifacts. Second, create another new channel labelled `EEG064` and set its channel type as “other” (this is just a free-floating electrode, and does not contain data of interest).

One more job we need to do is specify “bad” channels. These only exist for the EEG (MEG bad channels are assumed to be corrected by the prior MaxFiltering step). Bad EEG channels were identified by hand by an experienced EEG operator and saved in a file called `bad_channels.mat` in each subject’s directory. It is important to note that there are many other ways that bad channels can be defined automatically by SPM, using the “artefact” module for example, but the various options for this are not explored in this chapter. To specify bad channels, add a second Prepare module, select the “New: Set/unset bad channels”, then under “Channel selection”, replace the default “All” with the option “New: Channel file”, and then select the `bad_channels.mat` file in this subject’s directory (note it is important that you delete the “All (1)” option first). In fact, this subject does not have any bad channels, so this step could be ignored if you only want to analyse this subject. But if you want to use the preprocessing script that we will use below for the remaining 15 subjects, you need to set-up this stage now (i.e. the `bad_channels.mat` files for the other subjects are not empty).

This will complete the preparation stage, as indicated by the “play” button turning green. (Note that you can also use this “Prepare” option interactively, in order, for example, to create and save a bad channel selection file).

### 42.3.3 Downsample

The data were sampled at 1000Hz, but for the analyses below, we rarely care about frequencies above 100Hz. So to save processing time and disk space (by a factor of 5) we can therefore downsample the data to 200Hz (which includes a lowpass filtering to avoid aliasing). Select “downsampling” from the module list “SPM – M/EEG – Preprocessing”, click on “File Name”, select “Dependency” and in the pop-up window, select the prepared datafile at the bottom of the list. Next, set the downsampling rate by clicking the “New sampling rate” variable within the current module box. Type 200 into the pop-up window that appears and use “OK” to set this value. The output file of this stage will be prepended with a `d`.



### 42.3.4 Baseline Correction

The next step in this preprocessing pipeline is to baseline correct the data by subtracting from each channel the mean from -100ms to 0ms relative to the stimulus onset. Select “Baseline Correct” from under the “SPM – M/EEG – Preprocessing” menu. Select the input file name as being dependent on the output of the downsampling by selecting the “Dependency”. To define the baseline, highlight the “Baseline” button in the current module window and click on the “Specify” button. The baseline needs to be entered as a 1-by-2 array. Enter [-100 0] (units are milliseconds). The prefix of this file name will be `b`. Note, that to be on the safe side it might be a good idea to perform baseline correction before downsampling. The reason is that in the general case there might be large DC offsets in the data leading to prolonged ringing that even padding will not protect against. For our data, however, the order shown here works equally well.

### 42.3.5 Deleting intermediate steps (optional)

The four steps (modules) described above create a preprocessing pipeline for the data. If this pipeline is run straight away, there will be four different output files. If you are short of disk space, you might want to delete some of the intermediate files. To do this, select “SPM” from the top toolbar of the batch editor window and choose “M/EEG – Other – Delete” several times. Then you will need to specify the File Names to delete. Highlight each “Delete” module and set the File Name as the output of the Prepare step using the “Dependency” button to delete any output from the conversion/prepare step onward. In fact, all processing steps up until the most recent step (Baseline-correction) can be deleted. To do this, right click on “Delete” in the Module List and select “Replicate Module”, but change the dependency to the downsampled file.

**Creating a script for combining pipelines within a subject.** Once you have created a linear pipeline, you might want to repeat it on multiple sessions (runs) within a subject, or even across multiple subjects. In the present case, there were 6 independent MEG runs (separated only by a short period to give the subjects a rest), which can all be processed identically. One option would be to save the batch file, manually alter the “File Name” that is initially loaded into the batch editor, and repeat this process separately for each run. A more powerful approach is to create a script. To do this, select “File” from the Batch Editor window, and select “Save Batch and Script”. This will produce two files: a batch file (same as that created when you save a batch) but also a MATLAB script that calls that batch file. So if you call the batch file `batch_preproc_meeg_convert`, you will get a batch file called `batch_preproc_meeg_convert_job.m` and a script file called `batch_preproc_meeg_convert.m`.

The script file `batch_preproc_meeg_convert.m` will automatically be loaded into the MATLAB editor window, and should appear something like this:

```
% List of open inputs
% Conversion: File Name – cfg_files
% Conversion: Trial File – cfg_files
% Prepare: Channel file – cfg_files
nrun = X; % enter the number of runs here
jobfile = {'batch_preproc_meeg_convert_job.m'};
jobs = repmat(jobfile, 1, nrun);
inputs = cell(3, nrun);
for crun = 1:nrun
    inputs{1, crun} = MATLAB_CODE_TO_FILL_INPUT; % Conversion: File Name – cfg_files
    inputs{2, crun} = MATLAB_CODE_TO_FILL_INPUT; % Conversion: Trial File – cfg_files
    inputs{3, crun} = MATLAB_CODE_TO_FILL_INPUT; % Prepare: Channel file – cfg_files
end
spm('defaults', 'EEG');
spm_jobman('run', jobs, inputs{:});
```

At the top of this script is listed the variable `nrun = X`: replace `X` with 6 for the six runs you wish to convert. You also need to complete the missing MATLAB code needed for each run: here, 1) the raw input file to convert, 2) the trial definition file for that run, and 3) the channel file containing the bad channels (which is actually the same for each run, but differs across subjects, which will matter later when we extend the script to loop over subjects). In order to automate selection of these files, you need to know some basic MATLAB. For example, because the files are named systematically by run, we can complete the relevant lines of the script with:

```

inputs{1, crun} = cellstr(fullfile(rawpth, 'Sub15', 'MEEG', sprintf('run_%02d_sss.fif', crun)));
inputs{2, crun} = cellstr(fullfile(rawpth, 'Sub15', 'MEEG', 'Trials', sprintf('run_%02d_trldef.mat',
    crun)));
inputs{3, crun} = cellstr(fullfile(rawpth, 'Sub15', 'MEEG', 'bad_channels.mat'));

```

where `rawpth` refers to your base directory, `Sub15` is the subject directory that contains `MEG` and `Trials` sub-directories (assuming you mirrored the directory structure from the FTP site) and `%02d` refers to the run number, expressed as two digits.

The other file – the batch or “job” file – can be reloaded into the batch editor at any time. It can also be viewed in the MATLAB editor. If you type `edit batch_preproc_meeg_convert_job.m`, you will see your selections from the earlier GUI steps. But we need to change those selections that depend on the run number (so they can be passed from the script instead). To make the batch file accept variables from the script file, we need change three of the specifications to `'<UNDEFINED>'` instead. So edit the following lines so they read:

```

matlabbatch{1}.spm.meeg.convert.dataset = '<UNDEFINED>';
matlabbatch{1}.spm.meeg.convert.mode.epochs.trlfile = '<UNDEFINED>';
...
matlabbatch{2}.spm.meeg.preproc.prepare.task{4}.setbadchan.channels{1}.chanfile = '<UNDEFINED>';

```

Then save these edits (overwriting the previous batch file). If you're unsure, the batch file should look like the `batch_preproc_meeg_convert_job.m` file in the `SPM12batch` part of the `SPMscripts` directory on the FTP site.

This completes the first part of the preprocessing pipeline. You can then run this script by selecting the green play button on the upper toolbar of the script MATLAB Editor window. The results will be 6 files labelled `bdspmeeg_run_%02d_sss.mat`, where `%02d` refers to the run number 1-6. If you want to view any of these output files, press “Display” on the main SPM menu pane, select “M/EEG”, then select one of these files. You will be able to review the preprocessing steps as a pipeline from the “History” section of the “Info” tab, and can view single trials by selecting one of the EEG, MEG (magnetometer) or MPLANAR (gradiometer) tabs (see other chapters for how to use the buttons in the M/EEG Review window).

### 42.3.6 Merging (concatenating runs)

To analyse the data as one file, the six runs need to be merged. To do this, select “Merging” from “SPM – M/EEG – Preprocessing – Merging”, select “File Names”, “specify”, and select the 6 file names `bdspmeeg_run_%02d_sss.mat`. If you were to run this stage now, the output file would match the first input file, but be prepended with a `c`, i.e. `cbdspmeeg_run_01_sss.mat`. However, we will wait to add some more modules before running, as below. At this stage, you could also add delete modules to delete all the previous individual run files (since the concatenated file will contain all trials from all runs, i.e. contain the same data).

### 42.3.7 Prepare (a montage for re-referencing the EEG)

Below, we want to re-reference the EEG data to the average across channels (as is sometimes conventional for ERP analyses; note the MEG data have no reference). We can do this with the “montage” module below, which is a general purpose module for creating new channel data from linear combinations of existing channel data. However, we first need to create a montage file, which includes a matrix that, when multiplied by the existing data, creates the new channel data. There is another sub-function (task) of the “Prepare” module that does this, so add another “Prepare” module, select the dependency on the previous merged file as the “FileName”, but for the “task”, select “Create average reference montage” and enter `avref_montage.mat` as the output filename. (If you want to look at this montage, you can run this module, load `avref_montage.mat` into MATLAB and look at the `montage.tra` matrix, where you can see that each new EEG channel is equal to the old EEG channel minus the average of all other channels.) Note that this montage will differ across subjects because different subjects had different EEG channels marked as bad (from steps above), and bad channels need to be excluded when estimating the average EEG signal across channels.

At this point, we can also do one more bit of house-keeping within the same “Prepare” module, which is simply to re-order the condition labels. This only matters for the final stage of “Con-

trasting conditions” below, where the contrast weights assume a certain order of the conditions. The current order of conditions is based purely on the order they appear in the raw data (e.g., if the first few trials of the first run were: Scrambled, Unfamiliar, Unfamiliar, Scrambled, Famous..., then the condition labels will be ordered Scrambled-Unfamiliar-Famous), and this may vary across subjects. To set the condition order to be invariant across subjects, add a new task by selecting the “Sort conditions” task, then “Specify conditions lists” add three “New: Condition labels”, and name them “Famous”, “Unfamiliar” and “Scrambled” (in that order). Note that this operation does not physically reorder the trials at this stage, but just defines the order that will be used where required at later steps.

### 42.3.8 Montage

Now we have the montage file, we can apply it, in order to re-reference the EEG data to the average. Select “Montage” from the “Preprocessing” menu, and specify the “File Name” as being dependent on the output of the “Merging” module above. For the “Montage file name”, choose a different dependency, namely the output of the “Prepare” module above. Next, highlight “keep other channels” and select “yes” in the “Current Item” box, in order to keep all the MEG channels (which are unchanged). All other default values can remain the same. The output file will be prepended with M.

As with the previous pipeline, if you are short of diskspace (particularly if you later run all 16 subjects), the outputs produced from the intermediate stages can be deleted using the “SPM – M/EEG – Other – Delete” function (see earlier). However, refrain from deleting the montaged data, as these will be used later in the Section on Time-Frequency analysis.

**Save batch and review.** This completes the second part of the preprocessing pipeline. At this point, you can run the batch. Alternatively, you can save and script, and run the batch from a script. The resulting script can also be combined with the previous script created (e.g., in the `SPMscripts` FTP directory, scripts for all the stages are appended into a single master script called `master_script.m`, which loops over each subject too). Remember that, if you want to pass variables from a script to a batch, you need to first ensure the relevant fields in the batch file are set to '`<UNDEFINED>`' (see for example the `batch_preproc_meeg_merge_job.m` file in the `SPM12batch` FTP directory).

To view the output, press “Display” on the main SPM menu pane, select “M/EEG”, then select `Mcbdspmeeg_run_01_sss.mat`. Again, you will be able to review the preprocessing steps from the “History” section of the “Info” tab.

## 42.4 Evoked analysis

At this point, the preprocessing forks into two strands: one for trial-averaged amplitude analysis and one for time-frequency analysis. The first of these corresponds to a typical evoked response (ER) analysis, where we simply average across trials in each condition (note this will attenuate any non-phase locked, i.e. induced responses; to detect these, we will later perform time-frequency analysis before averaging). Before averaging though, we will crop the 400ms buffer around each trial (which is only necessary for the time-frequency analysis).

### 42.4.1 Crop

To crop the data, select the crop option from “SPM – M/EEG – Preprocessing – Crop”. Select the datafile called `Mcbdespmeeg_run_01_sss.mat` produced from the final EEG re-referencing (“Montage”) step above. A 100ms pre-stimulus baseline period is normally sufficient, and we do not care about responses after 800ms for this particular analysis, so we can cut off 400ms at the start and end of each trial to produce an epoch from -100ms to +800ms. To do this, select “Time Window” from the “Current Module” window, then the “Specify” button. Within the pop up window, enter `[-100 800]` as a 1-by-2 array. The channel selection will be “all”. This file output will be prepended with a `p`.

### 42.4.2 Artefact detection

There are many ways to define artifacts (including special toolboxes; see other SPM manual chapters). Here we focus on just one simple means of detecting blinks by thresholding the EOG channels. Select “Artefact detection” from the “SPM – M/EEG – Preprocessing” menu. For the input file, select a dependency on the output of the previous step (“Crop”). Next, select “New: Method” from the box titled “Current Item: How to look for artefacts”. Back in the “Current Module” window, highlight “Channel selection” to list more options, choose “Select channels by type” and select “EOG”. Then do not forget to also delete the default “All” option! Then press the “<-X” to select “threshold channels”, click the “Specify” button and set this to 200 (in units of microvolts). The result of this thresholding will be to mark a number of trials as “bad” (these can be reviewed after the pipeline is run if you like). Bad trials are not deleted from the data, but marked so they will be excluded from averaging below. The output file will be prepended with the letter “a”.

### 42.4.3 Combine Planar Gradiometers

The next step is only necessary for scalp-time statistics on planar gradiometers. For scalp-time images, one value is needed for each sensor location. Neuromag’s planar gradiometers measure two orthogonal directions of the magnetic gradient at each location, so these need to be combined into one value for a scalar (rather than vector) topographic representation. The simplest way to do this is to take the Root Mean Square (RMS) of the two gradiometers at each location (i.e. estimate the 2D vector length). In SPM, this will create a new sensor type called MCOMB. Note that this step is NOT necessary for source reconstruction (where, the forward model captures both gradiometers). Note also that the RMS is a nonlinear operation, which means that zero-mean additive noise will no longer cancel by averaging across trials, in turn meaning that it is difficult to compare conditions that differ in the number of trials. To take the RMS, select “Combine Planar” from the “SPM – M/EEG – Preprocessing” menu, highlight “File Name”, select the “dependency” button, and choose the Artefact-corrected file above. Leave the “copying mode” as default – “Replace planar”. The produced file will be prepended with P.

### 42.4.4 Trial averaging

To average the data across trials, select “SPM – M/EEG – Average – Averaging”, and define the input as dependent on the output of the planar combination module. Keep the remaining options as the default values. (If you like, you could change the type of averaging from “standard” to “Robust”. Robust averaging is a more sophisticated version of normal averaging, where each timepoint in each trial is weighted according to how different it is from the median across trials. This can be a nice feature of SPM, which makes averaging more robust to atypical trials, though in fact it does not make much difference for the present data, particularly given the large numbers of trials, and we do not choose it here simply because it takes much longer than conventional averaging.) Once completed, this file will have a prefix of m.

### 42.4.5 Contrasting conditions

We can also take contrasts of our trial-averaged data, e.g., to create a differential ER between faces and scrambled faces. This is sometimes helpful to see condition effects, and plot their topography. These contrasts are just linear combinations of the original conditions, and so correspond to vectors with 3 elements (for the 3 conditions here). Select “SPM – M/EEG – Average – Contrast over epochs”, and select the output of “Averaging” above as in the dependent input. You can then select “New Contrast” and enter as many contrasts as you like. The resulting output file is prepended with w.

For example, to create an ER that is the difference between faces (averaged across Famous and Unfamiliar) and scrambled faces, enter the vector  $[0.5 \ 0.5 \ -1]$  (assuming conditions are ordered Famous-Unfamiliar-Scrambled; see comment earlier in “Prepare” module), and give it a name. Or to create the differential ER between Famous and Unfamiliar faces, enter the vector  $[1 \ -1 \ 0]$ . Sometimes it is worth repeating the conditions from the previous averaging step by entering, in this case, three contrasts:  $[1 \ 0 \ 0]$ ,  $[0 \ 1 \ 0]$  and  $[0 \ 0 \ 1]$ , for Famous, Unfamiliar

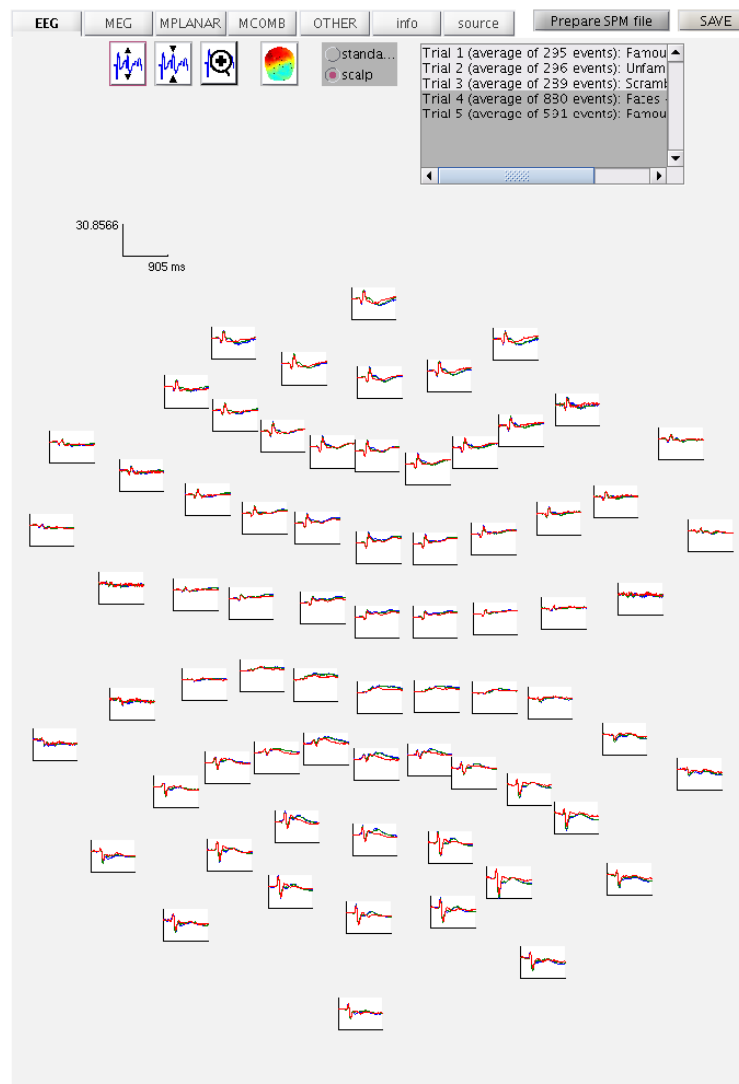


Figure 42.2: Trial-averaged ERPs for each condition over all EEG channel positions on the scalp.

and Scrambled conditions respectively. These will be exactly the same as in the averaged file above, but now we can examine them, as well as the differential responses, within the same file (i.e. same graphics window when we review that file), and so can also delete the previous `m` file.

**Save batch and review.** At this point, you can save batch and script again. The resulting batch file should look like the `batch_preproc_meeeg_erp_job.m` file in the `SPM12batch` part of the `SPMscripts` FTP directory. The script file can be run (and possibly combined with the previous script created).

We will start by looking at the trial-averaged ERs to each of the three conditions. Select the “Display” button on the SPM Menu and select the file `wmPapMcbdspmeeg_run_01_sss.mat`. Then select, for example, the “EEG” tab, and you will see each channel as a row (“strip”, or “standard view”) for the mean ER for Famous faces. If you press “scalp” instead, the channels will be flat-projected based on their scalp position (nose upwards). You can now display multiple conditions at once by holding the shift-key and selecting Trials 2 and 3 (Unfamiliar and Scrambled) as well (as in Figure 42.2, after zooming the y-axis slightly). If you press the expand y-axis button (top left) a few times to up-scale the data, you should see something like in Figure 42.2. You can see the biggest evoked potentials (relative to average over channels) at the back of the head.

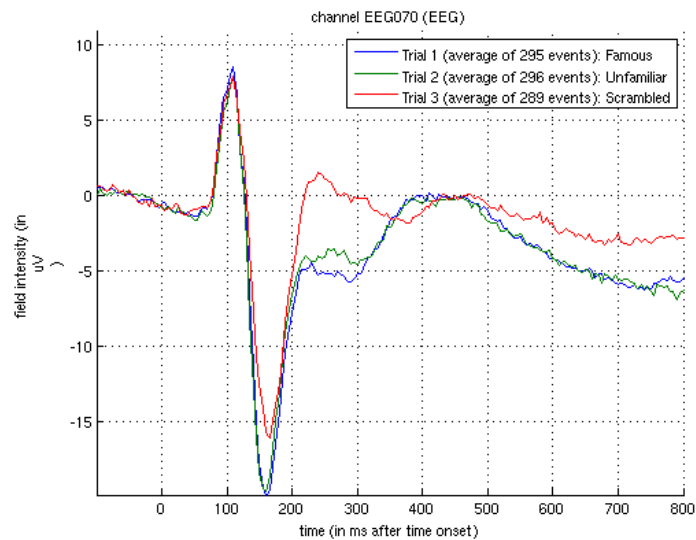


Figure 42.3: Trial-averaged ERPs for each condition from EEG channel 70 (right posterior).

If you press the magnifying glass icon, then with the cross-hairs select Channel 70 (in bottom right quadrant of display), you will get a new figure like in Figure 42.3 that shows the ERPs for that channel in more detail (and which can be adjusted using the usual MATLAB figure controls). You can see that faces (blue and green lines) show a more negative deflection around 170ms than do scrambled faces (red line), the so-called “N170” component believed to index the earliest stage of face processing.

To see the topography of this differential N170 component, select instead the fourth trial (contrast) labelled “Faces – Scrambled”. Then press the coloured topography icon, and you will get a new figure with the distribution over the scalp of the face-scrambled difference. If you shift the time-slider on the bottom of that window to the leftmost position, and then repeatedly click on the right arrow, you will see the evolution of the face effect, with no consistent difference during the prestimulus period, or until about 155ms, at which point a clear dipolar field pattern should emerge (Figure 42.4).

You can of course explore the other sensor-types (magnetometers, MEG) and combined gradiometers (MCOMB), which will show an analogous “M170”. You can also examine the EOG and ECG channels, which appear under the “OTHER” tab. (Note that the VEOG channel contains a hint of an evoked response: this is not due to eye-movements, but due to the fact that bipolar channels still pick up a bit of brain activity too. The important thing is that there is no obvious EOG artefact associated with the difference between conditions, such as differential blinks.)

But how do we know whether this small difference in amplitude around 150-200ms is reliable, given the noise from trial to trial? And by looking at all the channels and timepoints, in order to identify this difference, we have implicitly performed multiple comparisons across space and time: so how do we correct for these multiple comparisons (assuming we had no idea in advance where or when this face-related response would occur)? We can answer these questions by using random field theory across with scalp-time statistical parametric maps. But first, we have to convert these sensor-by-time data into 3D images of 2D-location-by-time.

#### 42.4.6 Time-Sensor images

To create 3D scalp-time images for each trial, the 2D representation of the scalp is created by projecting the sensor locations onto a plane, and then interpolating linearly between them onto a  $32 \times 32$  pixel grid. This grid is then tiled across each timepoint. To do this, you need to select the “SPM – M/EEG – Images – Convert2Images” option in the batch editor. For the input file, select the `PapMcbdspmeeg_run_01_sss.mat` file that contains every cropped trial (i.e, before averaging),

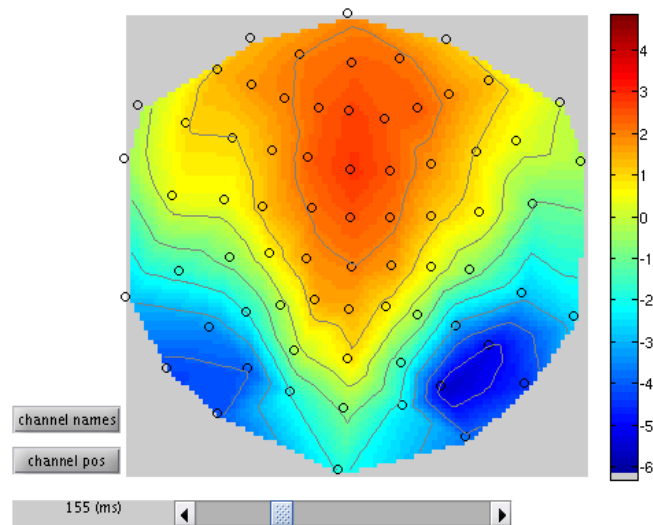


Figure 42.4: *Topography of differential ERP for faces (famous and unfamiliar) vs scrambled at 155ms.*

but with bad trials marked (owing to excessive EOG signals; see earlier). Next select “Mode”, and select “scalp x time”. Then, select “conditions”, select “Specify” and enter the condition label “Famous”. Then repeat for the condition labels “Unfamiliar” and “Scrambled”.

To select the channels that will create your image, highlight the “Channel selection”, and then select “New: Select channels by type” and select “EEG”. The final step is to name the Directory prefix `eeg_img_st` this can be done by highlighting “directory prefix”, selecting “Specify”, and the prefix can then be entered.

This process can be repeated for the MEGMAG channels, and the MEGCOMB channels (although we will focus only on the EEG here). If so, the easiest way to do this is to right-click “Convert2Images” in the Module List, and select “replicate module”. You will have to do this twice, and then update the channels selected, and the directory prefix to `mag_img_mat` and `grm_img_mat` to indicate the magnetometers (MEGMAG) and the gradiometers (MEGCOMB) respectively.

**Save batch and review.** At this point, you can save batch and script again. The resulting batch file should look like the `batch_preproc_meeeg_erp_images_job.m` file in the `SPM12batch` FTP directory. Once you have run this script, a new directory will be created for each channel-type, which is based on the input file and prefix specified above (e.g., `eeg_img_st_PapMcbdspmeeg_run_01_sss` for the EEG data). Within that directory will be three 4D NIFTI files, one per condition. It is very important to note that these 4D files contain multiple “frames” (i.e. 3D scalp-time images), one per trial (i.e. 296 in the case of unfamiliar faces). To view one of these, press “Display – Images” in the SPM Menu window, and select, say, the `condition_Unfamiliar.nii` file. But note that by default you will only see the first scalp-time image in each file (because the default setting of “Frames” within the Select Image window is 1). To be able to select from all frames, change the “Frames” value from 1 to `Inf` (infinite), and now you will see all 296 frames (trials) that contained Unfamiliar faces. If you select, say, number 296, you should see an image like in Figure 42.5 (this was created after selecting “Edit – Colormap” from the top toolbar, then “Tools – Standard Colormap – Jet”, and entering `[0 0 165]` as the coordinates in order to select 165ms post-stimulus). You can scroll with the cross-hair to see the changes in topography over time.

Note that Random Field Theory, used to correct the statistics below, assumes a certain minimum smoothness of the data (at least three times the voxel size). The present data meet this requirement, but in other cases, one could add an additional step of Gaussian smoothing of the

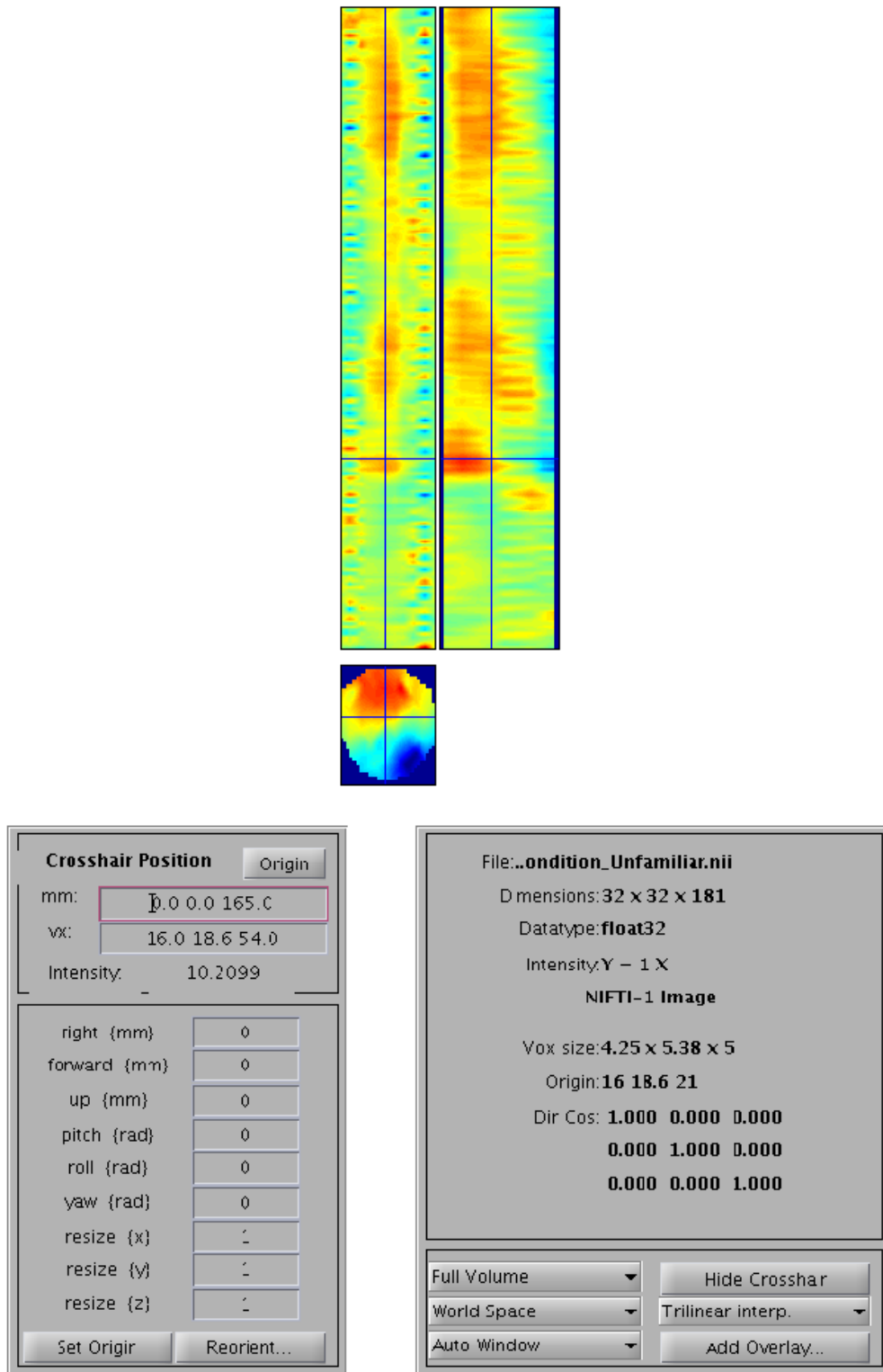


Figure 42.5: 3D Scalp-Time image for 296th trial in the Unfamiliar condition.



images to ensure this smoothness criterion is met.

## 42.5 Scalp-Time Statistics across trials within one subject

Now we have one image per trial (per condition), we can enter these into a GLM using SPM's statistical machinery (as if they were fMRI or PET images). If we ignore temporal autocorrelation across trials, we can assume that each trial is an independent observation, so our GLM corresponds to a one-way, non-repeated-measures ANOVA with 3 levels (conditions).

### 42.5.1 Model Specification

To create this model, open a new batch, select "Factorial design specification" under "Stats" on the SPM toolbar at the top of the batch editor window. The first thing is to specify the output directory where the SPM stats files will be saved. So first create such a directory within the subject's sub-directory, calling it for example `STStats`, and then create a sub-directory `eeg` within `STStats` (and potentially two more called `mag` and `grm` if you want to examine other sensor-types too). Then go back to the batch editor and select this new `eeg` directory.

Highlight "Design" and from the current item window, select "One-way ANOVA". Highlight "Cell", select "New: Cell" and repeat until there are three cells. Select the option "Scan" beneath each "Cell" heading (identified by the presence of a "<-X"). Select "Specify", and in the file selector window, remember to change the "Frames" value from 1 to `Inf` as previously to see all the trials. Select all of the image files for one condition (by using the right-click "select all" option). It is vital that the files are selected in the order in which the conditions will later appear within the Contrast Manager module (i.e., Famous, Unfamiliar, Scrambled). Next highlight "Independence" and select "Yes", but set the variance to "Unequal". Keep all the remaining defaults (see other SPM chapters for more information about these options).

Finally, to make the GLM a bit more interesting, we will add 3 extra regressors that model the effect of time within each condition (e.g. to model practice or fatigue effects). (This step is optional if you'd rather omit.) Press "New: Covariate" under the "Covariates" menu, and for the "Name", enter "Order Famous". Keep the default "None" to interactions, and "Overall mean" for "centering". We now just need to enter a vector of values for every trial in the experiment. These trials are ordered Famous, Unfamiliar and Scrambled, since this is how we selected them above. So to model linear effects of time within Famous trials, we need a vector that goes from 1:295 (since there are 295 Famous trials). However, we also need to mean-correct this, so we can enter `detrend([1:295],0)` as the first part of the vector (covariate) required. We then need to add zeros for the remaining Unfamiliar and Scrambled trials, of which there are  $296+289=585$  in total. So the complete vector we need to enter (for the Famous condition) is `[detrend([1:295],0) zeros(1,585)]`. We then need to repeat this time covariate for the remaining two conditions. So press "New: Covariate" again, but this time enter "Order Unfamiliar" as the name, and `[zeros(1,295) detrend([1:296],0) zeros(1,289)]` as the vector. Finally, press "New: Covariate", but this time enter "Order Scrambled" as the name, and `[zeros(1,591) detrend([1:289],0)]` as the vector.

This now completes the GLM specification, but before running it, we will add two more modules.

### 42.5.2 Model Estimation

The next step within this pipeline is to estimate the above model. Add a module for "Model Estimation" from the "Stats" option on the SPM toolbar and define the file name as being dependent on the results of the factorial design specification output. For "write residuals", keep "no". Select classical statistics.

### 42.5.3 Setting up contrasts

The final step in the statistics pipeline is create some planned comparisons of conditions by adding a "Contrast Manager" module from the "Stats" bar. Define the file name as dependent on the

model estimation. The first contrast will be a generic one that tests whether significant variance is captured by the 6 regressors (3 for the main effect of each condition, and 3 for the effects of time within each condition). This corresponds to an F-contrast based on a 6x6 identity matrix. Highlight contrast sessions and select a new F-contrast session. Name this contrast “All Effects”. Then define the weights matrix by typing in `eye(6)` (which is MATLAB for a 6x6 identity matrix). (Since there is only one “session” in this GLM, select “Don’t replicate” from the “replicate over sessions” question.) We will use this contrast later to plot the parameter estimates for these 6 regressors.

More interestingly perhaps, we can also define a contrast that compares faces against scrambled faces (e.g. to test whether the N170 seen in the average over trials in right posterior EEG channels in Figure 42.3 is reliable given the variability from trial to trial, and to also discover where else in space or time there might be reliable differences between faces and scrambled faces). So make another F-contrast, name this one “Faces (Fam+ Unf) <> Scrambled”, and type in the weights `[0.5 0.5 -1 0 0 0]` (which contrasts the main effect of faces vs scrambled faces, ignoring any time effects (though SPM will complete the final zeros if you omit). Note that we use an F-test because we don’t have strong interest in the polarity of the face-scrambled difference (whose distribution over the scalp depends on the EEG referencing). But if we did want to look at just positive and negative differences, you could enter two T-contrasts instead, with opposite signs on their weights.

**Save batch and review** Once you have added all the contrasts you want, you can save this batch file (it should look like the `batch_stats_ANOVA_job.m` file in the `SPM12batch` FTP directory). This only runs a GLM for one sensor-type (we cannot combine the sensors until we get to source space later), so you can write a script around this batch that calls it three times, once per sensor-type (i.e. for magnetometers and gradiometer RMS too), just changing the output directory and input files (see `master_script.m` on the `SPM12batch` FTP directory).

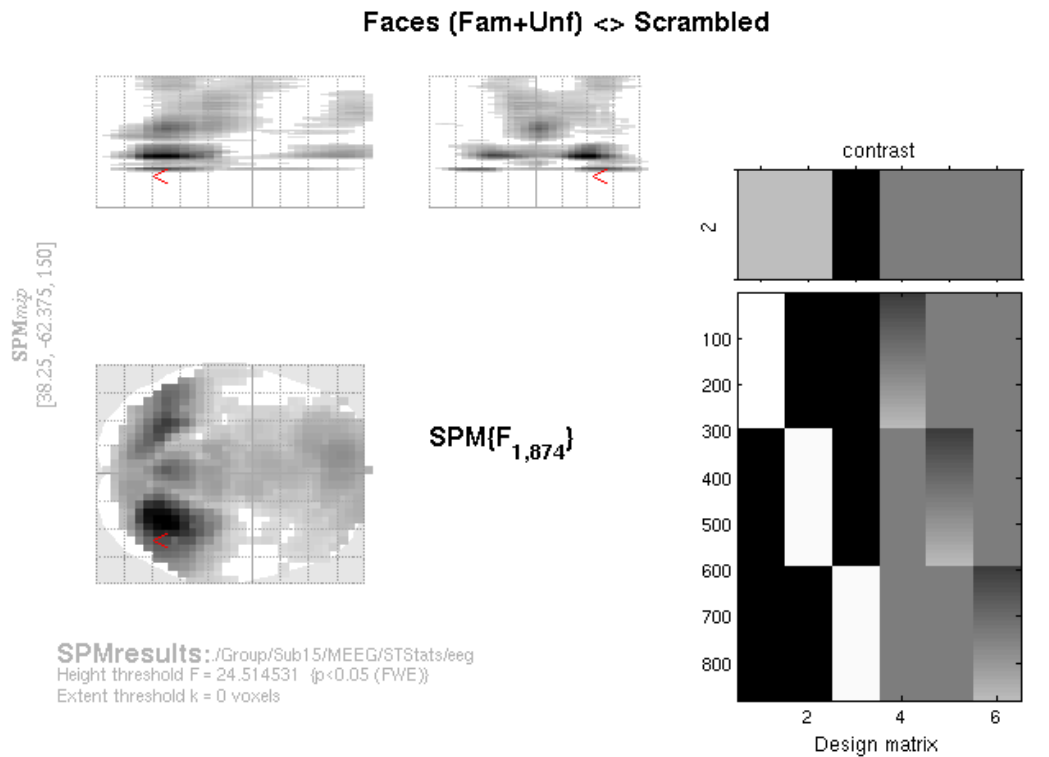
The results of this output can be viewed by selecting “Results” from the SPM Menu window. Select the `SPM.mat` file in the `STStats/eeg` directory, and from the new “Contrast Manager” window, select the pre-specified contrast “Faces (Fam+Unf) <> Scrambled”. Within the Interactive window which will appear on the left hand side, select the following: Apply Masking: None, P value adjustment to control: FWE, keep the threshold at 0.05, extent threshold {voxels}: 0; Data Type: Scalp-Time. The Graphics window should then show what is in Figure 42.6.

If you move the cursor to the earliest local maximum – the third local peak in the first cluster – this corresponds to  $x=+38\text{mm}$ ,  $y=-62\text{mm}$  and  $t=150\text{ms}$  (i.e. right posterior scalp, close to the sensor shown in Figure 42.3, though note that distances are only approximations). If you then press “Plot – Contrast Estimates – All Effects”, you will get 6 bars like in Figure 42.7. The first three reflect the three main conditions (the red bar is the standard error from the model fit). You can see that Famous and Unfamiliar faces produce a more negative amplitude at this space-time point than Scrambled faces (the “N70”). The next three bars show the parameter estimates for the modulation of the evoked response by time. These effects are much smaller relative to their error bars (i.e., less significant), but suggest that the N170 to Famous faces becomes less negative with time, and that to scrambled faces becomes larger (though one can test these claims formally with further contrasts).

There are many further options you can try. For example, within the bottom left window, there will be a section named “Display”, in the second drop-down box, select “Overlay – Sections” and from the browser, select the `mask.nii` file in the analysis directory. You will then get a clearer image of suprathreshold voxels within the scalp-time-volume. Or you can of course examine other contrasts, such as the difference between famous and unfamiliar faces, which you will see is a much weaker and slightly later effect.

## 42.6 Time-Frequency Analysis (Evoked and Induced power)

The above statistical test only identifies significant evoked effects (i.e., that are phase-locked across trials) within one individual. We can also look at induced as well as evoked energy by performing a time-frequency analysis before averaging over trials. Here we will use Morlet wavelets to decompose each trial into power and phase across peristimulus time and frequency. We will



**Statistics: p-values adjusted for search volume**

set-level		cluster-level				peak-level					mm mm ms	
D	C	D <sub>FWE-corr</sub>	q <sub>FDR-corr</sub>	k <sub>E</sub>	D <sub>uncorr</sub>	D <sub>FWE-corr</sub>	q <sub>FDR-corr</sub>	F	(Z <sub>≡</sub> )	D <sub>uncorr</sub>		
0.000	31	0.000	0.000	4463	0.000	0.000	0.000	174.46	Inf	0.000	30	-57 240
						0.000	0.000	132.06	Inf	0.000	-30	-57 245
						0.000	0.000	123.95	Inf	0.000	38	-62 150
		0.000	0.000	6539	0.000	0.000	0.000	109.95	Inf	0.000	0	-57 425
						0.000	0.000	100.30	Inf	0.000	0	-57 450
						0.000	0.000	95.90	Inf	0.000	0	-57 410
		0.000	0.000	345	0.000	0.000	0.000	106.49	Inf	0.000	-42	-57 145
		0.000	0.000	2624	0.000	0.000	0.000	77.41	Inf	0.000	-8	56 240
						0.000	0.000	71.57	Inf	0.000	-8	50 260
						0.000	0.000	70.38	Inf	0.000	4	50 245
		0.000	0.000	614	0.000	0.000	0.000	60.22	7.54	0.000	-4	8 150
						0.000	0.000	57.47	7.37	0.000	13	2 150
						0.000	0.000	56.49	7.30	0.000	0	24 150
		0.000	0.000	289	0.000	0.000	0.000	59.24	7.48	0.000	0	-57 190
						0.000	0.001	42.00	6.30	0.000	0	-30 195
						0.000	0.006	36.91	5.90	0.000	-8	-25 205
		0.000	0.000	929	0.000	0.000	0.000	56.73	7.45	0.000	42	-57 230
						0.000	0.000	54.66	7.19	0.000	38	-57 785
						0.000	0.000	54.32	7.16	0.000	38	-52 745
		0.000	0.000	1307	0.000	0.000	0.000	56.26	7.42	0.000	21	67 535
						0.000	0.000	51.95	7.01	0.000	17	67 525
						0.000	0.000	51.74	6.99	0.000	21	67 575
		0.000	0.000	136	0.000	0.000	0.006	37.02	5.91	0.000	34	50 405
						0.001	0.017	34.40	5.69	0.000	47	34 410
						0.001	0.023	33.75	5.63	0.000	42	8 415
		0.000	0.014	25	0.005	0.000	0.013	35.10	5.75	0.000	36	-57 600
						0.007	0.170	28.84	5.20	0.000	38	-57 530
		0.001	0.030	19	0.012	0.001	0.017	34.54	5.70	0.000	-60	-19 465
		0.000	0.006	32	0.002	0.001	0.021	33.96	5.65	0.000	36	-57 630

table shows 3 local maxima more than 8.0mm apart

Height threshold: F = 24.51, p = 0.000 (0.050)  
 Extent threshold: k = 0 voxels  
 Expected voxels per cluster, <k> = 2.699  
 Expected number of clusters, <c> = 0.05  
 FWEp: 24.515, FDRp: 31.832, FWEc: 1, FDRc: 19

Degrees of freedom = [1 0, 874.0]  
 FWHM = 24.8 36.1 12.2 mm mm ms; 5.8 6.7 2.4 (voxels)  
 Volume: 16559549 = 144981 voxels = 1390.7 resels  
 Voxel size: 4.2 5.4 5.0 mm mm ms; (resel = 95.52 voxels)  
 Page 7



Figure 42.6: Scalp-Time SPM for F-contrast, thresholded at p < .05 FWE corrected, for faces vs scrambled faces across trials for one subject.

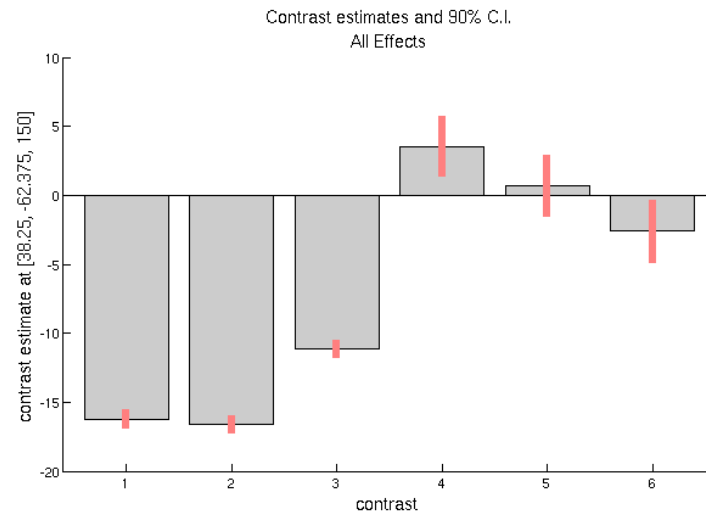


Figure 42.7: *Effects of interest from sub-peak +38mm -62mm +150ms. First three bars are mean evoked response amplitude vs baseline for Famous, Unfamiliar and Scrambled faces; next three bars are modulations about mean by time throughout experiment.*

then average the power estimate across trials and across channels, to create a 2D time-frequency image, which we can then test for reliability, but now across subjects rather than across trials. This is the second fork in the preprocessing pipeline.

### 42.6.1 Wavelet estimation

The first module to be added to the pipeline is the “Time-Frequency Analysis” under “SPM – M/EEG – Time-frequency”. Highlight “File Name”, choose “Specify” box, and select the file `Mcbdspmeeg_run_01_sss.mat`. This file has not yet been cropped because we need the 400ms buffer to reliably estimate the lower-frequency wavelets (in fact, for a 5-th order 6Hz wavelet, one needs 2.5 periods =  $2.5 \times 133\text{ms} \sim 400\text{ms}$  to reliably estimate power at the central timepoint). Note that this file does not have artifacts removed, but most of these identified for the evoked analysis above are blinks, which are typically below  $\sim 3\text{Hz}$ , i.e., below the lowest frequency of interest here.

Next, ensure “channel selection” is set to “all”. Then highlight the “Frequencies of interest”, click “Specify” and enter 6:40 (which will estimate frequencies from 6 to 40Hz in steps of 1Hz). Then highlight “Spectral estimation” and select “Morlet wavelet transform”. Then highlight “Number of wavelet cycles” and “Specify” this as 5. The fixed time window length should be set at 0 as default. To reduce the size of the file, we will also subsample in time by a factor of 5 (i.e., every 25ms, given the additional downsampling to 5ms done earlier in preprocessing). Do this by selecting “subsample” and “Specify” as 5. Finally, select “Yes” for “Save phase”. This will produce two files, one for power (prefixed with `tf`) and one for phase (prefixed with `tph`).

### 42.6.2 Crop

Once we have estimated power and phase at every peristimulus timepoint, we can cut the 400ms buffer. So add the “SPM – M/EEG – Preprocessing – Crop” module (like for the evoked analysis). This module will have to be added twice though, once for power and once for phase. For the first module, select the power file as the dependency, and enter `[-100 800]` as the timewindow. Then select “replicate module”, and change only the input dependency to the phase file.

### 42.6.3 Average

As with cropping, we now need to add two averaging modules, one with dependency on the power file from the Crop step, and the other with dependency on the phase file from the Crop step. For the power averaging, choose straight averaging. For the phase averaging however, we need to additionally select “Yes” to the “Compute phase-locking value”. This corresponds to “circular” averaging, because phase is an imaginary number. This produces a quantity called the phase-locking value (PLV). The prefix will be *m* in both cases.

### 42.6.4 Baseline rescaling

To baseline correct time-frequency data, we need to select the “Time-Frequency Rescale” option from the “Time-Frequency” menu (note: not the “Baseline Correction” module from the “Pre-processing” menu). There are several options to baseline-correct time-frequency data. For the power, it also helps to scale the data with a log transform, because changes at high-frequency tend to be much smaller than changes at lower frequencies. We will therefore use the log-ratio (“LogR”) option, where all power values at a given frequency are divided by the mean power from -100 to 0ms at that frequency, and the log taken (which is equivalent to subtracting the log of the baseline power). So select the power file output from the previous phase as the dependency, select the log ratio option, and enter the baseline time window as -100 0. The output file prefix will be *r*. We won’t bother to baseline-correct the phase-data.

### 42.6.5 Contrasting conditions

Finally, like with the evoked fork, we can take contrasts of our trial-averaged data, e.g., to create a time-frequency image of the difference in power, or in PLV, between faces and scrambled faces. Create two “SPM – M/EEG – Average – Contrast over epochs” modules, one with the average power file as input, and one with the averaged phase (PLV) file as input. You can then select “New Contrast” and enter contrasts like [0.5 0.5 -1] (for faces vs scrambled; see earlier) and [1 -1 0] (for famous vs unfamiliar). The resulting output file is prepended with *w*.

As before, if you want to save file-space, you can add further “Delete” modules, since we will not need many of the intermediate files. The only files we need to keep are the averaged power and phase files (since these are used to create time-frequency images below) and the contrasted versions (for visual inspection of effects within each subject).

**Save batch and review.** You can now save this time-frequency batch file (it should look like the `batch_preproc_meeeg_tf_job.m` file in the `SPM12batch` FTP directory). Once you have run it, you can then review the contrast files, e.g. for power (`wmprtf_Mcbdspmeeg_run_01_sss.mat`) or phase (`wmptph_Mcbdspmeeg_run_01_sss.mat`). When displaying the power for the EEG data, if you magnify Channel 70 (right posterior, as in Figure 42.3), you should see something like that in Figure 42.8. The increase in power from 13 to 16Hz between 75 and 200ms is most likely the evoked energy corresponding to the N170 in Figure 42.3, but additional power changes can be seen that may not be so phase-locked (i.e. induced). This is supported by looking at the difference in PLV for that channel (Figure 42.9), where faces increase phase-locking relative to scrambled faces in a similar time-frequency window (suggesting a change in phase of ongoing alpha oscillations, as well as their power).

### 42.6.6 Creating 2D time-frequency images

Later we will repeat the above steps for every subject, and create a statistical parametric map of power changes across subjects, in order to localise face-induced power changes in frequency and time. This requires that we create 2D images, one per condition (per subject). Select the “convert2images” option and select the baseline-rescaled, trial-averaged power file as the dependency from the stage above. Select the mode “time x frequency”. Note that this implicitly means averaging across all channels. Next select channel selection by type. Here, choose EEG for “Channel selection” and define the prefix as `eeg_img_pow`. (Of course, this module can be repeated for MEGMAG and MEGPLANAR sensors if wanted.) You can save this batch, run it, and display the images output if you like (the FTP batch is called `batch_preproc_meeeg_tf_images_job.m`).

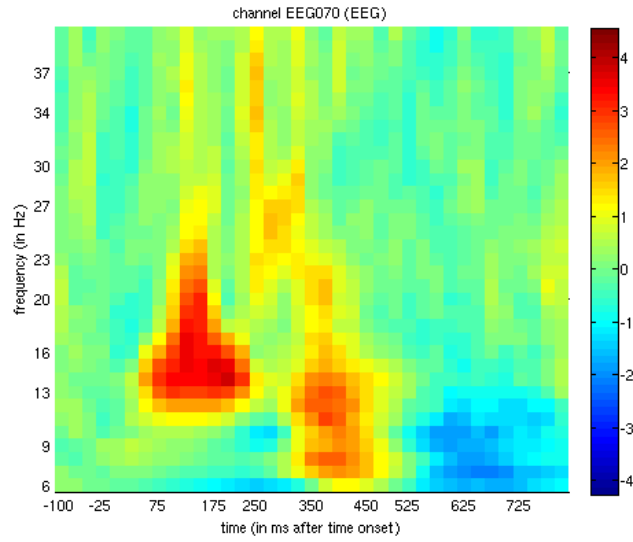


Figure 42.8: *Trial-averaged power for faces > scrambled in EEG channel 70 (right posterior).*

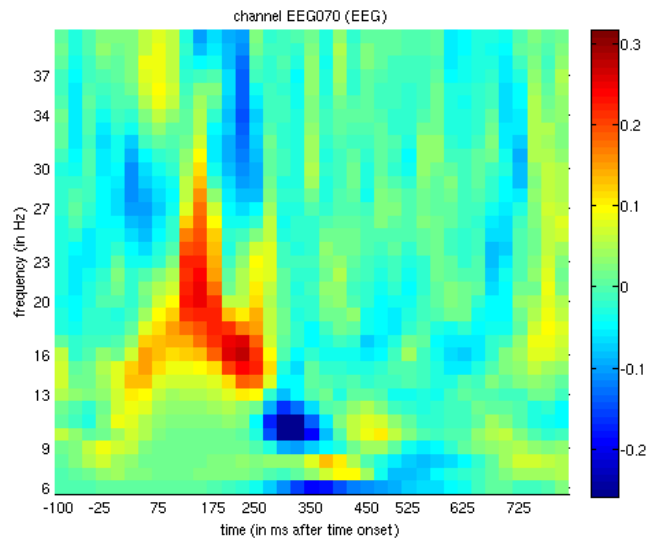


Figure 42.9: *Trial-averaged PLV for faces > scrambled in EEG channel 70 (right posterior).*

**Creating a script for analysis across subjects.** Now that we have created pipelines for various steps of preprocessing one subject, including the two forks for evoked and time-frequency analyses, we want to run these on the remaining 15 subjects. For this, a bit of MATLAB knowledge is required to call these pipelines within a `for` (or `parfor`) loop across subjects. Below is an example from the `master_script.m`:

```

for s = 1:nsub

    %% Change to subject's directory
    swd = fullfile(outpth,subdir{s},'MEEG');
    cd(swd);

    %% Convert & epoch, prepare, downsample, baseline-correct each run
    jobfile = {fullfile(scrpth,'batch_preproc_meeg_convert_job.m')};
    jobs = repmat(jobfile, 1, nrun);
    n = 1;
    inputs = cell(nrun*3, 1);
    for r = 1:nrun
        inputs{n,1} = cellstr(fullfile(rawpth,subdir{s},'MEEG',sprintf('run_%02d_sss.fif',r)));
        inputs{n+1,1} = cellstr(fullfile(rawpth,subdir{s},'MEEG','Trials',sprintf('run_%02d_trldef.mat',r)));
        inputs{n+2,1} = cellstr(fullfile(rawpth,subdir{s},'MEEG','bad_channels.mat'));
        n = n + 3;
    end
    spm_jobman('serial', jobs, '', inputs{:});

    %% Concatenate runs and montage (reref EEG)
    jobfile = {fullfile(scrpth,'batch_preproc_meeg_merge_job.m')};
    inputs = cell(3, 1);
    inputs{1} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^bdbspmeeg.*\.mat$'));
    inputs{2} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^bdbspmeeg.*\.mat$')); %
        (For deletion)
    spm_jobman('serial', jobfile, '', inputs{:});

    %% Fork 1. For ERP/ERF: crop to -100 to +800, detect artifacts (blinks) by thresholding EOG,
        combine planar grads, average over trials and create contrasts of conditions
    jobfile = {fullfile(scrpth,'batch_preproc_meeg_erp_job.m')};
    inputs = cell(1);
    inputs{1} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^Mcbdspmeeg.*\.mat$'));
    spm_jobman('serial', jobfile, '', inputs{:});

    %% Write out scalp-time images
    if strcmp(subdir{s},'Sub15')
        jobfile = {fullfile(scrpth,'batch_preproc_meeg_erp_images_job.m')};
        inputs = cell(3,1);
        inputs{1} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^PapMcbdspmeeg.*\.mat$')
            );
        inputs{2} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^PapMcbdspmeeg.*\.mat$')
            );
        inputs{3} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^PapMcbdspmeeg.*\.mat$')
            );
        spm_jobman('serial', jobfile, '', inputs{:});
    end

    %% Fork 2. For Time-freq: Morlet wavelet transform, crop, average, baseline-correct (power) and
        contrast
    jobfile = {fullfile(scrpth,'batch_preproc_meeg_tf_job.m')};
    inputs = cell(1);
    inputs{1} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^Mcbdspmeeg.*\.mat$'));
    spm_jobman('serial', jobfile, '', inputs{:});

    %% Write out power and phase images for each modality
    jobfile = {fullfile(scrpth,'batch_preproc_meeg_tf_images_job.m')};
    inputs = cell(6,1);
    inputs{1} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^mprtf_Mcbdspmeeg.*\.mat$')
        ');
    inputs{2} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^mprtf_Mcbdspmeeg.*\.mat$')
        ');
    inputs{3} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^mprtf_Mcbdspmeeg.*\.mat$')
        ');

```



```

inputs{4} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^mptph_Mcbdspmeeg.*\.mat
    $'));
inputs{5} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^mptph_Mcbdspmeeg.*\.mat
    $'));
inputs{6} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^mptph_Mcbdspmeeg.*\.mat
    $'));
spm_jobman('serial', jobfile, '', inputs{:});

end

```

**Time-frequency Stats across Subjects** We can now enter the frequency-time power images (one per condition per subject) into a group (2nd-level) analysis that corresponds to repeated-measures ANOVA. Note that one difference from the ANOVA used for scalp-time analysis across trials within one subject above is that we now have 3 observations (conditions) from each subject, so need to correct for the potential correlation (nonsphericity) of the error induced by these repeated measurements. We can also add another 16 columns into our GLM that serve to remove between-subject variance (which we don't care about), improving the sensitivity of contrasts of our conditions of interest.

The first thing is to specify the output directory where the SPM stats files will be saved. Because this is now an analysis across subjects, we can create a directory above `Sub15` in the directory tree. So create a new top-level directory called, for example, "MEEG" (because later we will have group stats for the fMRI data and source-reconstructed MEEG data too), then a sub-directory within it called `TFStats`, and then a further sub-directory called `PowStats` and a further sub-directory called `eeg` (and potentially two more called `mag` and `grd` if you want to examine other sensor-types too).

### 42.6.7 Model Specification, Estimation and Contrasts

To create this repeated-measures ANOVA, open a new batch and select "Factorial design specification" under "Stats" on the "SPM" toolbar. Then select this new `MEEGTFStats/PowStats/eeg` directory.

Highlight "Design" and from the current item window, select "One-way ANOVA –within subject" (somewhat confusingly, this is not an analysis within one subject, as above, but an analysis in which multiple measures come from "within" each subject!). Highlight "Subjects" and create a "New:subject". In the "scans" field, you can now select the 3 power images for the first subject (which should have been created in the `Sub01/MEEG/eeg_img_pow_mprtf_Mcbdspmeeg_run_01_sss` directory if you ran the script above), and enter the "Conditions" as [1 2 3]. It is important for the contrasts below that you select the files in the order Famous-Unfamiliar-Scrambled (or if not, that you change the order of 1-3 in the Conditions vector, such that Condition 1 is Famous, Condition 2 Unfamiliar, etc.). You can then select "Replicate: Subject" under the "Subjects" item, keeping the "Conditions" unchanged, but changing the "Scans" to those in `Sub02/MEEG/eeg_img_pow_mprtf_Mcbdspmeeg_run_01_sss`. You can then repeat these steps for the remaining subjects. Or if you prefer (because this is a pain to do via the GUI!), you can create 16 blank "Subject" items, save the batch script, and then populate the "Scans" field (and Conditions field) via a MATLAB script (see below). Finally, set the variance to "Unequal" and the "Independence" to "No" (to model the error correlation, i.e., nonsphericity, mentioned above). Keep all the remaining defaults.

The next step is to add a module for model estimation from the "Stats" option and define the file name as being dependent on the results of the factorial design specification output. For "write residuals", keep "no". Select classical statistics.

The final step is to add a module for creating contrasts. Define the file name as dependent on the model estimation. The first contrast will be a generic one that tests whether significant variance is captured by the first 3 regressors. This corresponds to an F-contrast based on a  $3 \times 3$  identity matrix. Highlight contrast sessions and select a new F-contrast session, using the current item module. Name this contrast "All Effects". Then define the weights matrix by typing in `[eye(3) ones(3,16)/16]` (which is MATLAB for a  $3 \times 3$  identity matrix, followed by  $1/16$  for each of the 16 subject effects; the latter being necessary if one wants to see absolute changes in



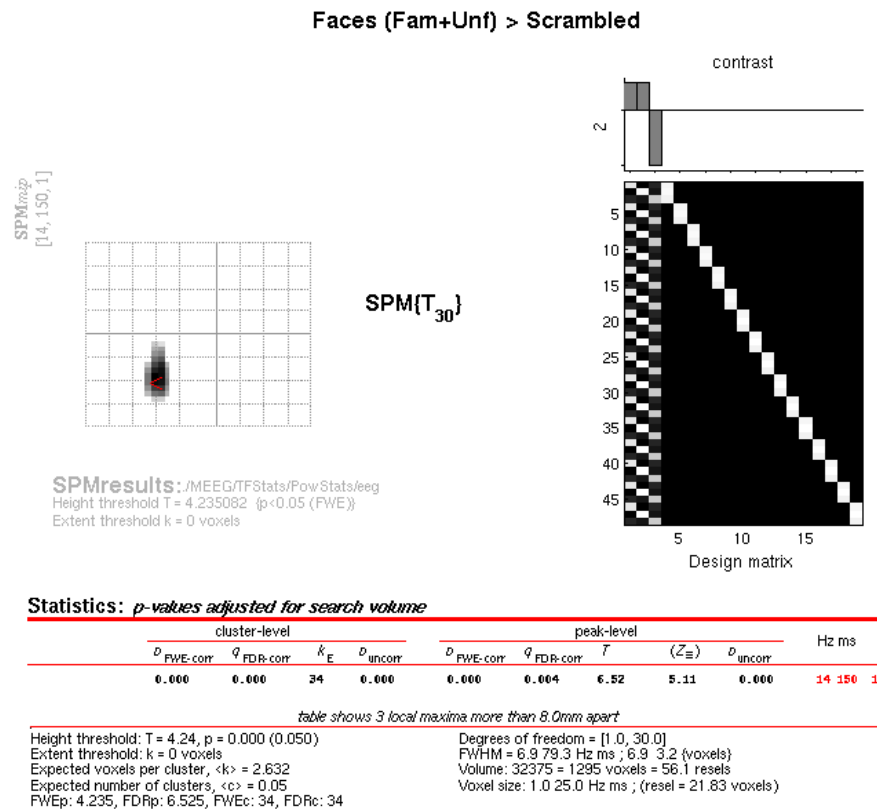


Figure 42.10: Time-Frequency SPM for  $T$ -contrast of greater power for faces than scrambled faces across subjects, thresholded at  $p < .05$  FWE corrected.

power vs baseline). You can use this contrast later to plot the parameter estimates for the 3 conditions.

More interestingly perhaps, we can also define a contrast that compares faces against scrambled faces, i.e., to test whether the average power increase across trials seen in Channel 70 of Subject 15 in Figure 42.8 is reliable when averaging across channels and across subjects. So this time make a  $T$ -contrast, name this one “Faces (Fam+Unf) > Scrambled”, and type in the weights [0.5 0.5 -1]. (If you want to look at power decreases, you can create another  $T$ -contrast and reverse the sign of these contrast weights.)

**Save batch and review** Once you had added all the contrasts you want, you can save this batch file (it should look like the `batch_stats_rmANOVA_job.m` file in the SPM12batch FTP directory). Then run it, and when it has finished, press “Results” from the SPM Menu window. Select the `SPM.mat` file in the `MEEG/TFstats/eeg` directory, and from the new Contrast Manager window, select the pre-specified  $T$ -contrast “Faces (Fam+Unf) > Scrambled”. Within the Interactive window, when given the option, select the following: Apply Masking: None, P value adjustment to control: FWE, keep the threshold at 0.05, extent threshold voxels: 0; Data Type: Time-frequency. The Graphics window should then show what is in Figure 10 below. Note the increase in power at 14Hz, 150ms that survives correction (in all sensor types in fact). (If you examine the reverse contrast of greater power for scrambled than intact faces, you will see a decrease in the beta range, 22Hz, 475ms).

## 42.7 fMRI Preprocessing and Statistics

We will keep description of the preprocessing and analysis of the fMRI data to a minimum, given that fMRI analysis is covered in several other chapters. We start with preprocessing Subject 15's data. For the fMRI experiment, there were 9 runs (sessions), rather than the 6 in the M/EEG experiment. This is because the SOA was increased in the fMRI experiment, by virtue of jittering, which is necessary to estimate the BOLD response versus interstimulus baseline (e.g., to compare with the evoked EEG/MEG response versus prestimulus baseline). If you already launched SPM to analyse the M/EEG data from the previous sections, select "fMRI" instead of "EEG" in the pull-down menu of the main SPM window, otherwise start SPM by typing `spm_fmri`. Preprocessing involves the following modules, which can be found under the menu "SPM – Spatial":

### 42.7.1 Realignment of EPI (fMRI) data

The first step is to coregister the 9 runs of 208 images to one another (i.e., correcting for movement) using a rigid-body transform. Select "Realign: Estimate & Reslice", and on the "Data" item, add nine new "Sessions". Then, for the first session, select the 208 `f*.nii` images in the `BOLD/Run_01` directory of Subject 15 (using right-click, "Select All"), and then repeat for the remaining eight sessions. On the "Reslice Options" item, change the default value of "All Images + Mean Image" to "Mean Image Only". This is because we do not re-slice the EPI data: the coregistration parameters will be stored in the header information of the NIfTI file, and can be combined with the normalisation parameters when re-slicing the normalised images below (re-slicing can introduce interpolation artifacts so generally best to reduce number of re-slicings). However, we do need a re-sliced mean image, which we can use for coregistration with the T1 below. Thus the contents of the `f*.nii` files will change (as header updated), but no new (`rf*.nii`) files will be output, except for the `meanf*.nii` file.

### 42.7.2 Normalisation/Segmentation of T1 images

We will use SPM12's unified segmentation to estimate normalisation parameters to MNI space. Select "Normalise: Estimate" item, add a new "Subject" in "Data" and select the `mprage.nii` image in the "SMRI" directory of Subject 15 for "Image to Align". The output of this step will be a file `y_mprage.nii` containing the estimated deformation field that warps data from subject to MNI space.

### 42.7.3 Coregistration of mean EPI (fMRI) to T1 (sMRI)

Because we have determined the normalisation parameters from the subject's native space to MNI space via the unified segmentation of their T1 image, we need to coregister all the EPI images to that T1 image, so that the normalisation warps can be later applied. Select "Coregister - Estimate" item, and for the "Reference Image", select the same `mprage.nii` image in the `SMRI` directory. For the "Source Image", select "Dependency" and then use "Realign: Estimate & Reslice: Mean Image". For "Other Images", select "Dependency" and then use the `Ctrl` key to select all 9 sessions from the "Realign" stage.

### 42.7.4 Application of Normalisation parameters to EPI data

We can now apply the normalisation parameters (warps) to each of the EPI volumes, to produce new, re-sliced `wf` images. Select "Normalise – Write" item, and for the "Deformation Field", use select the "Normalise: Estimate: Deformation" dependency. For the "Images to Write", select the "Coregister: Estimate: Coregistered Images". You can also change the default voxel size from `[2 2 2]` to `[3 3 3]` if you want to save disk space, since the original data are `[3 3 3.9]` (for Subject 15), so interpolation does not really gain new information.

### 42.7.5 Smoothing

Finally, we smooth the normalised images by an 8mm isotropic Gaussian kernel to produce `swf*` images. So select the “Smooth” item, and select the input to depend on the output of the prior “Normalisation: Write” stage.

**Save batch and review** You can save this batch file (it should look like the `batch_preproc_fmri_job.m` file in the `SPM12batch` FTP directory), and then run it. You can inspect the realignment parameters, normalisations, etc. as described in other chapters. Make a new directory called `Stats` in the `Sub15/BOLD` directory.

### 42.7.6 Creating a 1st-level (fMRI) GLM

Select the “fMRI model specification” option from the “SPM – Stats” menu. Select the new `Stats` directory you created as the output directory. Set the “Units for design” to “seconds” (since our onsets files are in units of seconds) and the “interscan interval” (TR) to 2. Then under the “Data & Design” option, create a new Session, and then select all the `swf*.nii` images in the `Run_01` directory as the “Scans”. Then under the “Multiple conditions” option, press “Specify” and select the file `run_01_spmdef.mat` that has been provided in the `Trials` sub-directory. This is a MATLAB file that contains the onsets, durations and names of every trial in `Run1` (for this subject). Then under the “Multiple regressors” option, press “Specify” and select the file matching `rp*.txt` in the `Run_01` directory. This is a text file that contains the 6 movement parameters for each scan, which was created during “Realignment” above, and we will add to the GLM to capture residual motion-related artifacts in the data.

For the basis functions, keep “Canonical HRF”, but change the “model derivatives” from “no” to “time and dispersion derivatives” (see earlier Chapter manuals). Then keep the remaining options as their defaults.

You then need to replicate this for the remaining 8 sessions, updating all three fields each time: i.e., the scans, conditions and (movement) regressors. It is at this point, that you might want to switch to scripting, which is much less effort – see e.g. this:

```
swd = '../Sub15/BOLD'; % folder containing Subject 15's fMRI data
clear matlabbatch
matlabbatch{1}.spm.stats.fmri_spec.dir = {fullfile(swd,'Stats')};
matlabbatch{1}.spm.stats.fmri_spec.timing.units = 'secs';
matlabbatch{1}.spm.stats.fmri_spec.timing.RT = 2;
for r=1:9
    matlabbatch{1}.spm.stats.fmri_spec.sess(r).scans = ...
        cellstr(spm_select('FPLIST',fullfile(swd,sprintf('Run_%02d',r)), '^swf.*\.nii'));
    matlabbatch{1}.spm.stats.fmri_spec.sess(r).multi = ...
        cellstr(fullfile(swd,'Trials'),sprintf('run_%02d_spmdef.mat',r));
    matlabbatch{1}.spm.stats.fmri_spec.sess(r).multi_reg = ...
        cellstr(spm_select('FPLIST',fullfile(swd,sprintf('Run_%02d',r)), '^rp.*\.txt'));
end
matlabbatch{1}.spm.stats.fmri_spec.bases.hrf.derivs = [1 1];
spm_jobman('interactive',matlabbatch);
```

### 42.7.7 Model Estimation

Add a module for “Model estimation” from the “SPM – Stats” menu and define the `SPM.mat` file name as being dependent on the results of the fMRI model specification output. For “write residuals”, keep “No” and stick to “Classical” estimation.

### 42.7.8 Setting up contrasts

To create some contrasts, select “Contrast Manager” from the “SPM – Stats” menu. Define the `SPM.mat` file name as dependent on the model estimation. The first contrast will be a generic one that tests whether significant variance is captured by the 3 canonical HRF regressors (one per condition). So create a new F-contrast, call it the “Canonical HRF effects of interest”, and enter as the weights matrix (SPM will automatically perform zero padding over the movement regressors):

```
[1 0 0 0 0 0 0 0 0
 0 0 0 1 0 0 0 0 0
 0 0 0 0 0 0 1 0 0]
```

Then select “Replicate” to reproduce this across the 9 sessions. We can also define a T-contrast to identify face-related activation, e.g., “Faces > Scrambled Faces”, given the weight matrix  $[0.5 \ 0 \ 0 \ 0.5 \ 0 \ 0 \ -1 \ 0 \ 0]$ , again replicated across sessions. (Note that there are 3 basis functions per condition, and the zeros here ignore the temporal and dispersion derivatives, but if you want to include them, you can add them as separate rows and test for any face-related differences in BOLD response with an F-contrast; see earlier chapters).

Finally, for the group-level (2nd-level) analyses below, we need an estimate of activation of each condition separately (versus baseline), averaged across the sessions. So create three new T-contrasts, whose weights correspond to the three rows of the above F-contrast, i.e. that pick out the parameter estimate for the canonical HRF (with “Replicate over sessions” set to “Replicate”):

```
for Famous: [1 0 0 0 0 0 0 0 0],
for Unfamiliar: [0 0 0 1 0 0 0 0 0],
for Scrambled: [0 0 0 0 0 0 1 0 0].
```

These T-contrasts will be numbered 3-5, and used in group analysis below.<sup>4</sup>

**Save batch and review** You can save this batch file (it should look like the `batch_stats_fmri_job.m` file in the `SPM12batch` FTP directory). When it has run, you can press “Results” from the SPM Menu window, select the `SPM.mat` file in the `BOLD` directory, and explore some of the contrasts. However, we will wait for the group analysis below before showing results here.

**Creating a script for analysis across subjects** Now that we have created a pipeline for fMRI preprocessing and analysis for a single subject, we can script it to run on the remaining 15 subjects. Below is an example from the `master_script.m`:

```
nrun = 9;

spm_jobman('initcfg');
spm('defaults', 'fMRI');
for s = 1:nsub

    %% Change to subject's directory
    swd = fullfile(outpth,subdir{s},'BOLD');
    cd(swd);

    %% Preprocessing
    jobfile = {fullfile(scrpth,'batch_preproc_fmri_job.m')};
    inputs = cell(nrun+2,1);
    for r = 1:nrun
        inputs{r} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'BOLD',sprintf('Run_%02d',r)), '^fMR.*\.nii$'));
    end
    inputs{10} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'SMRI'),'^mprage.*\.nii$'));
    inputs{11} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'SMRI'),'^mprage.*\.nii$'));
    spm_jobman('serial', jobfile, '', inputs{:});

    %% 1st-level stats
    jobfile = {fullfile(scrpth,'batch_stats_fmri_job.m')};
    inputs = {}; %cell(nrun*3+1,1);
    inputs{1} = {fullfile(swd,'Stats')};
    try mkdir(inputs{1}{1}); end
    for r = 1:nrun
        inputs{end+1} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'BOLD',sprintf('Run_%02d',r)), '^swfMR.*\.nii$'));
        inputs{end+1} = cellstr(fullfile(outpth,subdir{s},'BOLD','Trials',sprintf('run_%02d_spmdef.mat',r)));
    end
end
```

<sup>4</sup> Note that you can create a MATLAB variable containing the weights of the F-contrast with `C = kron(eye(3), [1 0 0]),` and then enter `C, 0.5*C(1,:)+0.5*C(2,:)-C(3,:), C(1,:), C(2,:)` and `C(3,:)` respectively for the 5 contrasts specified above.

```

        inputs{end+1} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'BOLD',sprintf('Run_%02d'
            ,r)), '^rp.*\.txt$'));
    end
    spm_jobman('serial', jobfile, '', inputs{:});

    eval(sprintf('!rm -r %s/Run*/f*',swd)) % to save diskspace (take care!)
    eval(sprintf('!rm -r %s/Run*/wf*',swd)) % to save diskspace (take care!)
end

```

(The last two lines in the loop are optional, to delete intermediate files and save disk space.) Once you have run this script, we can do 2nd-level (group) statistics on resulting contrast images for each condition (averaged across 9 runs).

### 42.7.9 Group Statistics on fMRI data

Now we have a new set of  $16 \times 3$  NIfTI images for each subject and each condition, we can put them into the same repeated-measures ANOVA that we used to test for differences in power across sensors in the time-frequency analysis above, i.e. re-use the `batch_stats_rmANOVA_job.m` file created above. This can be scripted as:

```

fmristsatdir = fullfile(outpth,'BOLD');
if ~exist(fmristsatdir)
    eval(sprintf('!mkdir %s',fmristsatdir));
end

jobfile = {fullfile(scrpth,'batch_stats_rmANOVA_job.m')};

inputs = cell(nsub+1, 1);
inputs{1} = {fmristsatdir};
for s = 1:nsub
    inputs{s+1,1} = cellstr(strvcat(spm_select('FPList',fullfile(outpth,subdir{s},'BOLD','Stats'),'
        con_000[345].nii'))); % Assumes that these T-contrasts in fMRI 1st-level models are famous,
        unfamiliar, scrambled (averaged across sessions)
end

spm_jobman('serial', jobfile, '', inputs{:});

```

**Save batch and review.** When the script has run, press “Results” from the SPM Menu window and select the `SPM.mat` file in the BOLD directory. From the Contrast Manager window, select the pre-specified T-contrast “Faces (Fam+Unf) > Scrambled”. Within the “Stats: Results” window, when given the option, select the following: Apply Masking: None, P value adjustment to control: FWE, keep the threshold at 0.05, extent threshold voxels: 0; Data Type: Volumetric 2D/3D. The Graphics window should then show what is in Figure 42.11 below. Note the left and right OFA and FFA (more extensive on right), plus a small cluster in left medial temporal lobe.

Later, we can use these five clusters as priors for constraining the inversion of our EEG/MEG data. To do this, we need to save these as an image. Press the “save...” button in the bottom right of the SPM Results window, and select the “all clusters (binary)” option. The window will now prompt you for an output filename, in which you can type in `fac-scr_fmri_05_cor`. This image will be output in the BOLD directory, and we will use it later.

## 42.8 Source Reconstruction

To estimate the cortical sources that give rise to the EEG and MEG data, we will return to Subject 15, in order to demonstrate forward and inverse modelling. We need to use the structural MRI of the subject to create a “head model” (that defines the cortex, skull and scalp in terms of meshes) and then a “forward model” (that uses a conductor model to simulate the signal at each sensor predicted by a dipolar source at each point in the cortical mesh). This corresponds to an “imaging” or “distributed” solution to the “Inverse problem”, but you should note that SPM offers other inverse solutions, such as a Bayesian implementation of Equivalent Current Dipoles, and also DCM, which can be viewed as a type of inverse solution (see other chapters in this manual).

You can view the structural MRI of Subject 15 by displaying the NIfTI file `mprage.nii` in the SMRI (T1) sub-directory. This image was manually positioned to roughly match Talairach space,

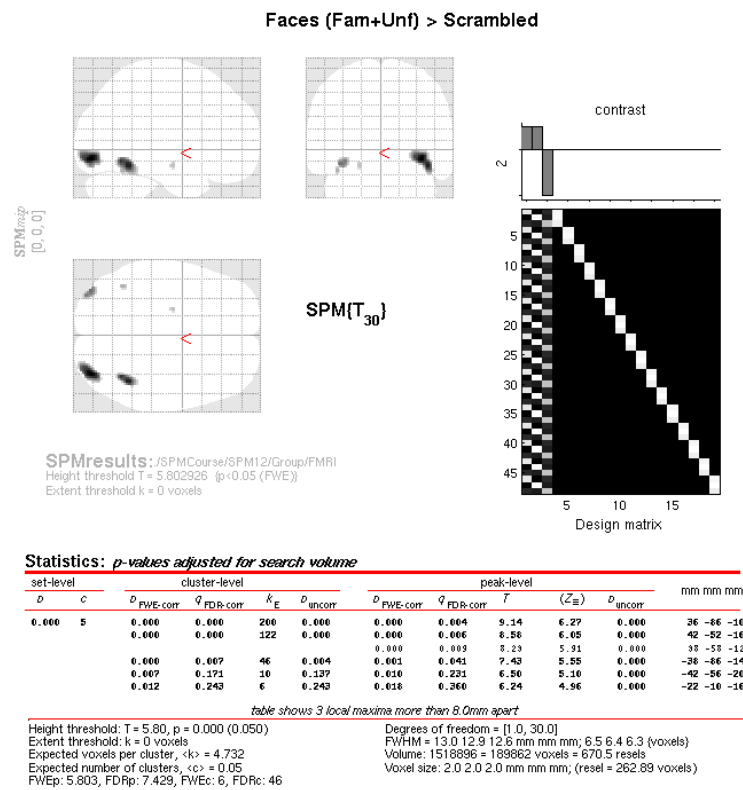


Figure 42.11: Group SPM for Faces vs Scrambled fMRI data at  $p < .05$  FWE-corrected.

with the origin close to the Anterior Commissure. The approximate position of 3 fiducials within this MRI space – the nasion, and the left and right pre-auricular points – are stored in the file `mri_fids.mat`. These were identified manually (based on anatomy) and are used to define the MRI space relative to the EEG and MEG spaces, which need to be coregistered (see below).

To estimate total power (evoked and induced) of the cortical sources, we need to have the original data for each individual trial. Therefore our input file will be `apMcbdspmeeg_run_01_sss.mat` (we could select the trial-averaged file if we just wanted to localise evoked effects). Note that one cannot localise power (or phase) data directly, nor can one localise RMS data from combined gradiometers.

### 42.8.1 Create Head Model

Select the source reconstruction option in the batch window, and select “Head model specification”. Select the file `apMcbdspmeeg_run_01_sss.mat` as the “M/EEG datasets”, and the inversion index as 1 (this index can track different types of forward models and inverse solutions, for example if you want to compare them in terms of log-evidence, e.g., Henson et al, 2009). Additional comments relating to each index can be inserted if “comments” is selected.

The next step is to specify the meshes. Highlight “meshes” and select “mesh source”. From here select “Individual structural image” and select the `mprage.nii` in the `SMRI` directory. The mesh resolution can be kept as normal (approximately 4000 vertices per hemisphere). Note that the cortical mesh (and scalp and skull meshes) are created by warping template meshes from a brain in MNI space, based on normalising this subject’s MRI image to that MNI brain (see papers on “canonical” meshes in SPM).

To coregister the MRI and MEEG data, you must select “specify coregistration parameters”. First you need to specify the select fiducials. You will need to select at least three of these, with coordinates in the space of the MRI image selected. Here we will define “Nasion”; “LPA”; “RPA”. You can do this by loading the MRI and clicking, but here we will use the coordinates provided in the `mri_fids.mat` file described above. You can view these coordinates by loading that file into MATLAB but we repeat them below for convenience. For the Nasion, select “type MRI coordinates” and enter: `[4 112 1]`; for LPA, enter `[-80 21 -12]`; for RPA, enter `[79 9 -31]`.

As well as the fiducials, a number of “head-points” across the scalp were digitised. These were read from the FIF file and stored in the SPM MEEG file. These can help coregistration, by fitting them to the scalp surface mesh (though sometimes they can distort coregistration, e.g. if the end of the nose is digitised, since the nose does not appear on the scalp mesh, often because it has poor contrast on T1-weighted MRI images). If you keep “yes” for the “use headshape points” option, these points will be used, but you will notice that alignment of the fiducials is not as good (as if you don’t use the headshape points), most likely because the nose points are pulling it too far forward. So here we will say “no” to the “use headshape points” option, so as to rely on the fiducials alone, and trust the anatomical skills of the experimenter. (Alternatively, you could edit the headpoints via the command line or a script so as to remove inappropriate ones, but we will not go into the details here).

Finally, for the forward model itself, select “EEG head model”, and specify this as “EEG BEM”; select “MEG head model” and specify this as “Single Shell”. This can then be run. Note that the model parameters are saved, but the gain matrix itself is not estimated until inversion.

**Save batch and review.** You can now save this inversion batch file (it should look like the `batch_localise_forward_model_meeg_job.m` file in the `SPM12batch` FTP directory). Once you have run it, you can explore the forward model by pressing the “3D Source Reconstruction” button within the SPM Menu window. This will create a new window, in which you can select “Load” and choose the `apMcbdspmeeg_run_01_sss.mat` file. On the left hand side of the “source localisation” window, select the “display” button below the “MRI” button. This will bring up the scalp (orange), inner and outer skull (red) and cortical (blue) meshes of Subject 15’s brain, like in Figure 42.12 (left, after rotating slightly with MATLAB’s 3D tool). Note that the fiducials are shown by cyan disks.

Next, select the “display” button beneath “Co-register” and then select “EEG” when asked what to display. The Graphics window should then display an image like in Figure 12 (right)



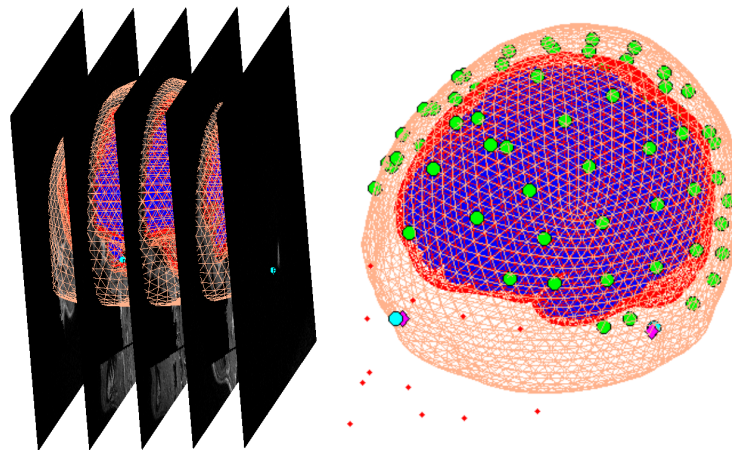


Figure 42.12: *Coregistration of meshes with MRI (left) and meshes with MEG (right).*

that displays the electrode locations in green disks, the digitized headpoints in small red dots, the fiducials in the EEG data as purple diamonds, and the MRI fiducials as cyan disks again. The overlap between the EEG fiducials and MRI fiducials indicates how well the data have been coregistered (assuming no error in marking these anatomical features).

Finally, select “Display” located beneath “Forward Model” and this time select “EEG”. You should see an image displaying the EEG electrode locations relative to the four meshes.

### 42.8.2 Model Inversion

We will compare two approaches to inverting the above forward model (both within a Parametric Empirical Bayesian framework). The first one is called “Multiple Sparse Priors”, which is a novel approach unique to SPM. This corresponds to a sparse prior on the sources, namely that only a few are active. Go back to the batch editor, and select “M/EEG – Source reconstruction – Source Inversion”. Select the same input file `apMcbdspmeeg_run_01_sss.mat`, and set the inversion index to 1. Highlight “what conditions to include” and select “All”. Next highlight inversion parameters, choose “custom” and set the inversion type to “GS”. This is one of several fitting algorithms for optimising the MSP approach: Greedy Search (GS), Automatic Relevance Detection (ARD) and GS+ARD. We choose GS here because it is quickest and works well for these data. Then enter the time window of interest as `[-100 800]`. Set the frequency window of interest to `[0 256]`. Select “yes” for the “PST Hanning window”. Keep all the remaining parameters at their defaults, including the Modalities as “All” (which will simultaneously invert, or “fuse”, the data from the EEG, magnetometers and gradiometers [Henson et al, 2009]).

The second type of inversion we will examine corresponds to a L2-minimum norm (MNM), i.e. fitting the data at the same time as minimising the total energy of the sources. In SPM, this is called “IID” because it corresponds to assuming that the prior probability of each source being active is independent and identically distributed (i.e., an identity matrix for the prior covariance). Go back to the batch editor, add another “M/EEG – Source reconstruction – Source Inversion” module, and select the same input files as before (`apMcbdspmeeg_run_01_sss.mat`), but this time set the inversion index to 2. Set the inversion parameters to “custom”, but the inversion type to be “IID”. The remaining parameters should be made to match the MSP (GS) inversion above.

### 42.8.3 Time-frequency contrasts

Here we are inverting the whole epoch from -100 to +800ms (and all frequencies), which will produce a timecourse for every single source. If we want to localise an effect within the cortical mesh, we need to summarise this 4D data by averaging power across a particular time-frequency window. To do this, select “M/EEG – Source reconstruction – Inversion Results”. Specify the input as dependent on the output of the source inversion, and set the inversion index to 1. Based



on the results of the group sensor-level time-frequency analyses in the previous section, set the time window of interest to [100 250] and the frequency window of interest to [10 20]. For the contrast type, select “evoked” from the current item window, and the output space as “MNI”. Then replicate this module to produce a second “inversion results” module, simply changing the index from 1 to 2 (i.e. to write out the time-frequency contrast for the MNM (IID) as well as MSP (GS) solution).

Now the source power can be written in one of two ways: 1) either as a volumetric NIfTI “Image”, or as 2) a surface-based GifTI “Mesh”. The source data are currently represented on the cortical mesh, so writing them to a volumetric image involves interpolation. If we chose this, we could treat the resulting contrast images in the same way that we do MRI images, and use 3D Random Field Theory (RFT) for voxel-wise statistics. This is likely to require considerable 3D smoothing however, to render the interpolated cortical surface more suitable for RFT. This is what was done in SPM8. However, in SPM12, RFT can also be applied to 2D surfaces (that live in a 3D space), in which case we can restrict smoothing to a small amount across the surface (rather than volume), which is closer to the nature of the data. Therefore, we will chose “Mesh” here to write out GifTI surfaces (which are also much smaller in filesize), keeping the default cortical smoothing of 8.

**Save batch and review** You can now save this inversion batch file (it should look like the `batch_localise_evoked_job.m` file in the SPM12batch FTP directory). It will take a while to run (because it has to create the gain matrix for the first time), after which you can review the inverse results from within the same “3D Source Reconstruction” interface that you used to examine the forward model above. You have to re-“Load” the `apMcbdspmeeg_run_01_sss.mat` file. The latest inversion index will be shown (2 in this case), which corresponds to the IID inversion. Press the “mip” button below the “Invert” button, and you should see something like Figure 42.13. The top plot shows the evoked responses for the three conditions from the peak vertex (at +52 -59 -21, i.e. right fusiform) at 165ms, with the red line being the currently selected condition, here “1” for Famous faces (press the “condition” button to toggle through the other conditions). If you press “display” under the “Window” button, you can see a MIP for the time-frequency contrast limited to the 100-250ms, 10-20Hz specified above, or if you press the “display” under the “Image” button, you will see a rendered version.

If you press the “previous” button, you can select the previous inversion (1), which here corresponds to the MSP inversion. Press the “mip” button again, and you should see results that are sparser and deeper inside the brain, in medial and anterior temporal cortex. This solution actually has a higher model evidence (even though it explains a smaller % of the data variance) because it corresponds to a less complex model (i.e. the posterior deviates less from the prior). We will compare these two inverse solutions in a different way when we do group statistics below.

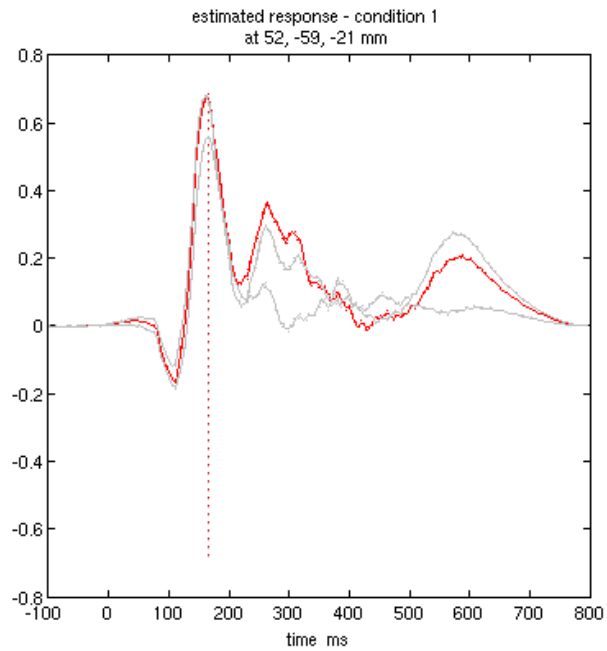
If you like, you can also explore other inversion options, either with batch or with this reconstruction window (e.g., creating new inversion indices, though keep in mind that the associated `apMcbdspmeeg_run_01_sss.mat` file can get very large). For example, you can compare localisation of EEG data alone, or Magnetometers alone, etc.

**Creating a script for analysis across subjects** Now that we have created a pipeline for forward and inverse modelling, we can script it to run on the remaining 15 subjects. Below is an example from the `master_script.m`:

```
for s = 1:nsub

    %% Change to subject's directory
    swd = fullfile(outpth,subdir{s},'MEEG');
    cd(swd);

    jobfile = {fullfile(scrpth,'batch_localise_forward_model_meeg_job.m')};
    inputs = cell(5,1);
    inputs{1} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^apMcbdspmeeg.*\.mat$'));
    ;
    inputs{2} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'SMRI'),'^mprage.*\.nii$'));
    f = load(spm_select('FPList',fullfile(outpth,subdir{s},'SMRI'),'^mri_fids.*\.mat$'));
    inputs{3} = f.mri_fids(1,:);
    inputs{4} = f.mri_fids(2,:);
end
```



PPM at 165 ms (100 percent confidence)  
512 dipoles  
Percent variance explained 88.23 (68.92)  
log-evidence = 8940858.0

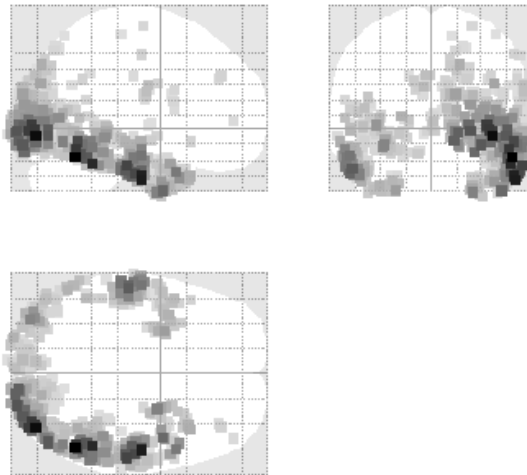


Figure 42.13: *Maximal Intensity Projection (MIP) for Minimum Norm (MNE) inversion of Subject 15's fused EEG and MEG data.*

```

inputs{5} = f.mri_fids(3,:);
spm_jobman('serial', jobfile, '', inputs{:});

%% MSP inversion of EEG, MEG, MEGPLANAR, ALL, then IID inversion of ALL and a time-freq contrast
jobfile = {fullfile(scrpth,'batch_localise_evoked_job.m')};
inputs = cell(1,4);
inputs{1} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^apMcbdspmeeg.*\.mat$'))
;
inputs{2} = {''}; % No fMRI priors
inputs{3} = cellstr(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),'^apMcbdspmeeg.*\.mat'));
inputs{4} = {''}; % No fMRI priors
spm_jobman('serial', jobfile, '', inputs{:});
end

```

Once you have run this script, we can do statistics on the source power GIFTI images created for each subject (using the same repeated-measures ANOVA model that we used for the time-frequency sensor-level analyses above).

#### 42.8.4 Group Statistics on Source Reconstructions

Once we have the  $16 \times 3$  GIFTI images for the power between 10-20Hz and 100-250ms for each subject for each condition, we can put them into the same repeated-measures ANOVA that we used above, i.e. the `batch_stats_rmANOVA_job.m` file. We actually want to do two ANOVAs: one for the MSP inversion and one for the MNM inversion. So we can again script this, like below:

```

srcstatsdir{1} = fullfile(outpth,'MEEG','IndMSPStats');
srcstatsdir{2} = fullfile(outpth,'MEEG','IndMNMStats');

jobfile = {fullfile(scrpth,'batch_stats_rmANOVA_job.m')};

for val = 1:length(srcstatsdir)
    if ~exist(srcstatsdir{val})
        eval(sprintf('!mkdir %s',srcstatsdir{val}));
    end

    inputs = cell(nsub+1, 1);
    inputs{1} = {srcstatsdir{val}};
    for s = 1:nsub
        inputs{s+1,1} = cellstr(strvcat(spm_select('FPList',fullfile(outpth,subdir{s},'MEEG'),sprintf('
^apMcbdspmeeg_run_01_sss_%d.*\.gii$',val))));
    end

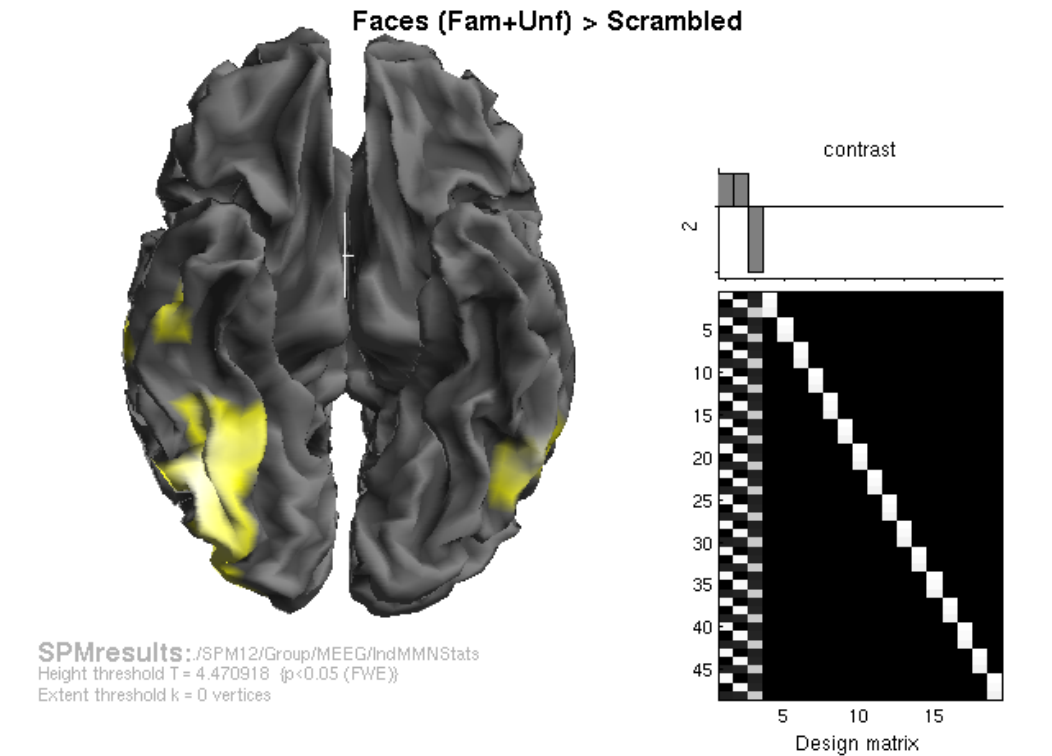
    spm_jobman('serial', jobfile, '', inputs{:});
end

```

where the “Ind” in the output directories refers to “individual” source reconstructions, in contrast to the group-based reconstructions we will do below.

When it has run, press “Results” from the SPM Menu window and select the `SPM.mat` file in the `MEEG/IndMNMStats` directory to look at the results of the minimum norm inversion. From the Contrast Manager window, select the pre-specified T-contrast “Faces (Fam+Unf) > Scrambled”. Within the “Stats: Results” window, select the following: Apply Masking: None, P value adjustment to control: FWE, keep the threshold at 0.05, extent threshold voxels: 0; Data Type: Volumetric 2D/3D. The Graphics window should then show what is in Figure 42.14 below (after having right-clicked on the rendered mesh, selecting “View” and then “x-y view (bottom)”, in order to reveal the underside of the cortex). Note the broad right fusiform cluster, with additional clusters on left and more anteriorly on right. You can compare this to the fMRI group results in the previous section.

You can also look at the results of the MSP inversion by selecting the `SPM.mat` file in the `MEEG/IndMSPStats` directory. This will not reveal any face-related activations that survive  $p < .05$  FWE-corrected. The likely reason for this is that the sparse solutions for each individual subject are less likely to overlap at the group level (than the “smoother” minimum-norm solution). If you change the threshold to  $p < .001$  uncorrected, you will see some activation in posterior right fusiform. However, we can improve the MSP reconstructions by pooling across subjects when actually inverting the data – so-called “group inversion” that we consider next.



**Statistics: *p*-values adjusted for search volume**

set-level		cluster-level				peak-level					mm mm mm		
<i>D</i>	<i>C</i>	<i>D</i> <sub>FWE-corr</sub>	<i>q</i> <sub>FDR-corr</sub>	<i>k</i> <sub>E</sub>	<i>D</i> <sub>uncorr</sub>	<i>D</i> <sub>FWE-corr</sub>	<i>q</i> <sub>FDR-corr</sub>	<i>T</i>	( <i>Z</i> <sub>≡</sub> )	<i>D</i> <sub>uncorr</sub>			
<b>0.000</b>	<b>6</b>	<b>0.000</b>	<b>0.000</b>	<b>114</b>	<b>0.000</b>	<b>0.002</b>	<b>0.209</b>	<b>5.82</b>	<b>4.72</b>	<b>0.000</b>	<b>38</b>	<b>-59</b>	<b>-9</b>
						0.002	0.209	5.74	4.68	0.000	45	-69	-15
						0.003	0.209	5.63	4.61	0.000	40	-45	-18
		<b>0.003</b>	<b>0.147</b>	<b>25</b>	<b>0.049</b>	<b>0.003</b>	<b>0.209</b>	<b>5.65</b>	<b>4.62</b>	<b>0.000</b>	<b>-53</b>	<b>-60</b>	<b>-19</b>
						0.011	0.399	5.09	4.29	0.000	-49	-49	-11
		<b>0.015</b>	<b>0.449</b>	<b>10</b>	<b>0.299</b>	<b>0.012</b>	<b>0.399</b>	<b>5.07</b>	<b>4.28</b>	<b>0.000</b>	<b>52</b>	<b>-17</b>	<b>-36</b>
						0.019	0.413	4.87	4.15	0.000	47	-12	-33
		<b>0.005</b>	<b>0.202</b>	<b>19</b>	<b>0.101</b>	<b>0.016</b>	<b>0.406</b>	<b>4.95</b>	<b>4.20</b>	<b>0.000</b>	<b>48</b>	<b>-22</b>	<b>-8</b>
		<b>0.039</b>	<b>0.786</b>	<b>2</b>	<b>0.786</b>	<b>0.019</b>	<b>0.413</b>	<b>4.88</b>	<b>4.15</b>	<b>0.000</b>	<b>34</b>	<b>4</b>	<b>-18</b>
		<b>0.035</b>	<b>0.786</b>	<b>3</b>	<b>0.696</b>	<b>0.027</b>	<b>0.539</b>	<b>4.73</b>	<b>4.06</b>	<b>0.000</b>	<b>-43</b>	<b>-84</b>	<b>13</b>

*table shows 3 local maxima more than 8.0mm apart*

Height threshold: T = 4.47, p = 0.000 (0.050)  
 Extent threshold: k = 0 vertices  
 Expected vertices per cluster, <k> = 8.284  
 Expected number of clusters, <c> = 0.05  
 FWEp: 4.471, FDRp: Inf, FWEc: 2, FDRc: 114

Degrees of freedom = [1 0, 30.0]  
 FWHM = 4.2 4.2 4.2 (vertices)  
 Volume: 8196 vertices = 107.5 resels  
 (resel = 76.25 vertices)

Figure 42.14: Group SPM for Faces vs Scrambled power on cortical mesh between 10-20Hz and 100-250ms across all 16 subjects at  $p < .05$  FWE-corrected, using Individual MNM inversions.

## 42.9 Group Source Reconstruction

Because of the indeterminacy of the inverse problem, it is helpful to provide as many constraints as possible. One constraint is to assume that every subject has the same underlying source generators, that are simply seen differently at the sensors owing to different anatomy (head models) and different positions with respect to the sensors (forward models). In the PEB framework, this corresponds to assuming the same set of source priors across subjects (allowing for different sensor-level noise; see [Henson et al, 2011]). This group-based inversion can be implemented in SPM simply by selecting multiple input files to the inversion routine, which can be scripted like this (using the same batch file as before, noting that this includes two inversions – MNM and MSP – hence the two inputs of the same data files below):

Note that once you have run this, the previous “individual” inversions in the data files will have been overwritten (you could modify the batch to add new inversion indices 3 and 4, so as to compare directly the group inversions with the previous individual inversions, but the files will get very big).

### 42.9.1 Group Statistics on Source Reconstructions

Now we have a new set of  $16 \times 3$  GIFTI images for the power between 10-20Hz and 100-250ms for each subject for each condition after group-inversion, we can put them into the same repeated-measures ANOVA that we used above, i.e., the `batch_stats_rmANOVA_job.m` file. This can be scripted as (i.e, simply changing the output directories at the start from, e.g, `IndMSPStats` to `GrpMSPStats`).

```
srcstatsdir{1} = fullfile(outpth,'MEEG','GrpMSPStats');
srcstatsdir{2} = fullfile(outpth,'MEEG','GrpMNMStats');

jobfile = {fullfile(scrpth,'batch_stats_rmANOVA_job.m')};

for val = 1:length(srcstatsdir)
    if ~exist(srcstatsdir{val})
        eval(sprintf('!mkdir %s',srcstatsdir{val}));
    end

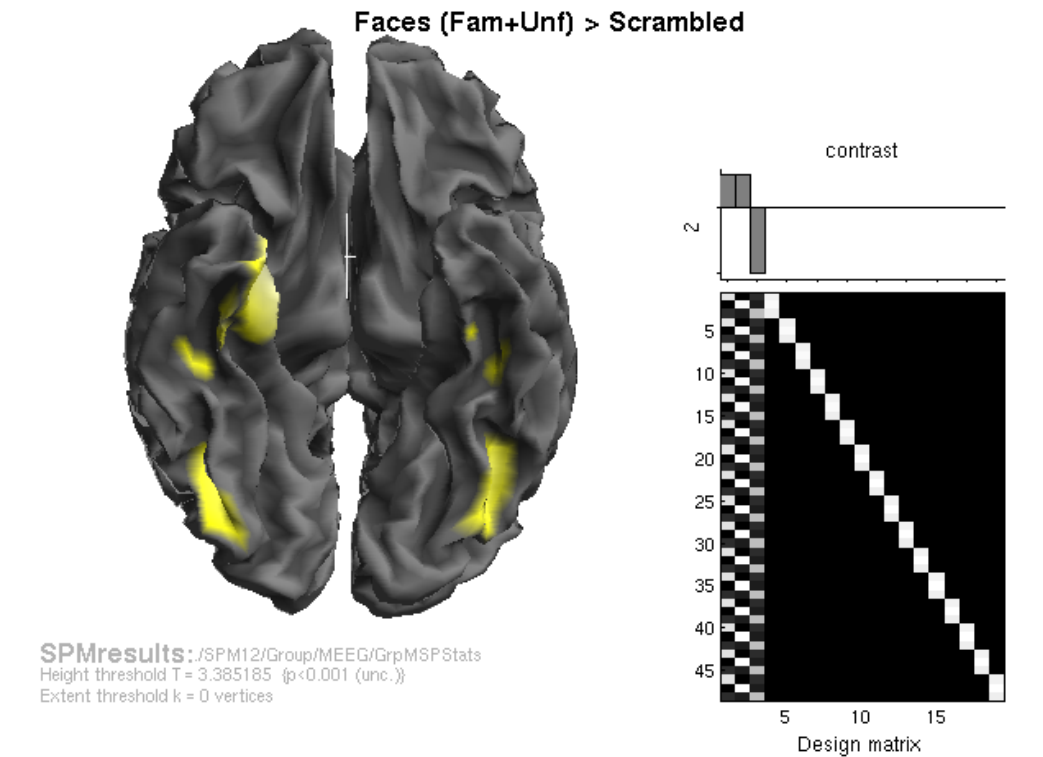
    inputs = cell(nsub+1, 1);
    inputs{1} = {srcstatsdir{val}};
    for s = 1:nsub
        inputs{s+1,1} = cellstr(strvcat(spm_select('FPLIST',fullfile(outpth,subdir{s},'MEEG'),sprintf('
^apMcbdspmeeg_run_01_sss_%d.*\gii',val))));
    end

    spm_jobman('serial', jobfile, '', inputs{:});
end
```

When it has run, press “Results” from the SPM Menu window and select the `SPM.mat` file in the relevant output directories. You will notice that the results for minimum norm have not changed much – a lot of voxels remain significant after correction, but in a broadly distributed swathe of ventral temporal lobe. For the results in the `MEEG/GrpMSPStats` directory, there is a small anterior right temporal cluster that survives correction. But if you lower the threshold to  $p < .001$  uncorrected, you should see results like in Figure 42.15, which includes more focal regions in the ventral temporal lobe, and importantly, more such regions that for the individual MSP inversions the `MEEG/IndMSPStats` directory (demonstrating the advantage of group-based inversion).

## 42.10 Group MEEG Source Reconstruction with fMRI priors

Finally, in an example of full multi-modal integration, we will use the significant clusters in the group fMRI analysis as separate spatial priors for the group-optimised source reconstruction of the fused MEG and EEG data (see [Henson et al, 2011]). Each cluster becomes a separate prior, allowing for fact that activity in those clusters may occur at different post-stimulus times.



**Statistics: *p*-values adjusted for search volume**

set-level		cluster-level				peak-level					mm mm mm		
<i>D</i>	<i>C</i>	<i>D</i> <sub>FWE-corr</sub>	<i>q</i> <sub>FDR-corr</sub>	<i>k</i> <sub>E</sub>	<i>D</i> <sub>uncorr</sub>	<i>D</i> <sub>FWE-corr</sub>	<i>q</i> <sub>FDR-corr</sub>	<i>T</i>	<i>(Z<sub>≡</sub>)</i>	<i>D</i> <sub>uncorr</sub>			
0.000	7	0.046	0.652	39	0.093	0.042	0.485	4.48	3.89	0.000	24	3	-34
		0.280	0.941	7	0.653	0.063	0.435	4.27	3.74	0.000	25	-9	-31
		0.147	0.734	19	0.315	0.088	0.485	4.16	3.67	0.000	38	-18	-36
		0.147	0.734	19	0.315	0.180	0.721	3.82	3.42	0.000	-38	-65	-13
		0.343	0.941	3	0.833	0.203	0.721	3.76	3.37	0.000	36	-66	-12
		0.343	0.941	3	0.833	0.253	0.750	3.64	3.29	0.001	-44	-32	-21
		0.377	0.941	1	0.941	0.281	0.750	3.58	3.24	0.001	46	-30	-21
						0.384	0.962	3.40	3.10	0.001	-36	-21	-31

*table shows 3 local maxima more than 8.0mm apart*

Height threshold: T = 3.39, p = 0.001 (0.396)  
 Extent threshold: k = 0 vertices  
 Expected vertices per cluster, <k> = 16.430  
 Expected number of clusters, <c> = 0.50  
 FWEp: 4.406, FDRp: Inf, FWEc: 39, FDRc: Inf

Degrees of freedom = [1, 0, 30.0]  
 FWHM = 4.5 4.5 4.5 (vertices)  
 Volume: 8189 vertices = 91.9 resels  
 (resel = 89.11 vertices)

Figure 42.15: Group SPM for Faces vs Scrambled power on cortical mesh between 10-20Hz and 100-250ms across all 16 subjects at  $p < .001$  uncorrected, using Group-optimised MSP.

This group-based inversion can be implemented in SPM simply by selecting the binary (thresholded) image we created from the group fMRI statistics (`fac_scr_fmri_05_cor.nii` in the BOLD directory), which contains non-zero values for voxels to be included in the clustered priors. This is simply an option in the inversion module, so can be scripted like this (using the same batch file as before, noting that this includes two inversions – MNM and MSP – hence the two inputs of the same data files below):

```
jobfile = {fullfile(scrpth, 'batch_localise_evoked_job.m')};
tmp = cell(nsub,1);
for s = 1:nsub
    tmp{s} = spm_select('FPList', fullfile(outpth, subdir{s}, 'MEEG'), '^apMcbdspmeeg.*\.mat');
end
inputs = cell(4,1);
inputs{1} = cellstr(strvcat(tmp{:}));
inputs{2} = {fullfile(outpth, 'BOLD', 'fac_scr_fmri_05cor.nii')}; % Group fMRI priors
inputs{3} = cellstr(strvcat(tmp{:}));
inputs{4} = {fullfile(outpth, 'BOLD', 'fac_scr_fmri_05cor.nii')}; % Group fMRI priors
spm_jobman('serial', jobfile, '', inputs{:});
```

Note again that once you have run this, the previous “group” inversions in the data files will have been overwritten (you could modify the batch to add new inversion indices 5 and 6, so as to compare with previous inversions above, but the file will get very big). Note also that we have used group-defined fMRI priors, but the scripts can easily be modified to define fMRI clusters on each individual subject’s 1st-level fMRI models, and use subject-specific source priors here.

### 42.10.1 Group Statistics on Source Reconstructions

After running the attached script, we will have a new set of  $16 \times 3$  GIFTI images for the power between 10-20Hz and 100-250ms for each subject for each condition after group-inversion using fMRI priors, and can put them into the same repeated-measures ANOVA that we used above, i.e. the `batch_stats_rmANOVA_job.m` file. This can be scripted as (i.e. simply changing the output directories at the start from, e.g. `GrpMNMStats` to `fMRIGrpMNMStats`).

```
srcstatsdir{1} = fullfile(outpth, 'MEEG', 'fMRIGrpMSPStats');
srcstatsdir{2} = fullfile(outpth, 'MEEG', 'fMRIGrpMNMStats');

jobfile = {fullfile(scrpth, 'batch_stats_rmANOVA_job.m')};

for val = 1:length(srcstatsdir)
    if ~exist(srcstatsdir{val})
        eval(sprintf('!mkdir %s', srcstatsdir{val}));
    end

    inputs = cell(nsub+1, 1);
    inputs{1} = {srcstatsdir{val}};
    for s = 1:nsub
        inputs{s+1,1} = cellstr(strvcat(spm_select('FPList', fullfile(outpth, subdir{s}, 'MEEG'), ...
            sprintf('^apMcbdspmeeg_run_01_sss_%d.*\.gii$', val))));
    end

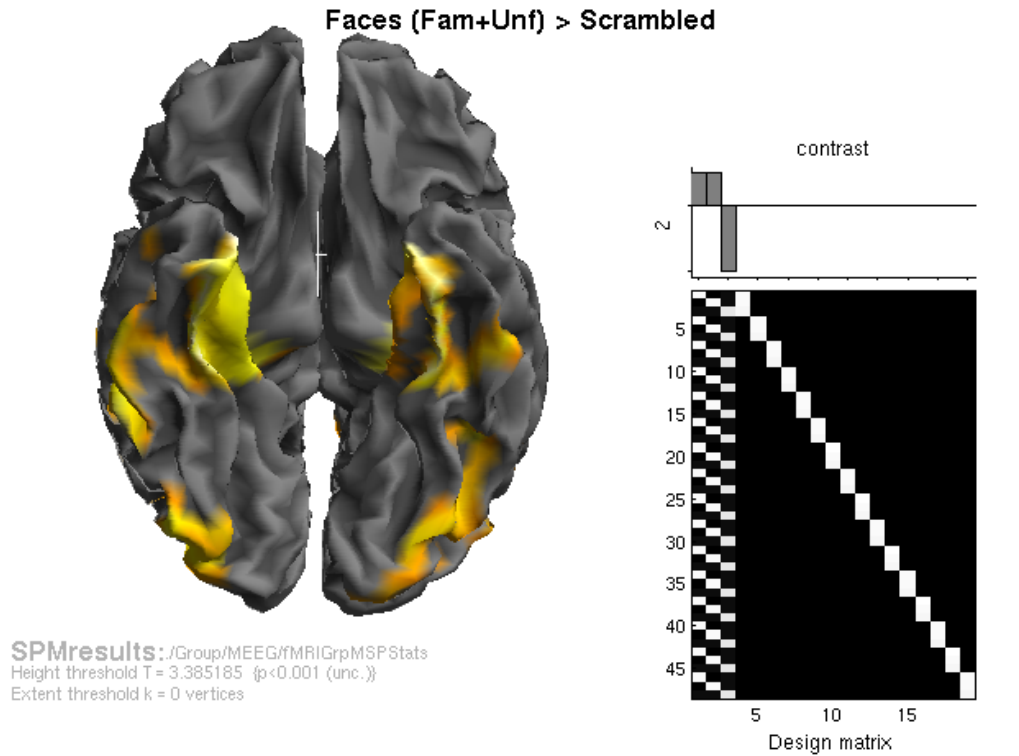
    spm_jobman('serial', jobfile, '', inputs{:});
end
```

When it has run, press “Results” from the SPM Menu window and select the `SPM.mat` file from the `fMRIGrpMSPStats` directory, and choose an uncorrected threshold of  $p < .001$ . You should see results like in Figure 42.16, which you can compare to Figure 42.15. The fMRI priors have improved consistency across subjects, even in medial temporal lobe regions, as well as increasing significance of more posterior and lateral temporal regions (cf., Figure 42.11, at  $p < .001$  uncorrected).

## 42.11 References

1. Henson, R.N., Mattout, J., Phillips, C. and Friston, K.J. (2009). Selecting forward models for MEG source-reconstruction using model-evidence. *Neuroimage*, 46, 168-176.
2. Henson, R.N., Wakeman, D.G., Litvak, V. and Friston, K.J. (2011). A Parametric Empirical





**Statistics: *p*-values adjusted for search volume**

set-level		cluster-level				peak-level						mm mm mm		
<i>D</i>	<i>C</i>	<i>D</i>	<i>q</i>	<i>k<sub>E</sub></i>	<i>D<sub>uncorr</sub></i>	<i>D</i>	<i>q</i>	<i>T</i>	<i>(Z<sub>≡</sub>)</i>	<i>D<sub>uncorr</sub></i>				
		FWE-corr	FDR-corr			FWE-corr	FDR-corr							
0.000	14	0.000	0.000	216	0.000	0.002	0.081	5.75	4.68	0.000	24	6	-43	
						0.003	0.081	5.60	4.60	0.000	26	2	-29	
						0.003	0.081	5.55	4.57	0.000	35	-28	-27	
		0.000	0.001	167	0.000	0.005	0.081	5.32	4.43	0.000	-35	-28	-27	
						0.005	0.081	5.28	4.40	0.000	-26	4	-43	
						0.007	0.081	5.21	4.36	0.000	-34	-7	-41	
		0.038	0.332	45	0.085	0.017	0.139	4.82	4.11	0.000	45	-81	-11	
						0.031	0.166	4.56	3.94	0.000	45	-75	-15	
						0.082	0.330	4.13	3.65	0.000	31	-88	-9	
		0.049	0.332	40	0.112	0.042	0.193	4.43	3.86	0.000	-46	-51	-15	
						0.080	0.323	4.15	3.66	0.000	-26	-84	-17	
						0.183	0.610	3.74	3.36	0.000	-20	-98	-18	
		0.132	0.740	21	0.317	0.072	0.318	4.19	3.69	0.000	44	-50	13	
						0.075	0.320	4.18	3.68	0.000	64	-24	-0	
						0.087	0.334	4.11	3.63	0.000	55	-31	5	
		0.302	0.947	4	0.804	0.073	0.318	4.19	3.69	0.000	-48	-48	1	
						0.322	0.874	3.45	3.14	0.001	-58	-59	-1	
		0.218	0.947	11	0.548	0.188	0.552	3.83	3.43	0.000	-48	-78	5	
						0.324	0.874	3.45	3.14	0.001	-41	-82	6	
		0.346	0.947	1	0.947	0.192	0.613	3.73	3.35	0.000	50	-26	-3	
		0.331	0.947	2	0.896	0.226	0.713	3.65	3.29	0.000	51	7	-35	
		0.346	0.947	1	0.947	0.249	0.772	3.60	3.25	0.001	52	-21	-0	
		0.052	0.332	39	0.119	0.273	0.821	3.55	3.21	0.001	-7	-63	58	
						0.277	0.821	3.54	3.21	0.001	-9	-56	54	
						0.286	0.821	3.52	3.20	0.001	-12	-42	44	
		0.316	0.947	3	0.849	0.285	0.821	3.52	3.20	0.001	-8	-97	21	
		0.346	0.947	1	0.947	0.298	0.821	3.50	3.18	0.001	-60	-29	2	
		0.316	0.947	3	0.849	0.317	0.873	3.46	3.15	0.001	-33	-89	-5	

table shows 3 local maxima more than 8.0mm apart

Height threshold: T = 3.39, p = 0.001 (0.361)  
 Extent threshold: k = 0 vertices  
 Expected vertices per cluster, <k> = 18.290  
 Expected number of clusters, <c> = 0.45  
 FWEp: 4.355, FDRp: Inf, FWEc: 40, FDRc: 167

Degrees of freedom = [1, 0, 30, 0]  
 FWHM = 4.6 4.6 4.6 (vertices)  
 Volume: 7990 vertices = 80.5 resels  
 (resel = 99.20 vertices)

Figure 42.16: Group SPM for Faces vs Scrambled power on cortical mesh between 10-20Hz and 100-250ms across all 16 subjects at *p* < .001 uncorrected, using Group-optimised MSP and fMRI priors.



Bayesian framework for the EEG/MEG inverse problem: generative models for multisubject and multimodal integration. *Frontiers in Human Neuroscience*, 5, 76, 1-16.

3. Wakeman, D.G. and Henson, R.N. A multi-subject, multi-modal human neuroimaging dataset. *Scientific Data*, 2:150001.

## 42.12 Acknowledgements

This work was supported by MRC (A060\_MC\_5PR10). The author (RNH) thanks Rebecca Beresford, Hunar Abdulraham, Daniel Wakeman, Guillaume Flandin and Vladimir Litvak for their help.



## Chapter 43

# DCM for Induced Responses

This chapter shows an example of Dynamic Causal Modelling for Induced Responses (DCM-IR) [18]. The example is based on the analysis described by Chen et al. [17]. The purpose of the analysis is to look at the effective connectivity between cortical areas involved in the processing of faces and specifically at non-linearities in the connections expressed as cross-frequency coupling. DCM-IR is an example of a phenomenological rather than physiological DCM. The advantage of this approach is that it can directly model a particular feature extracted from the data, namely event-related spectral perturbations. This feature has been a popular object of study in the neuroscience literature. However, since computing event-related power involves discarding phase information, it is not possible to model this feature with a physiologically realistic model such as the one used in DCM for evoked responses. An important feature of DCM for induced responses is that it models the full time-frequency spectrum. This differs from typical approaches, where a few specific frequency bands are selected *a priori*. DCM-IR models spectral dynamics in terms of a mixture of frequency modes (obtained with singular value decomposition). The dynamics of each mode are encoded by the evolution of a state. It is this multi-state vector, for each source, that captures how the energy in different frequencies interacts, either linearly or non-linearly, among sources.

### 43.1 Data

We will use the epoched and merged MEG face-evoked dataset<sup>1</sup> saved in the files:

```
cdbespm12_SPM_CTF_MEG_example_faces1_3D.mat  
cdbespm12_SPM_CTF_MEG_example_faces1_3D.dat
```

DCM-IR also requires a head model and coregistration. If you have performed “Imaging” reconstruction of differential power and saved the results, the head model should already be defined. Otherwise, you will be asked to define the head model while configuring the DCM (see below).

### 43.2 Getting Started

You need to start SPM and toggle “EEG” as the modality (bottom-right of SPM main window), or start SPM with `spm eeg`. In order for this to work you need to ensure that the main SPM directory is on your MATLAB path.

### 43.3 Setting up DCM

After calling `spm eeg`, you see SPM’s graphical user interface, the top-left window. The button for calling the DCM-GUI is found in the second partition from the top, on the right hand side.

---

<sup>1</sup>Multimodal face-evoked dataset: <http://www.fil.ion.ucl.ac.uk/spm/data/mmfaces/>

When pressing the button, the GUI pops up (Figure 43.1). The GUI is partitioned into five parts, going from the top to the bottom. The first part is about loading and saving existing DCMs, and selecting the type of model. The second part is about selecting data, the third is for specification of the spatial forward model, the fourth is for specifying connectivity, and the last row of buttons allows you to estimate parameters and view results.

You have to select the data first and specify the model in a fixed order (data selection > spatial model > connectivity model). This order is necessary, because there are dependencies among the three parts that would be hard to resolve if the input could be entered in any order. At any time, you can switch back and forth from one part to the next. Also, within each part, you can specify information in any order you like.

### 43.3.1 load, save, select model type

At the top of the GUI, you can load an existing DCM or save the one you are currently working on. In general, you can **save** and **load** during model specification at any time. You can also switch between different DCM analyses (the left menu). The default is “ERP” which is DCM for evoked responses. You should switch to “IND” which is the option for DCM-IR. The menu on the right-hand side lets you choose the neuronal model. Once you switch to “IND”, it will be disabled since neuronal models are not relevant for DCM-IR, which is a phenomenological DCM.

### 43.3.2 Data and design

In this part, you select the data and model between-trial effects. Press “new data” and select the data file `cdbespm12_SPM_CTF_MEG_example_faces1_3D.mat`. The data file will usually be an epoched file with multiple trials per condition. These data must be in SPM-format. On the right-hand side you can enter trial indices of the evoked responses in this SPM-file. For example, if you want to model the first and second condition contained within an SPM-file, specify indices 1 and 2. You can type:

```
D = spm_eeg_load('cdbespm12_SPM_CTF_MEG_example_faces1_3D.mat');D.condlist
```

in the command line to see the list of condition labels in the order that corresponds to these indices. This order is defined in the dataset and can be modified by selecting “Sort Conditions” from the “Other” submenu in main SPM window (`spm_eeg_sort_conditions`). SPM should echo:

```
ans =  
  
    'faces'    'scrambled'
```

meaning that index 1 corresponds to presentation of faces and index 2 - to presentation of scrambled faces. The box below the list of indices allows specifying experimental effects on connectivity. The specification can be quite generic as in the design matrix for a General Linear Model (GLM). Our case is quite simple though. We have a baseline condition which is “scrambled” and we would like to know how the condition of interest “faces” differs from it. We will therefore enter:

```
1 0
```

in first row of the box, which means that there will be some additive modulation of connections that we will define later for “faces” (some coefficient multiplied by 1) and this modulation will not be there for “scrambled” (the same coefficient multiplied by 0). If we now click somewhere outside the box, a default name will be assigned to this effect - “effect1”. It will appear in the small text box next to the coefficients box. It is possible to change this name to something else e.g. “face”.

Now we can select the peristimulus time window we want to model. These are the two test boxes in the top left corner of the panel. Enter -50 in the left box and 300 in the right box to select the segment -50 to 300 ms relative to the presentation of the visual stimulus.

You can choose whether you want to remove low-frequency drifts of the data at sensor level. If you don’t, select 1 for “detrend”, to just remove the mean. Otherwise select the number of discrete cosine transform terms you want to remove. You can also subsample your data (prior

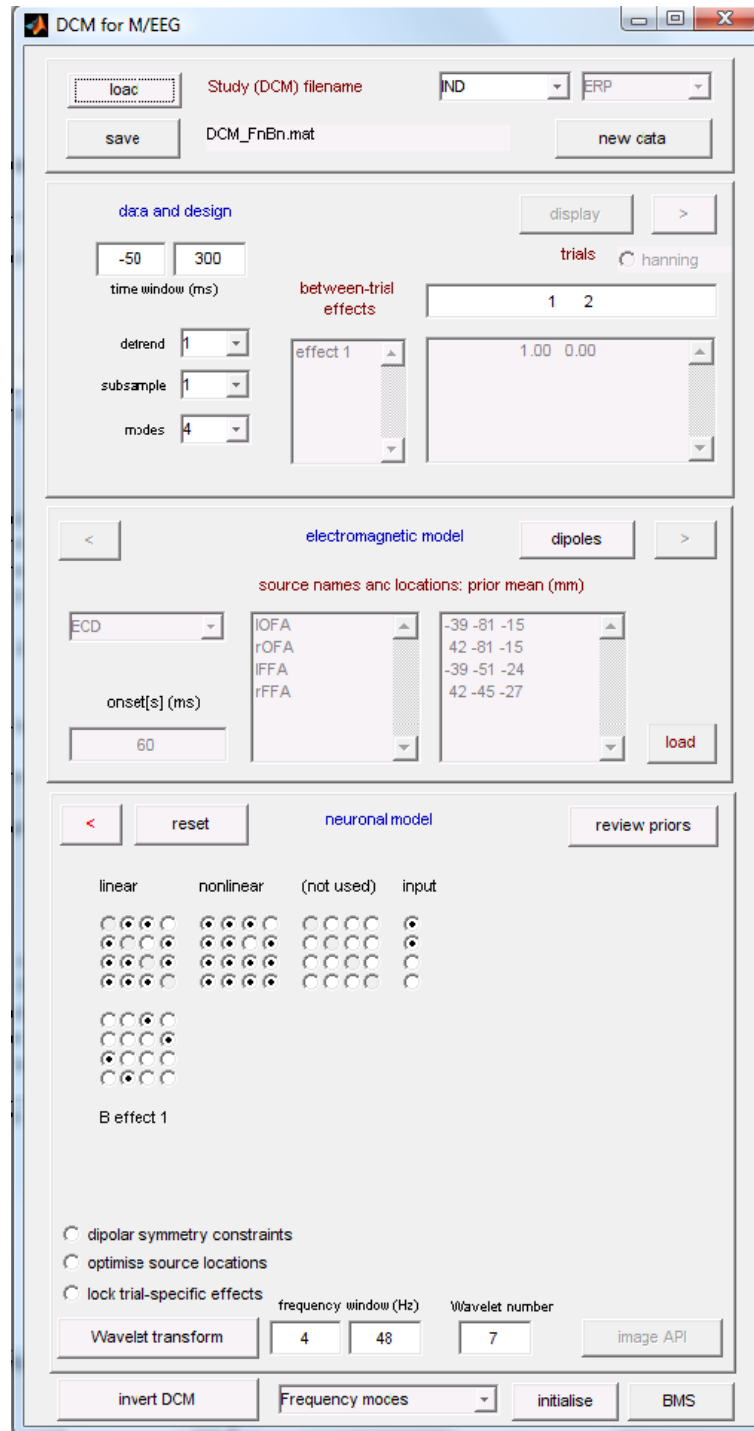


Figure 43.1: DCM window configured analysing induced responses and the *FnBn* model specified.

to computing the time-frequency decomposition) using the “subsample” option. In general, it is advised to filter out drifts and downsample the data during preprocessing. The options here are just to *play around* with, clean up or reduce the data to see what effect it might have without running additional processing steps outside DCM.

Press the “Display” button to look at the selected data. You will see the evoked responses for the two conditions (Figure 43.2) which help you get some idea about your choice of time window. It is possible to change the “detrend” and “subsample” values or the time window and press “Display” again to see what effect these changes have.

An important parameter for DCM-IR is the number of modes. These are the frequency modes mentioned above. The idea is that the main features of the time-frequency image can be represented by a small number of components with fixed frequency profiles that are modulated over time. These components can be determined automatically using “Singular Value Decomposition” (SVD). Generally SVD preserves information from the original time-frequency image and produces as many components as there are frequency bins. However, usually only the first few components are physiologically relevant and the rest are just noise. Using a small number of components will greatly speed-up DCM model inversion. You cannot know in advance what the optimal number of components for your data is. What you can do is try once with a relatively large number (e.g. 8) and then see from the time and frequency profile of the later components (in the Results view, see below) whether they are important. Then you can reduce the number and try again. For the example here it is sufficient to use 4 modes so change the number in “modes” from 8 to 4.

If you are happy with your data selection, the subsampling and the detrending terms, you can click on the > (forward) button, which will bring you to the next stage *electromagnetic model*. From this part, you can press the red < button to get back to the data and design part.

### 43.3.3 Electromagnetic model

With DCM-IR, you have two options for how to extract the source data for time-frequency analysis. Firstly, you can use 3 orthogonal single equivalent current dipoles (ECD) for each source and invert the resulting source model to get source waveforms. This option is suitable for multichannel EEG or MEG data. Alternatively, you can treat each channel as a source (LFP option). This is appropriate when the channels already contain source data either recorded directly with intracranial electrodes or extracted (e.g. using a beamformer).

Note that a difference with DCM for evoked responses is that the parameters of the spatial model are not optimized. This means that DCM-IR will project the data into source space using the spatial locations you provide.

We will use the ECD option. This requires specifying a list of source names in the left large text box and a list of MNI coordinates for the sources in the right large text box. Enter the following in the left box:

```
lOFA
rOFA
lFFA
rFFA
```

Now enter in the right text box:

```
-39 -81 -15
 42 -81 -15
-39 -51 -24
 42 -45 -27
```

These correspond to left Occipital Face Area, right Occipital Face Area, left Fusiform Face Area and right Fusiform Face Area respectively. See [18] for more details.

The onset-parameter determines when the stimulus, presented at 0 ms peri-stimulus time, is assumed to trigger the cortical induced response. In DCM, we usually do not model the rather small early responses, but start modelling at the first large deflection. Because the propagation of the stimulus impulse through the input nodes causes a delay, we found that the default value

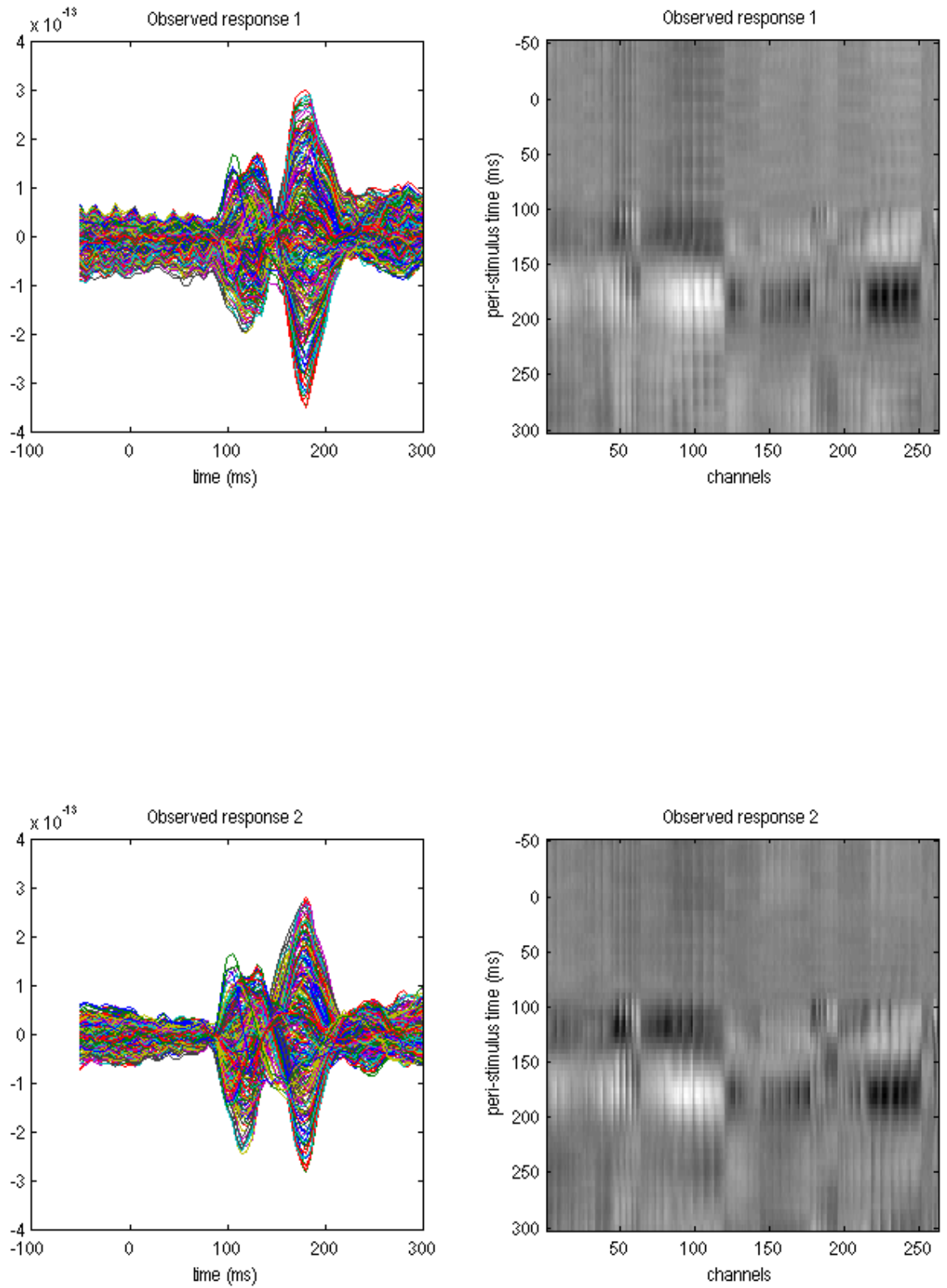


Figure 43.2: Averaged evoked responses after configuring the 'Data and design' section.

of 60 ms onset time is a good value for many responses where the first large deflection is seen around 100 ms. However, this value is a prior, i.e. the inversion routine can adjust it. The prior mean should be chosen according to the specific responses of interest. This is because the time until the first large deflection is dependent on the paradigm or the modality you are working in, e.g. audition or vision. You may also find that changing the onset prior has an effect on how your data are fitted. This is because the onset time has strongly nonlinear effects (a delay) on the data, which might cause differences in the maximum found at convergence, for different prior values. Note, that it is possible to enter more than one number in the “onset[s] (ms)” box. This will add several inputs to the model. These inputs can then be connected to different nodes and/or their timing and frequency profiles can be optimized separately.

When you want to proceed to the next model specification stage, hit the > (forward) button and proceed to the *neuronal model*. If you have not used the input dataset for 3D source reconstruction before you will be asked to specify the parameters of the head model at this stage.

## 43.4 Neuronal model

There are 4 (or more) matrices which you need to specify by button presses. In the first row there are matrices that define the connectivity structure of the model and in the second row there are matrices that specify which connections are affected by experimental effects. All the matrices except one are square. In each of these square matrices you specify a connection *from* a source area *to* a target area. For example, switching on the element (2, 1) means that you specify a connection from area 1 to 2 (in our case from lOFA to rOFA). Some people find the meaning of each element slightly counter-intuitive, because the column index corresponds to the source area, and the row index to the target area. This convention is motivated by direct correspondence between the matrices of buttons in the GUI and connectivity matrices in DCM equations and should be clear to anyone familiar with matrix multiplication.

The leftmost matrix in the first row specifies the *linear* connections. These are the connections where frequency dynamics in one source affects the dynamics at the same frequencies in another source. Note that all connections in the model should be at least linear, so if you think some connection should be present in the model, the corresponding button in this matrix should be on. Also the buttons on the leading diagonal of the matrix are always on because each node in the model has a linear intrinsic connection with negative sign. This means that the activity has a tendency to dissipate. To the right of the linear connectivity matrix there is a *nonlinear* connectivity matrix. The idea here is the same, just remember to enable the corresponding linear connection as well. When a connection is nonlinear, a frequency mode in the source node can affect all the frequency modes in the target node. Intrinsic connections can be made non-linear as well. It is actually recommended to always make the intrinsic connections non-linear unless there is a good theoretical reason not to do it. Since we are mainly interested in non-linearities in the extrinsic connections we would like to be over-conservative and first explain away anything that can be explained by non-linearities in the intrinsic connections.

The rightmost matrix in the first row is the input matrix. It is usually not square, and in the case of a single input, as we have here, is reduced to a column vector. The entries of this vector specify which areas receive the external input (whose onset time we specified above). In the case of several inputs the input matrix will have several columns.

The matrix (matrices) in the second row specify which of the connections defined in the first row can be modified by experimental effects. A connection which is not modified will have the same value for all conditions. If you don't allow modification of any of the connections, then exactly the same model will be fitted to all conditions. For the purpose of allowing modification by experimental effects, it does not matter whether a connection is linear or non-linear. Hence, there is one modulation matrix per experimental effect (defined in the “Data and design” panel). In our case there is only one effect - faces vs. scrambled faces. Also self connections can be modified by experimental effects, thus the diagonal entries of the second row matrices can also be toggled.

Figure 43.3 is taken from the paper of Chen et al. [17] and shows several alternative models that could apply to the data. We will start by specifying the model with nonlinear forward and backward connections (FnBn) and with effect of condition on these connections. The correspond-



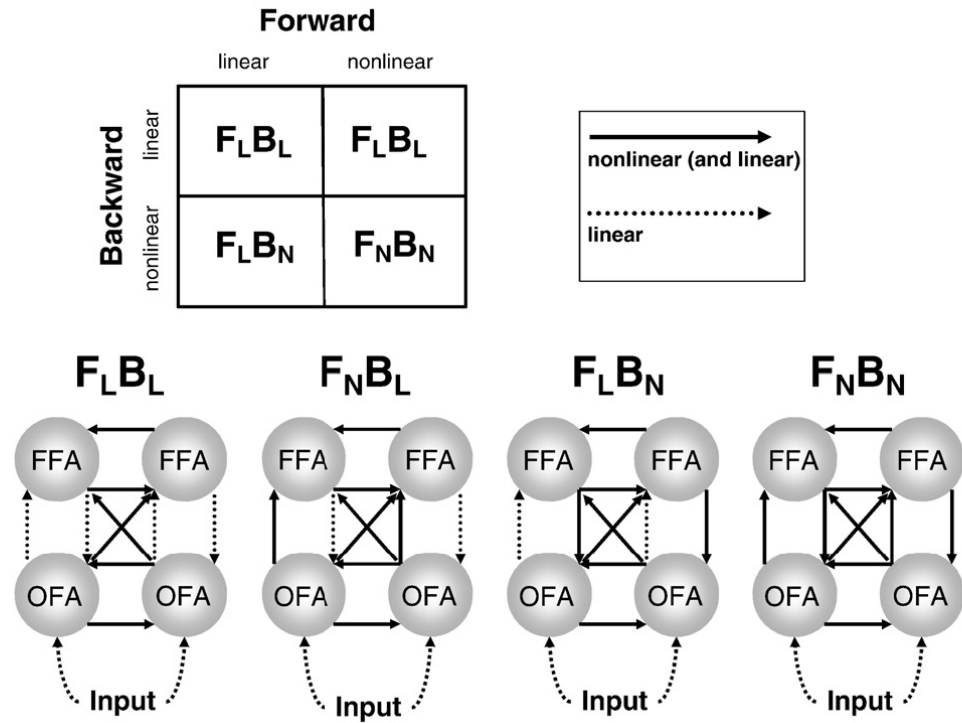


Figure 43.3: Four different DCM-IR models proposed by Chen et al. [17]

ing button configuration is shown in Figure 43.4. Compare the depiction of FnBn model in Figure 43.3 with the specification in Figure 43.4 to see the correspondence. Note that the effect of condition is not shown in Figure 43.3. Now copy the specification to the DCM GUI.

At the bottom of this panel there are additional radio buttons for options that are not relevant for DCM-IR. Below these buttons there are controls for specifying the parameters of the wavelet transform for computing the time-frequency decomposition. We will keep the default frequency window 4 to 48 Hz and increase the number of wavelet cycles to 7. You can press the *Wavelet transform* button to preview the time-frequency plots and optimize the parameters if necessary before inverting the model.

### 43.5 Estimation

When you have finished model specification, you can hit the *invert DCM* button in the lower left corner. DCM will now estimate the model parameters. You can follow the estimation process

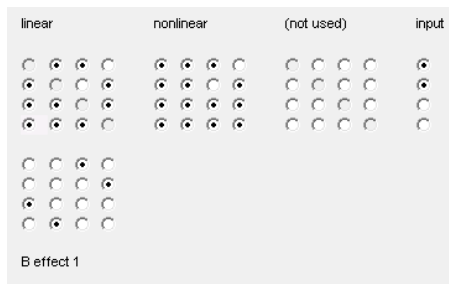


Figure 43.4: Connectivity configuration for the FnBn model.

by observing the model fit in the output window. Note that in DCM-IR there is no difference between the hidden states and the predicted responses because the dynamics of the hidden states fit directly the time course of frequency modes (shown as dotted lines in the middle plot). This is different from DCM for ERP where the hidden states correspond to neural dynamics and a subset of the hidden states (activation of pyramidal cells) are projected via the forward model to generate predictions of sensor data. In the MATLAB command window, you will see each iteration print an expectation-maximization iteration number, free energy  $F$ , and the predicted and actual change of  $F$  following each iteration step. At convergence, DCM saves the results in a DCM file, by default named `DCM_*.mat` where `*` corresponds to the name of the original SPM MEG file you specified. You can save to a different name, e.g. if you are estimating multiple models, by pressing 'save' at the top of the GUI and writing to a different name.

## 43.6 Results

After estimation is finished, you can assess the results by choosing from the pull-down menu at the bottom (middle).

### 43.6.1 Frequency modes

This will display the frequency profiles of the modes, identified using singular value decomposition of spectral dynamics in source space (over time and sources).

### 43.6.2 Time modes

This will display the observed time courses of the frequency modes (dashed lines) and the model predictions (solid lines). Here you can also see whether the activity picked up by the minor modes is noise, which is helpful for optimizing the number of modes.

### 43.6.3 Time-Frequency

This will display the observed time-frequency power data for all pre-specified sources (upper panel) and the fitted data features (lower panel).

### 43.6.4 Coupling (A-Hz)

This will display the coupling matrices representing the coupling strength from source to target frequencies. These matrices are obtained by multiplying the between-mode matrices estimated with the frequency profiles of the modes (see [18]). The arrangement of the matrices corresponds to arrangements of the buttons in the connectivity matrices above.

### 43.6.5 Coupling (B-Hz)

This presentation of results is similar to the above and reports modification of coupling by condition (eg. in our example it shows which frequency couplings are different for faces as opposed to scrambled faces).

### 43.6.6 Coupling (A-modes)

This will display the coupling matrices between modes and the conditional probabilities that the coefficients are different from zero. This representation is useful for diagnostics when something is wrong with the inversion, but the physiological interpretation is less straightforward.

### 43.6.7 Coupling (B-Hz)

This presentation is similar to the above and reports the modification of coupling by condition.

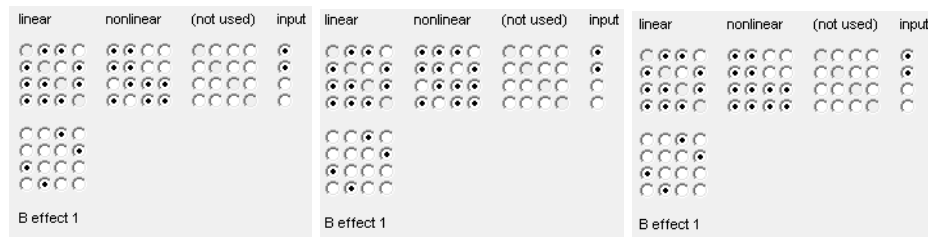


Figure 43.5: Connectivity configurations for the alternative models. Left to right: *FBLI*, *FBLn*, *FnBl*.

### 43.6.8 Input (C-Hz)

This shows the frequency profiles of the inputs estimated. This is again a multiplication between the mode-specific coefficients and the frequency profiles of the modes.

### 43.6.9 Input (u-ms)

This shows the time courses of the inputs.

### 43.6.10 Dipoles

This shows the positions of the sources as specified in the “Electromagnetic model” section.

### 43.6.11 Save as img

Here you can save the cross-frequency coupling matrices as images. If you are analyzing a group of subjects you can then enter these images into parametric statistical tests to find common features in coupling and coupling changes across subjects. The image names will include identifiers like “A12” or “B31” which relate to the source connection matrices; either the basic (A) or experimental effects (B).

## 43.7 Model comparison

You can now compare the fully nonlinear model with alternative models (eg. those shown in Figure 43.3). You can start by saving the DCM you have already specified under a different name using the *Save* button. Then just modify the connectivity matrices and reinvert the DCM by pressing the “Estimated” button (but not using previous posterior or prior estimates). As an exercise, you can specify the other models from Figure 43.3 yourself. If in doubt look at Figure 43.5 for the three alternative models. Once you have specified and inverted the three additional models, you can perform Bayesian model comparison.

Press the BMS button. This will open the SPM batch tool for model selection. Specify a directory to write the output file to. For the “Inference method” select “Fixed effects” (see [91] for additional explanations). Then click on “Data” and in the box below click on “New: Subject”. Click on “Subject” and in the box below on “New: Session”. Click on models and in the selection window that comes up select the DCM mat files for all the models (remember the order in which you select the files as this is necessary for interpreting the results). Then run the model comparison by pressing the green “Run” button. You will see, at the top, a bar plot of the log-model evidences for all models 43.6. At the bottom, you will see the posterior probability, for each model, given the data. By convention, a model can be said to be the best among a selection of other models, with strong evidence, if its log-model evidence exceeds all other log-model evidences by at least 3. In our case the FnBn model is superior to the other models as was found in the original paper [17] for a different group of subjects.

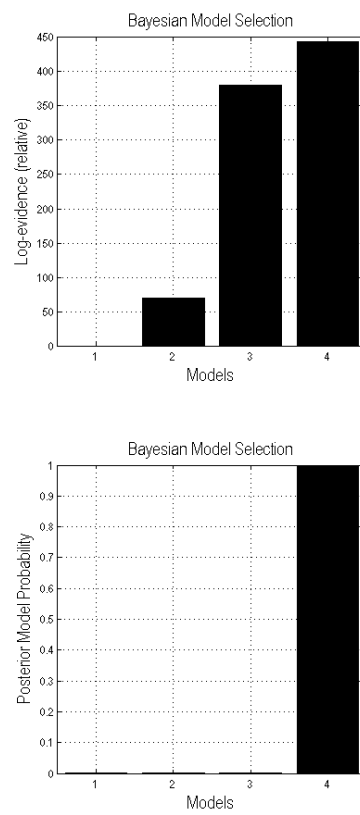


Figure 43.6: Bayesian comparison of the four DCM-IR models shown in Figure 43.3.

## Chapter 44

# DCM for Phase Coupling

This chapter presents an extension of the Dynamic Causal Modelling (DCM) framework to the analysis of phase-coupled data. A weakly coupled oscillator approach is used to describe dynamic phase changes in a network of oscillators. The influence that the phase of one oscillator has on the change of phase of another is characterised in terms of a Phase Interaction Function (PIF) as described in [84]. SPM supports PIFs specified using arbitrary order Fourier series. However, to simplify the interface, one is restricted to simple sinusoidal PIFs with the GUI.

### 44.1 Data

We will use the merged epoched MEG face-evoked dataset<sup>1</sup> saved in the files:

```
cdbespm12_SPM_CTF_MEG_example_faces1_3D.mat  
cdbespm12_SPM_CTF_MEG_example_faces1_3D.dat
```

DCM-Phase requires a head model and coregistration. If you have been following the previous chapters of this tutorial, these should already be available in the dataset. Otherwise, you should perform the 'Prepare' and '3D Source reconstruction' steps described earlier in the chapter, with the latter comprising the MRI, Co-register, Forward and Save sub-steps.

### 44.2 Getting Started

You need to start SPM and toggle “EEG” as the modality (bottom-right of SPM main window), or start SPM with `spm eeg`. In order for this to work you need to ensure that the main SPM directory is on your MATLAB path. After calling `spm eeg`, you see SPM’s graphical user interface, the top-left window. The button for calling the DCM-GUI is found in the second partition from the top, on the right hand side. When pressing the button, the GUI pops up (Figure 43.1).

### 44.3 Data and design

You should switch the DCM model type to “PHASE” which is the option for DCM-Phase. Press “new data” and select the data file `cdbespm12_SPM_CTF_MEG_example_faces1_3D.mat`. This is an epoched data file with multiple trials per condition. On the right-hand side enter the trial indices

```
1 2
```

for the 'face' and 'scrambled' evoked responses (we will model both trial types). The box below this list allows for specifying experimental effects on connectivity. Enter

```
1 0
```

---

<sup>1</sup>Multimodal face-evoked dataset: <http://www.fil.ion.ucl.ac.uk/spm/data/mmfaces/>

in the first row of the box. This means that “face” trial types can have different connectivity parameters than “scrambled” trial types. If we now click somewhere outside the box, a default name will be assigned to this effect - “effect1”. It will appear in the small text box next to the coefficients box. It is possible to change this name to something else e.g. “face”. Now we can select the peristimulus time window we want to model. Set it to:

```
1 300
```

ms. Select 1 for “detrend”, to remove the mean from each data record. The `sub-trials` option makes it possible to select just a subset of trials for the analysis (select 2 for every second trial, 3 - for every third etc.). This is useful because DCM-Phase takes quite a long time to invert for all the trials and you might want to first try a smaller subset to get an idea about the possible results. Here we will assume that you used all the trials (sub-trials was set to 1). You can now click on the > (forward) button, which will bring you to the next stage *electromagnetic model*. From this part, you can press the red < button to get back to the data and design part.

## 44.4 Electromagnetic model

With DCM-Phase, there are two options for how to extract the source data. Firstly, you can use 3 orthogonal single equivalent current dipoles (ECD) for each source, invert the resulting source model to get 3 source waveforms and take the first principal component. This option is suitable for multichannel EEG or MEG data. Alternatively, you can treat each channel as a source (LFP option). This is appropriate when the channels already contain source data either recorded directly with intracranial electrodes or extracted (e.g. using a beamformer).

Note that a difference to DCM for evoked responses is that the parameters of the spatial model are not optimized. This means that DCM-Phase (like DCM-IR) will project the data into source space using the spatial locations you provide.

We will use the ECD option and specify just two source regions. This requires specifying a list of source names in the left large text box and a list of MNI coordinates for the sources in the right large text box. Enter the following in the left box:

```
LOFA
LFFA
```

Now enter in the right text box:

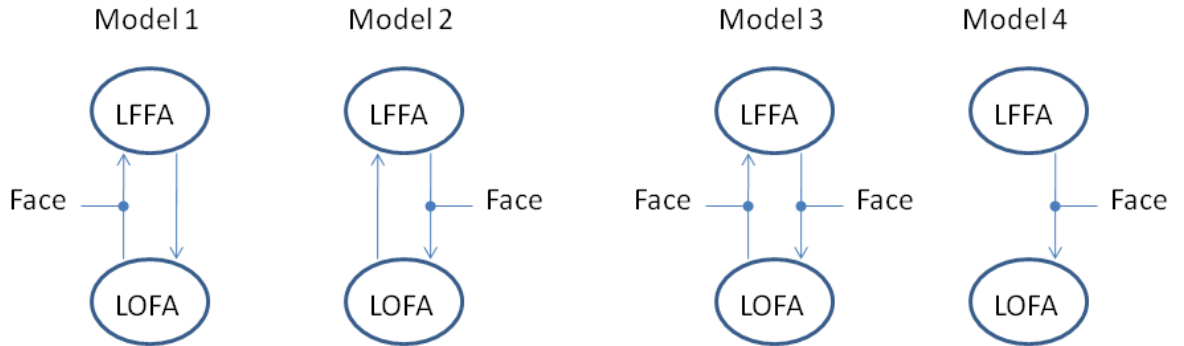
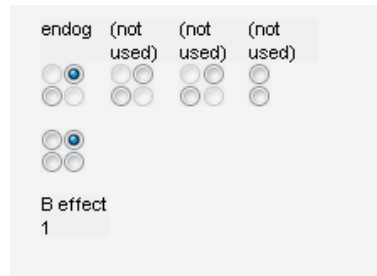
```
-39 -81 -15
-39 -51 -24
```

These correspond to left Occipital Face Area, and left Fusiform Face Area. The onset-parameter is irrelevant for DCM-Phase. Now hit the > (forward) button and proceed to the *neuronal model*. Generally, if you have not used the input dataset for 3D source reconstruction before you will be asked to specify the parameters of the head model at this stage.

## 44.5 Neuronal model

We will now define a coupled oscillator model for investigating network synchronization of alpha activity. To this end, we first enter the values 8 and 12 to define the frequency window. The wavelet number is irrelevant for DCM-Phase. After source reconstruction (using a pseudo-inverse approach), source data is bandpass filtered and then the Hilbert transform is used to extract the instantaneous phase. The DCM-Phase model is then fitted used standard routines as described in [84].

Figure 44.1 shows the four models we will apply to the M/EEG data. We will first fit model 4. This model proposes that alpha activity in region LOFA changes its phase so as to synchronize with activity in region LFFA. In this network LFFA is the master and LOFA is the slave. Moreover, the connection from LFFA to LOFA is allowed to be different for scrambled versus unscrambled faces.

Figure 44.1: *Four different DCM-Phase models*Figure 44.2: *Radio button configurations for DCM-Phase model 4*

The connectivity for Model 4 can be set up by configuring the radio buttons as shown in Figure 44.2. You can now press the **Invert DCM** button. It can take up to an hour to estimate the model parameters depending on the speed of your computer.

## 44.6 Results

After estimation is finished, you can assess the results by choosing from the pull-down menu at the bottom (middle). The **Sin(Data)-Region i** option will show the sin of the phase data in region  $i$ , for the first 16 trials. The blue line corresponds to the data and the red to the DCM-Phase model fit. The **Coupling(As)** and **Coupling(Bs)** buttons display the estimated endogenous and modulatory activity shown in Figure 44.3.

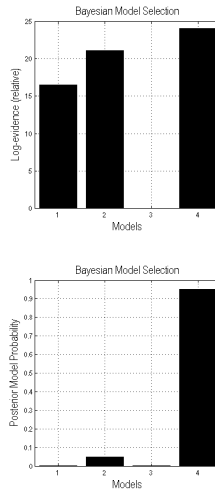
If one fits all the four models shown in Figure 44.1 then they can be formally compared using Bayesian Model Selection. This is implemented by pressing the **BMS** button. You will need to first create a directory for the results to go in e.g. **BMS-results**. For 'Inference Method' select **FFX** (the **RFX** option is only viable if you have models from a group of subjects). Under 'Data', Select 'New Subject' and under 'Subject' select 'New Session'. Then under 'Models' select the **DCM.mat** files you have created. Then press the green play button. This will produce the results plot shown in Figure 44.4. This leads us to conclude that LFFA and LOFA act in master slave arrangement with LFFA as the master.

## 44.7 Extensions

In the DCM-Phase model accessible from the GUI, it is assumed that the phase interaction functions are of simple sinusoidal form ie.  $a_{ij} \sin(\phi_j - \phi_i)$ . The coefficients  $a_{ij}$  are the values shown in the endogenous parameter matrices in eg. Figure 44.3. These can then be changed by an amount  $b_{ij}$  as shown in the modulatory parameter matrices. It is also possible to specify

changes in coupling (B)		face
Posterior mean		
LOFA	LFFA	
LOFA	0.000	-0.517
LFFA	0.000	0.000
Posterior probability (B <sub>ij</sub> > 0)		
LOFA	LFFA	
LOFA	NaN	0.965
LFFA	NaN	NaN

Figure 44.3: The figure shows the estimated parameters for endogenous coupling (left column) and modulatory parameters (right column) for the 4th DCM.



Data identity has not been verified

Figure 44.4: Bayesian comparison of the four DCM-Phase models shown in Figure 44.1.



and estimate DCM-Phase models using matlab scripts. In this case it is possible to specify more generic phase interaction functions, such as arbitrary order Fourier series. Examples are given in [84].



## Chapter 45

# DCM for Cross Spectral Densities: Anaesthesia Depth in Rodent Data

### 45.1 Overview

This chapter describes the analysis of a 2-channel Local Field Potential (LFPs) data set using dynamic causal modelling. The LFPs were recorded from a single rodent using intracranial electrodes [77]. We thank Marc Tittgemeyer for providing us with this data. The theory behind DCM for cross spectral densities (DCM-CSD) is described in [29]. This DCM is a generalization of DCM for Steady State Responses to the complex domain [78]. The generative model now reports coherence and signal covariance as well as complex spectral densities (from which the former are derived).

We measured local field potentials from primary (A1) and secondary auditory (A2) cortex in a rodent following the application of four different doses of the anaesthetic agent Isoflurane; 1.4, 1.8, 2.4 and 2.8%. The rodent was presented with a white noise auditory input for several minutes at each anaesthetised level and time series recordings were obtained for the entire epoch. We performed a DCM analysis to ask whether changes in neuronal activity induced by increasing levels of Isoflurane are best accounted for by *either* extrinsic *or* intrinsic changes in connectivity.

We demonstrate in this chapter the consistency of the model comparison and conditional parameter estimates across different population models. In particular we modeled the CSD as the output of a two region network comprising either “CMC” or “NMDA” – type neural mass models.

The CMC-type neural mass model comprises four subpopulations. It is a refinement of the Jansen and Rit convolution models that explicitly accommodates the neuronal sources of forward and backward connections in cortical hierarchies [96]. These are distinct superficial and deep pyramidal cell populations respectively that, crucially, may exhibit different spectral outputs. The CMC thus utilizes different types of subpopulations as the source of forward and backward connections. For the forward connections superficial pyramidal cells excite stellate cells and deep pyramidal neurons, while the backward connections inhibit superficial pyramidal cells and inhibitory interneurons (see `spm_fx_cmc`). From the graphical user interface trial specific effects can be selected for extrinsic connections or intrinsic connections, for the CMC case the intrinsic connection that is modulated is an inhibitory gain parameter on superficial pyramidal cells. The smaller this value, the greater the gain on this cell population due to the modulation.

The NMDA model uses an architecture comprising three subpopulations, each assigned to a particular cortical layer. An inhibitory interneuron subpopulation occupies agranular layers. This receives inputs from excitatory pyramidal cells, also in agranular layers which are, in turn, driven by excitatory spiny cells in the granular layer; layer IV. These three subpopulations are connected with intrinsic coupling parameters (which can be found in `spm_fx_mnn_nmda`). Forward connections correspond to afferent pyramidal axons and synapse on layer IV stellate cells, while backward afferents impinge upon pyramidal and inhibitory interneurons outside of layer IV. Lateral, inter-hemispheric connections are modelled with a postsynaptic response that is elicited in all layers. The model employs Morris Lecar-type differential equations to describe the time

evolution of a neuronal ensemble. In this model, cells possess AMPA, GABAA, and NMDA-like receptor dynamics, with appropriate ion-channel time constants and a voltage dependent switch for the NMDA channel [76]. From the graphical user interface trial specific effects can be selected for extrinsic connections or intrinsic connections, for the NMDA case the intrinsic connection that is modulated is an excitatory connection operating on all intrinsic excitatory connections. The greater this value, the greater the excitatory effect due to the modulation.

## 45.2 Main Results

Using Bayesian model comparison we found very strong evidence (Bayes Factor<sub>1,2</sub> > 100) in favour of a model comprising a network of two neural masses connected by forward and backward connections from A1 to A2 and A2 to A1, where the effect of anesthetic was best explained by changes in *intrinsic* connections (model 2). This outperformed a model comprising the same two neural masses with the same extrinsic connections but where the effect of isoflurane was expressed as a modulatory (B) effect on extrinsic connections between regions (model 1). This result was obtained for both types of neural mass models used.

## 45.3 Using the Graphical User Interface to Obtain those Results

In what follows, these results will be recreated step-by-step using SPM12. To proceed with the data analysis, first download the data set from the SPM website<sup>1</sup>. The data comprises a data file called `dLFP_white_noise_r24_anaes.dat` and its corresponding MAT-file `dLFP_white_noise_r24_anaes.mat`. This has been converted from ASCII data using `spm_lfp_txt2mat_anaes.m` also on the website and subsequently downsampled to 125 Hz. The conversion script can be altered to suit your own conditions/sampling parameters.

### 45.3.1 The data

- To check data parameters after conversion using ASCII files: in the SPM M/EEG GUI press Display/M/EEG.
- In our data set we can see there are five trials: four depths of anaesthetic: Iso14, Iso18, Iso24 and Iso28 and one awake trial awake.
- We are going to employ a 5 sec window of data (without ripples) for the DCM this data is from 25000 to 30000 ms.
- We are now ready to begin the DCM analysis. To open the DCM GUI press DCM in the SPM M/EEG GUI.

### 45.3.2 Dynamic Causal Modelling of Cross Spectral Densities

- Before you begin any DCM analysis you must decide on three things: the data feature from your time series, the model you wish to use and the hypothesis you wish to test.
- For our long time series we will examine the steady state and so in the top panel of the DCM GUI select CSD in the data drop-down menu.
- Next in the second drop down menu we select our model. For our first analysis we select the CMC model (we can toggle this button to select other types of neural masses later.) Then we are ready to load our data: press new data and select the file `dLFP_white_noise_r24_anaes.mat`.
- Press the red arrow to move forward.

<sup>1</sup>Anaesthesia Depth in Rodent Dataset: [http://www.fil.ion.ucl.ac.uk/spm/data/dcm\\_csd/](http://www.fil.ion.ucl.ac.uk/spm/data/dcm_csd/)

- The second panel allows you to specify the data and design. We will use 5 seconds of data towards the end of the recording for our analysis. To specify this timing enter 25000 and 30000 in the time window.
- Next we select the detrending parameters which we set to 1 for detrend, 1 for subsample (as the data has already been downsampled) and 2 for the modes (in this case this is the same as the number of channels) using the drop down menus.
- We can then specify which trials we want to use. Since we are interested in the anaesthetized trials we enter [1 2 3 4] under the trials label and Iso 1.8 Iso 2.4 Iso 2.8 are our three effects in the “between trial effects” panel. Next we specify the design matrix. This is entered numerically in the large panel. Since we have 4 trials and 3 between trial effects (one less) we enter a matrix with rows: [0 1 0 0] (row 1), [0 0 1 0] (row 2) and [0 0 0 1] (row 3). This will allow us to examine “main effect” differences between the four conditions.
- Press the red arrow to move forward.
- The third panel contains the spec for the electromagnetic model. This is very simple for local field potential recordings. In the drop down menu select LFP. In the source names panel, enter A1 and A2. You are finished.
- Press the red arrow to move forward.
- At this point all that is left to specify is the neuronal model in terms of its connectivity. We wish to compare two different models so we can save the specifications so far using the save button and reload the above specs for both neuronal models.
- To specify the neuronal model, load the DCM (that you just saved) as it has been so far specified.
- Our first model is the extrinsic modulation model.
- So we specify forward connections from A1 to A2 and backward connections from A2 to A1.
- We finally specify the B effects where we enter our hypothesis of connectivity changes between trial 1 (Iso1.4%) trial 2 (Iso1.8%) trial 3 (Iso2.4%) and trial 4 (Iso2.8%). Changes will be specified relative to trial 1.
- We enter the off diagonal entries to correspond to forward connections (as entered in the above panel) to specify extrinsic connectivity changes between A1 and A2 due to (anaesthetic) condition.
- Finally we enter the frequencies we are interested in: we will fit frequencies from 4 to 48 Hz.
- To invert the model press the “invert DCM” button.
- Repeat the procedure after loading the saved specs and repeating for new neuronal models as per figure 45.2. Here we enter our alternative hypothesis (model 2) and fit a second model where instead of extrinsic connectivity changes, the isoflurane related-changes are generated by connectivity differences within a region — we call this the intrinsic modulation model.
- This is specified by selecting the diagonal elements of the B-matrices.

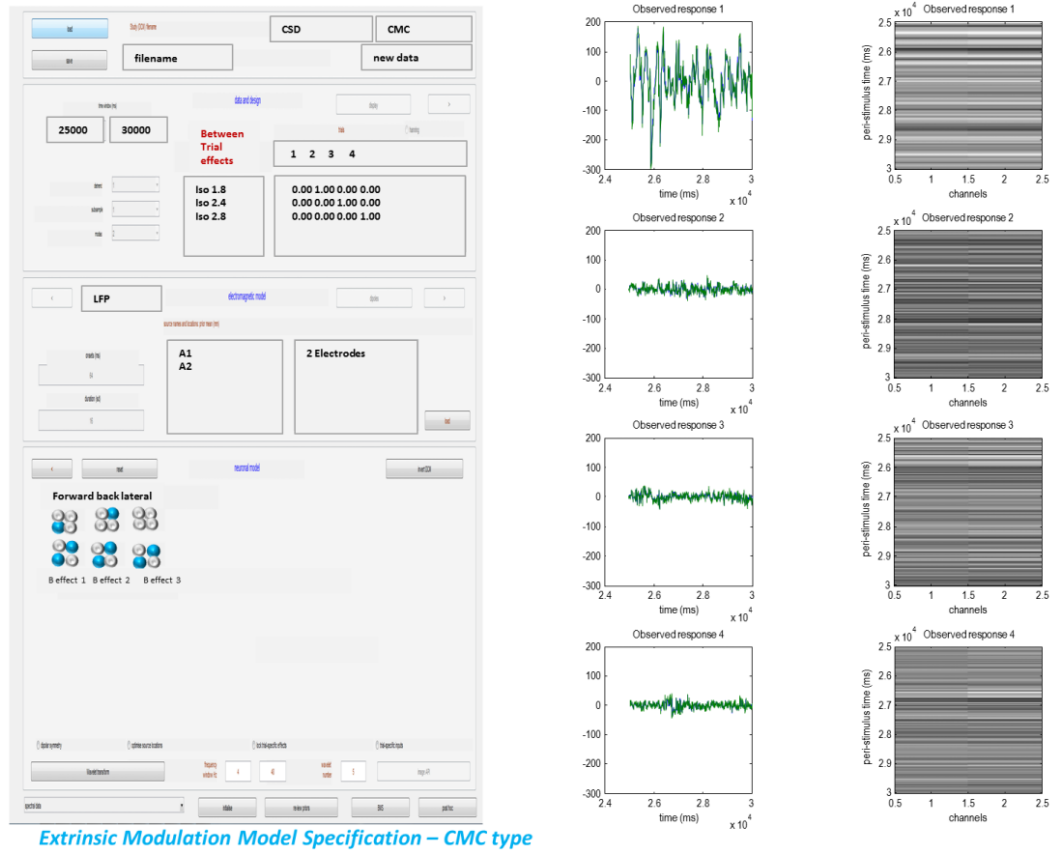
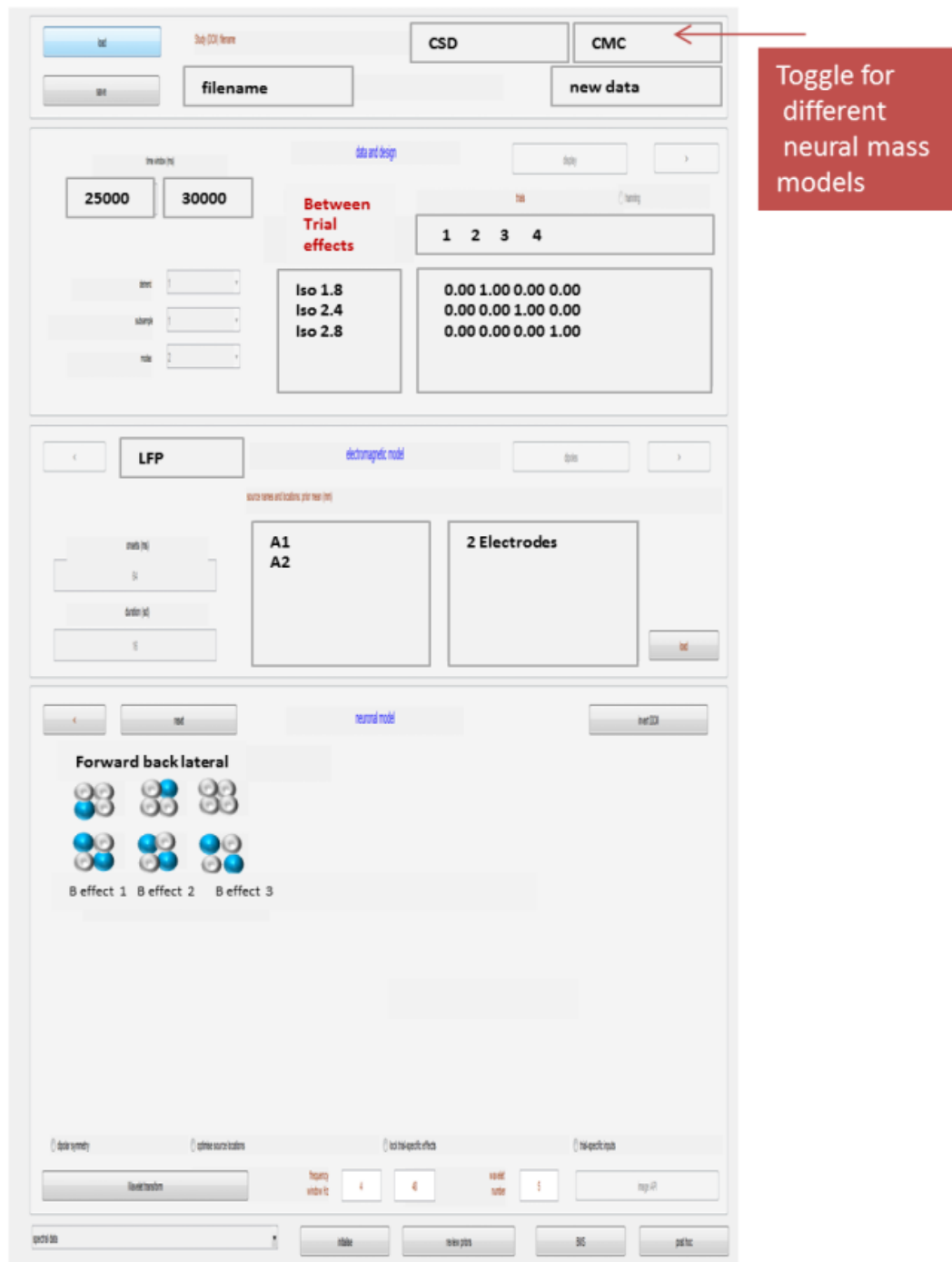


Figure 45.1: Left: Graphical User Interface to specify model 1: An extrinsic modulation by Isolurance. Here the B effects are clicked along the off-diagonal positions, representing changes in forward and backward extrinsic connections across the three trial types. Right: Data time series for two intracranial electrode recordings from 25000 to 30000 msec. Green and Blue lines represent different electrodes, panel rows correspond to the different trials – here recordings made at different depths of anaesthesia: trial 1 = 1.4% isoflurane, trial 2 = 1.8%, trial 3 = 2.4% and trial 4 = 2.8%.



### Intrinsic Modulation Model Specification – CMC type

Figure 45.2: GUI illustrating how to specify model 2 – the intrinsic modulation model. Here the B effects are clicked along diagonal position, representing changes in local (within-region) coupling. The top panel contains the neural model specification. This can be toggled to select different neural mass models as the basis for the DCM – eg. LFP or CMC type.

Extrinsic (Model 1) or Intrinsic (Model 2)  
Modulation due to Isoflurane based on CMC neural masses

Extrinsic (Model 1) or Intrinsic (Model 2)  
Modulation due to Isoflurane based on NMDA neural masses

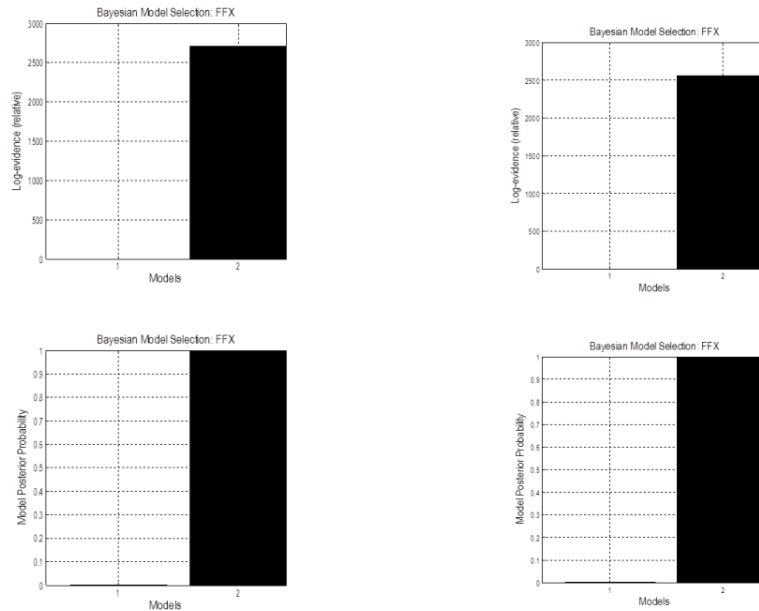


Figure 45.3: *Top: a bar plot of the log-model evidences for all models. Bottom: conditional probability for each model assuming equal priors on model space.*

### 45.3.3 Comparing models using Bayesian Model Selection

- Once both models have run, we compare their evidences to find the best or winning model. To do this press the BMS button. This will open the SPM batch tool for model selection. Specify a directory to write the output file to. For the Inference method select Fixed effects (see [91] for additional explanations). Then click on Data and in the box below click on New: Subject. Click on Subject and in the box below on New: Session. Click on models and in the selection window that comes up select the DCM mat files for all the models (remember the order in which you select the files as this is necessary for interpreting the results). Then run the model comparison by pressing the green Run button. You will see at the top, a bar plot of the log-model evidences for all models (Figure 45.3). The bottom panel displays the conditional probability, for each model assuming equal priors on model space. By convention, a model can be said to be the best among a selection of other models, with strong evidence, if its log-model evidence exceeds all other log-model evidences by at least 3. You can also compare model evidences manually if you load the DCMs into MATLAB's workspace and find the evidence in the structure under DCM.F.
- For our example we see that there is strong model in favor of model 2 (log Bayes Factor  $>2500$ ); ie. Isoflurane effects are better explained by a modulation of intrinsic connections.
- We repeated the steps above and inverted two models (again with either extrinsic or intrinsic modulations) using the CMC and the NMDA neural masses also. These yielded similar results in favor of model 2 – an intrinsic connectivity effect.
- Once inverted a results tab appears which allows you to view the fits to the spectral data, posterior parameter estimates, coherence and covariance (in channel and source space) and the transfer functions between regions. You can also examine the direction of the modulating effects under trial specific effects.



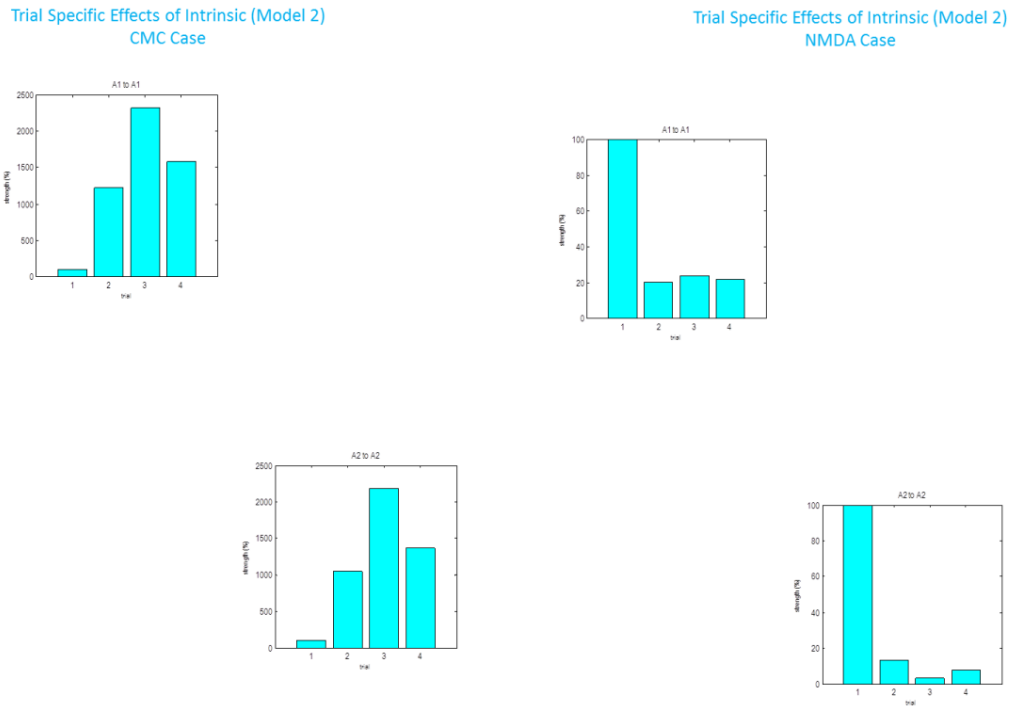


Figure 45.4:

- From our winning CMC model we examined the trial specific effects and found a large increase for trial 2 relative to trial 1 in A1 (top left panel). Remember for the CMC case this represents a decreases in pyramidal cell gain  $\hat{A}_S$  and so is a net inhibitory effect, consistent with the physiological effects of Isoflurane. The effect was even larger for trial 3 compared to trial 1 and decreased to a lower level for trial 4 (as reported in [75]). The effects were similar in A2 (bottom right panel). We found a very similar trial specific effect in the NMDA case, but here the parameter decreases as it represents a positive modulation of an excitatory connection. In other words the effect of increasing isoflurane levels inhibitory in a non-linear fashion (saturating at a level of 2.4%; trial 3).



# Chapter 46

## Using Dartel

Dartel<sup>1</sup> is a suite of tools for achieving more accurate inter-subject registration of brain images. It consists of several thousand lines of code. Because it would be a shame if this effort was wasted, this guide was written to help encourage its widespread use. Experience at the FIL would suggest that it offers very definite improvements for VBM studies – both in terms of localisation<sup>2</sup> and increased sensitivity<sup>3</sup>.

### 46.1 Using Dartel for VBM

The following procedures could be specified one at a time, but it is easier to use the batching system. The sequence of jobs (use the *Batch* button to start the batch manager) would be:

- **Module List**

- **SPM→Spatial→Segment**: To generate the roughly (via a rigid-body) aligned grey and white matter images of the subjects.
- **SPM→Tools→Dartel Tools→Run Dartel (create Template)**: Determine the nonlinear deformations for warping all the grey and white matter images so that they match each other.
- **SPM→Tools→Dartel Tools→Normalise to MNI Space**: Actually generate the smoothed “modulated” warped grey and white matter images.

Segmentation can require quite a lot of memory, so if you have large images (typically greater than about  $256 \times 256 \times 150$ ) and trying to run it on a 32 bit computer or have relatively little memory installed, then it may throw up an out of memory error.

Further details of the steps are described next.

#### 46.1.1 Using Spatial→Segment

*Note: This subsection will be elaborated on later.*

The first step is to classify T1-weighted scans<sup>4</sup> of a number of subjects into different tissue types via the Segmentation routine in SPM, which can be found under SPM→Spatial→Segment. With this option, the “imported” tissue class images (usually rc1.nii and rc2.nii) would be generated directly. It is also suggested that *Native Space* versions of the tissues in which you are interested are also generated. For VBM, these are usually the c1\*.nii files, as it is these images that will eventually be warped to MNI space. Both the imported and native tissue class image sets can be specified via the Native Space options of the user interface.

---

<sup>1</sup>Dartel stands for “Diffeomorphic Anatomical Registration Through Exponentiated Lie algebra”. It may not use a true Lie Algebra, but the acronym is a nice one.

<sup>2</sup>Less smoothing is needed, and there are fewer problems relating to how to interpret the differences.

<sup>3</sup>More sensitivity could mean that fewer subjects are needed, which should save shed-loads of time and money.

<sup>4</sup>Other types of scan may also work, but this would need some empirical exploration.

Segmentation can require quite a lot of memory, so if you have large images (typically greater than about  $256 \times 256 \times 150$ ) and trying to run it on a 32 bit computer or have relatively little memory installed, then it may throw up an out of memory error.

### 46.1.2 Using Dartel Tools→Run Dartel (create Template)

The output of the previous step(s) are a series of rigidly aligned tissue class images (grey matter is typically encoded by rc1\*.nii and white matter by rc2\*.nii – see Fig 46.2). The headers of these files encode two affine transform matrices, so the Dartel tools are still able to relate their orientations to those of the original T1-weighted images. The next step is to estimate the nonlinear deformations that best align them all together. This is achieved by alternating between building a template, and registering the tissue class images with the template, and the whole procedure is very time consuming. Specify *SPM→Tools→Dartel Tools→Run Dartel (create Template)*.

- **Run Dartel (create Template)**

- **Images**

- \* **Images:** Select all the rc1\*.nii files generated by the import step.

- \* **Images:** Select all the rc2\*.nii files, in the same subject order as the rc1\*.nii files. The first rc1\*.nii is assumed to correspond with the first rc2\*.nii, the second with the second, and so on.

- **Settings:** Default settings generally work well, although you could try changing them to see what happens. A series of templates are generated called Template\_basename\_0.nii, Template\_basename\_1.nii etc. If you run multiple Dartel sessions, then it may be a good idea to have a unique template basename for each.

The procedure begins by computing an initial template from all the imported data. If u\_rc1\*.nii files exist for the images, then these are treated as starting estimates and used during the creation of the initial template. If any u\_rc1\*.nii files exist from previous attempts, then it is usually recommended that they are removed first (this sets all the starting estimates to zero). Template generation incorporates a smoothing procedure, which may take a while (several minutes). Once the original template has been generated, the algorithm will perform the first iteration of the registration on each of the subjects in turn. After the first round of registration, a new template is generated (incorporating the smoothing step), and the second round of registration begins. Note that the earlier iterations usually run faster than the later ones, because fewer “time-steps” are used to generate the deformations. The whole procedure takes (in the order of) about a week of processing time for 400 subjects.

The end result is a series of templates (see Fig 46.1), and a series of u\_rc1\*.nii files. The first template is based on the average<sup>5</sup> of the original imported data, where as the last is the average of the Dartel registered data. The u\_rc1\*.nii files are flow fields that parameterise the deformations. Note that all the output usually contains multiple volumes per file. For the u\_rc1\*.nii files, only the first volume is visible using the Display or Check Reg tools in SPM. All volumes within the template images can be seen, but this requires the file selection to be changed to give the option of selecting more than just the first volume (in the file selector, the widget that says “1” should be changed to “1:2”).

### 46.1.3 Using Dartel Tools→Normalise to MNI Space

The next step is to create the Jacobian scaled (“modulated”) warped tissue class images, by selecting *SPM→Tools→Dartel Tools→Normalise to MNI Space*. The option for spatially normalising to MNI space automatically incorporates an affine transform that maps from the population average (Dartel Template space) to MNI space, as well as incorporating a spatial smoothing step.

- **Normalise to MNI Space**

---

<sup>5</sup>They are actually more similar to weighted averages, where the weights are derived from the Jacobian determinants of the deformations. There is a further complication in that a smoothing procedure is built into the averaging.

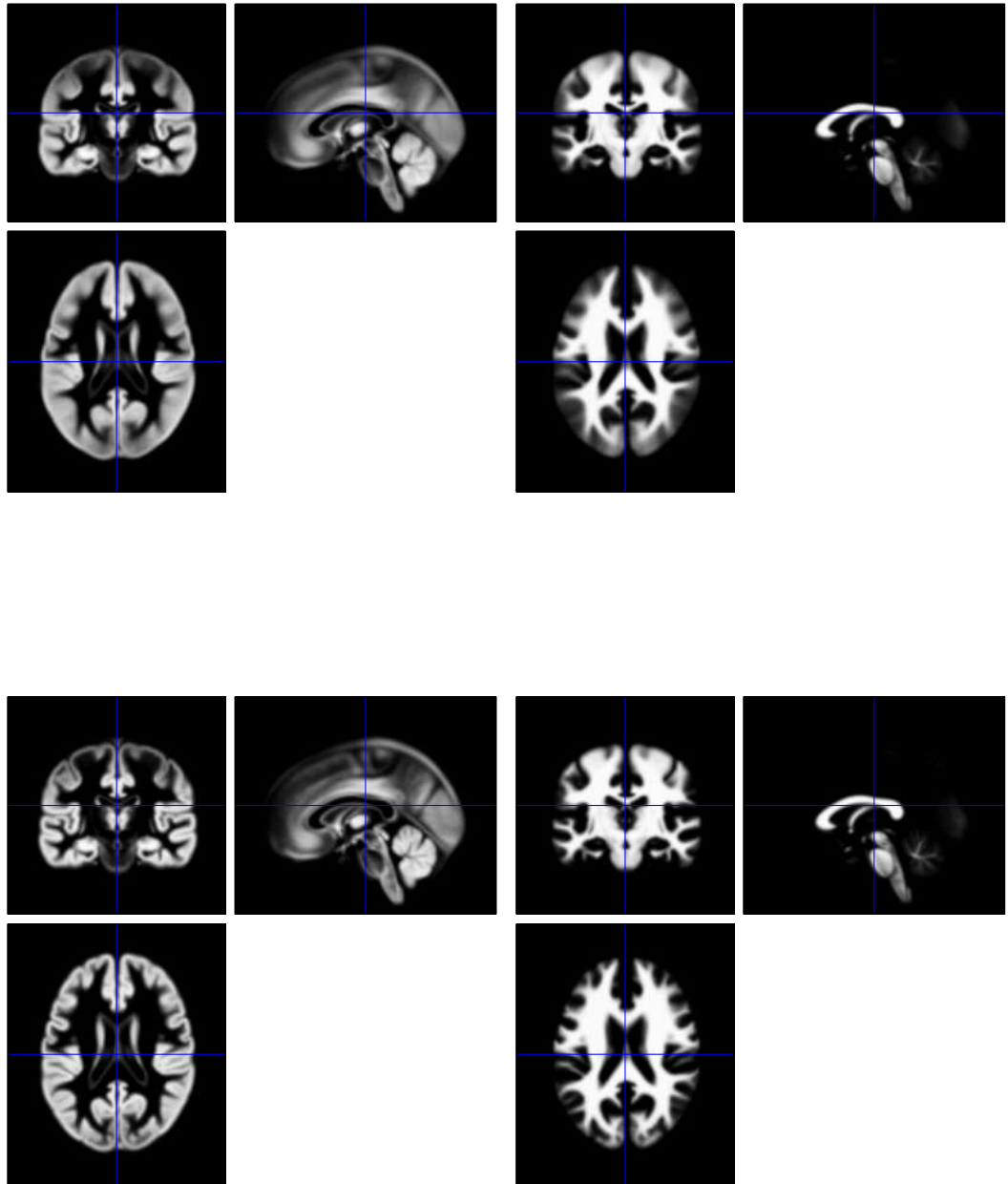


Figure 46.1: Different stages of template generation. Top row: an intermediate version of the template. Bottom row: the final template data.

- **Dartel Template:** Specify the last of the series of templates that was created by *Run Dartel (create Template)*. This is usually called *Template\_6.nii*. Note that the order of the  $N$  volumes in this template should match the order of the first  $N$  volumes of the *toolbox/Dartel/TPM.nii* file.
- **Select according to** either *Few Subjects* or *Many Subjects*. For VBM, the *Many Subjects* option would be selected.
  - \* **Flow Fields:** Specify the flow fields (*u\_rc1\*.nii*) generated by the nonlinear registration.
  - \* **Images:** You may add several different sets of images.
    - **Images:** Select the *c1\*.nii* files for each subject, in the same order as the flow fields are selected.
    - **Images:** This is optional, but warped white matter images can also be generated by selecting the *c2\*.nii* files.
- **Voxel sizes:** Specify the desired voxel sizes for the spatially normalised images (NaN, NaN, NaN gives the same voxel sizes as the Dartel template).
- **Bounding box:** Specify the desired bounding box for the spatially normalised images (NaN, NaN, NaN; NaN NaN NaN gives the same bounding box as the Dartel template).
- **Preserve:** Here you have a choice of *Preserve Concentrations* (ie not Jacobian scaled) or *Preserve Amount* (Jacobian scaled). The *Preserve Amount* would be used for VBM, as it does something similar to Jacobian scaling (modulation).
- **Gaussian FWHM:** Enter how much to blur the spatially normalised images, where the values denote the full width at half maximum of a Gaussian convolution kernel, in units of mm. Because the inter-subject registration should be more accurate than when done using other SPM tools, the FWHM can be smaller than would be otherwise used. A value of around 8mm (ie [8, 8, 8]) should be about right for VBM studies, although some empirical exploration may be needed. If there are fewer subjects in a study, then it may be advisable to smooth more.

The end result should be a bunch of *smwc1\*.nii* files<sup>6</sup> (possibly with *smwc2\*.nii* if white matter is also to be studied).

The final step is to perform the statistical analysis on the preprocessed data (*smwc1\*.nii* files), which should be in MNI space. The next section says a little about how data from a small number of subjects could be warped to MNI space.

## 46.2 Spatially normalising functional data to MNI space

Providing it is possible to achieve good alignment between functional data from a particular subject and an anatomical image of the same subject (distortions in the fMRI may prevent accurate alignment), then it may be possible to achieve more accurate spatial normalisation of the fMRI data using Dartel. There are several advantages of having more accurate spatial normalisation, especially in terms of achieving more significant activations and better localisation.

The objectives of spatial normalisation are:

- To transform scans of subjects into alignment with each other. Dartel was developed to achieve better inter-subject alignment of data.
- To transform them to a standard anatomical space, so that activations can be reported within a standardised coordinate system. Extra steps are needed to achieve this aim.

<sup>6</sup>The actual warping of the images is done slightly differently, with the aim that as much of the original signal is preserved as possible. This essentially involves pushing each voxel from its position in the original image, into the appropriate location in the new image - keeping a count of the number of voxels pushed into each new position. The procedure is to scan through the original image, and push each voxel in turn. The alternative (older way) was to scan through the spatially normalised image, filling in values from the original image (pulling the values from the original). The results of the pushing procedure are analogous to Jacobian scaled ("modulated") data. A minor disadvantage of this approach is that it can introduce aliasing artifacts (think stripy shirt on TV screen) if the original image is at a similar - or lower - resolution to the warped version. Usually, these effects are masked by the smoothing.

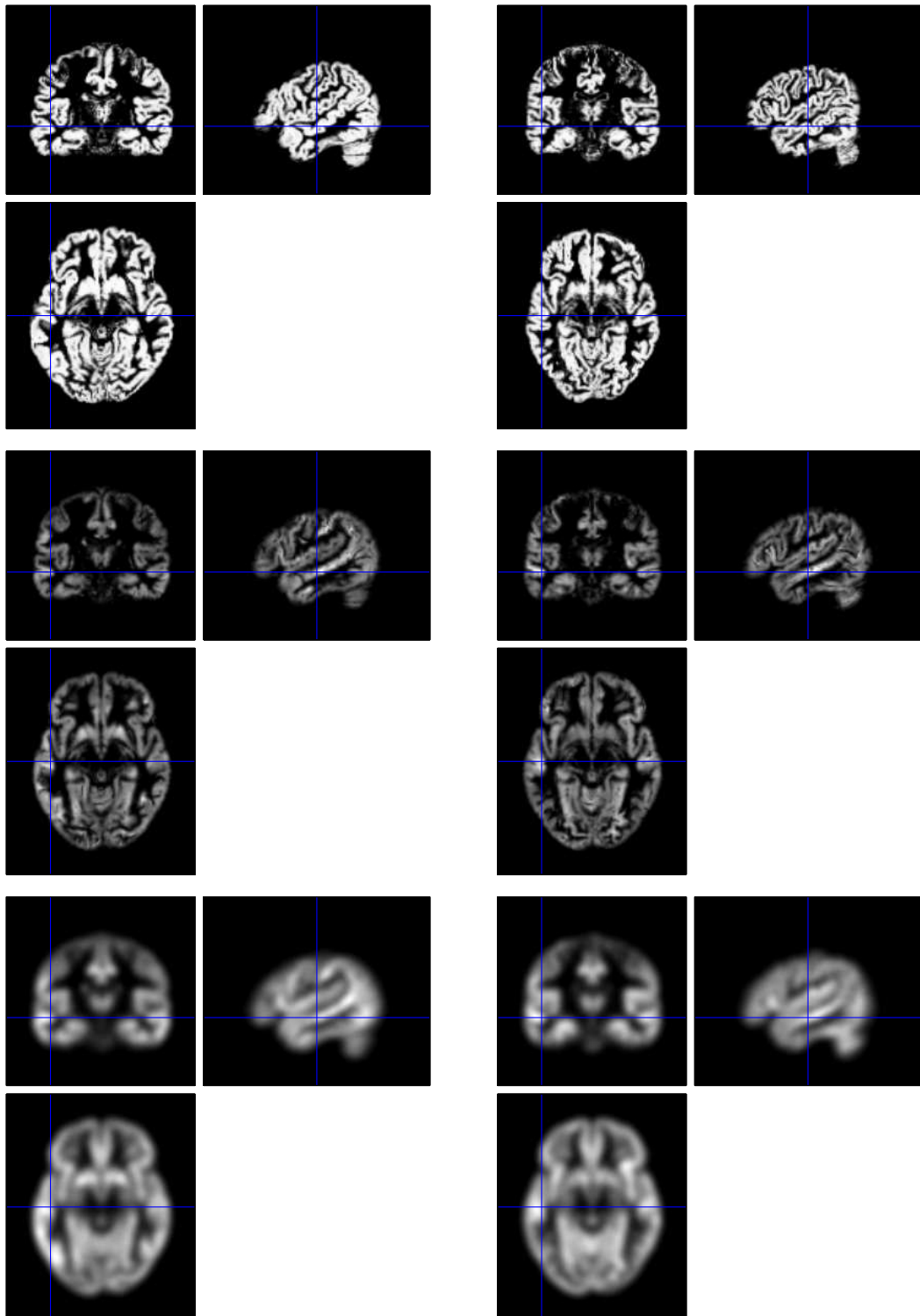


Figure 46.2: Pre-processing for VBM. Top row: Imported grey matter (rc1A.nii and rc1B.nii). Centre row: Warped with *Preserve Amount* option and zero smoothing (“modulated”). Bottom row: Warped with *Preserve Amount* option smoothing of 8mm (smwc1A.nii and smwc1B.nii).

The option for spatially normalising to MNI space automatically incorporates an affine transform that maps from the population average (Dartel Template space) to MNI space. This transform is estimated by minimising the KL divergence between the final template image generated by Dartel and tissue probability maps that are released as part of SPM (in the new segmentation toolbox). MNI space is defined according to affine matched images, so an affine transform of the Dartel template to MNI space would appear to be a reasonable strategy.

For GLM analyses, we usually do not wish to work with Jacobian scaled data. For this reason, warping is now combined with smoothing, in a way that may be a bit more sensible than simply warping, followed by smoothing. The end result is essentially the same as that obtained by doing the following with the old way of warping

- Create spatially normalised and “modulated” (Jacobian scaled) functional data, and smooth.
- Create spatially normalised maps of Jacobian determinants, and smooth by the same amount.
- Divide one by the other, adding a small constant term to the denominator to prevent divisions by zero.

This should mean that signal is averaged in such a way that as little as possible is lost. It also assumes that the procedure does not have any nasty side effects for the GRF assumptions used for FWE corrections.

Prior to spatially normalising using Dartel, the data should be processed as following:

- If possible, for each subject, use *SPM*→*Tools*→*FieldMap* to derive a distortion field that can be used for correcting the fMRI data. More accurate within-subject alignment between functional and anatomical scans should allow more of the benefits of Dartel for inter-subject registration to be achieved.
- Use either *SPM*→*Spatial*→*Realign*→*Realign: Estimate Reslice* or *SPM*→*Spatial*→*Realign Unwarp*. If a field map is available, then use the *Realign Unwarp* option. The images need to have been realigned and resliced (or field-map distortion corrected) beforehand - otherwise things are not handled so well. The first reason for this is that there are no options to use different methods of interpolation, so rigid-body transforms (as estimated by *Realign* but without having resliced the images) may not be well modelled. Similarly, the spatial transforms do not incorporate any masking to reduce artifacts at the edge of the field of view.
- For each subject, register the anatomical scan with the functional data (using *SPM* → *Spatial* → *Coreg* → *Coreg: Estimate*). No reslicing of the anatomical image is needed. Use *SPM*→*Util*→*Check Registration* to assess the accuracy of the alignment. If this step is unsuccessful, then some pre-processing of the anatomical scan may be needed in order to skull-strip and bias correct it. Skull stripping can be achieved by segmenting the anatomical scan, and masking a bias corrected version (which can be generated by the segmentation option) by the estimated GM, WM and CSF. This masking can be done using *SPM*→*Util*→*Image Calculator (ImCalc button)*, by selecting the bias corrected scan (*m\*.img*), and the tissue class images (*c1\*.img*, *c2\*.img* and *c3\*.img*) and evaluating “*i1. × ((i2+i3+i4)>0.5)*”. If segmentation is done before coregistration, then the functional data should be moved so that they align with the anatomical data.
- Segment the anatomical data and generate “imported” grey and white matter images.
- To actually estimate the warps, use *SPM*→*Tools*→*Dartel Tools*→*Run Dartel (create Templates)* in order to generate a series of templates and a flow field for each subject.

In principle (for a random effects model), you could run the first level analysis using the native space data of each subject. All you need are the contrast images, which can be warped and smoothed. Alternatively, you could warp and smooth the reslices fMRI, and do the statistical analysis on the spatially normalised images. Either way, you would select *SPM*→*Tools*→*Dartel Tools*→*Normalise to MNI Space*:



- **Normalise to MNI Space**

- **Dartel Template:** `Template_6.nii,1` is usually the grey matter component of the final template of the series. An affine transform is determined using this image.
- **Select according to** either *Few Subjects* or *Many Subjects*. For fMRI analyses, the *Few Subjects* option would be selected, which gives the option of selecting a flow field and list of images for each subject.
  - \* **Subject**
    - **Flow Field:** Specify the flow field ("`u_c1*.nii`") for this subject.
    - **Images:** Select the images for this subject that are to be transformed to MNI space.
- **Voxel sizes:** Specify the desired voxel sizes for the spatially normalised images (NaN, NaN, NaN gives the same voxel sizes as the Dartel template).
- **Bounding box:** Specify the desired bounding box for the spatially normalised images (NaN, NaN, NaN; NaN NaN NaN gives the same bounding box as the Dartel template).
- **Preserve:** Here you have a choice of *Preserve Concentrations* (ie not Jacobian scaled) or *Preserve Amount* (Jacobian scaled). The *Preserve Concentrations* option would normally be used for fMRI data, whereas *Preserve Amount* would be used for VBM.
- **Gaussian FWHM:** Enter how much to blur the spatially normalised images, where the values denote the full width at half maximum of a Gaussian convolution kernel, in units of mm.

An alternative approach is now presented, which does not attempt to make optimal use of the available signal.

### 46.2.1 An alternative approach for using Dartel to spatially normalise to MNI Space

During spatial normalisation of a brain image, some regions need to be expanded and other regions need to contract in order to match the template. If some structure is excessively shrunk by Dartel (because it has the freedom to estimate quite large deformations), then this will lead to a systematic reduction in the amount of BOLD signal being detected from that brain region. For this reason, the normalise to MNI space option would generally be preferred when working with functional data that is to be smoothed.

#### Affine transform of Dartel template to MNI space

Dartel works with images that are of average size. When Dartel is used to generate an average shaped template (represented by a series of tissue probability maps) from a group of scans of various individuals, the result is of average size. Brains normalised to MNI space are slightly larger than average. In order to spatially normalise to MNI space, the deformation that maps from MNI space to the space of the group average is required. Because the MNI space was derived by affine registration of a number of subjects to a common coordinate system, in most cases it should be possible to achieve a reasonable match of the template generated by Dartel using only an affine spatial normalisation. This can be achieved by matching the grey matter component of the template with a grey matter tissue probability map in MNI space. The spatial normalisation routine in SPM can be used to achieve this.

- **Normalise: Estimate**

- **Data**

- \* **Subject**

- **Source Image:** `Template_6.nii,1` is usually the grey matter component of the final template of the series.
- **Source Weighting Image:** `<None>`

### – Estimation Options

- \* **Template Image:** Should be the apriori/grey.nii file distributed in SPM.
- \* **Template Weighting Image:** <None>
- \* **Source Image Smoothing:** 8mm (the same as the apriori/grey.nii file has been smoothed).
- \* **Template Image Smoothing:** 0mm (because the data in the apriori folder are already smoothed by 8mm.)
- \* **Affine Regularisation:** Usually, you would specify “ICBM space template”.
- \* **Nonlinear Frequency Cutoff:** Set this to infinity (enter “Inf”) for affine registration.
- \* **Nonlinear Iterations:** Setting this to zero will also result in affine-only spatial normalisation.
- \* **Nonlinear Regularisation:** Setting this to infinity is another way of doing affine-only spatial normalisation.

For some populations of subjects, an affine transform may not be adequate for achieving good registration of the average shape to MNI space. Nonlinear spatial normalisation may be more appropriate for these cases. As ever, determining which procedure is better would involve a degree of empirical exploration.

### Combining deformations

Once you have the spatial transformation that maps from MNI space to the space of the Dartel template, it is possible to combine this with the DEFORMATIONS estimated by Dartel. Rather than warping the image data twice (introducing interpolation artifacts each time), the two spatial transforms can be combined by composing them together. The required deformation, for spatially normalising an individual to MNI space, is a mapping from MNI space to the individual image. This is because the spatially normalised images are generated by scanning through the (initially empty) voxels in the spatially normalised image, and figuring out which voxels in the original image to sample from (as opposed to scanning through the original image and putting the values into the right places in the spatially normalised version).

The desired mapping is from MNI space to Dartel template to individual scan. If  $A$  is the mapping from MNI to template, and  $B$  is the mapping from template to individual, then this mapping is  $B \circ A$ , where “ $\circ$ ” denotes the composition operation. Spatially normalising via the composed deformations can be achieved through the *Deformations* utility.

#### • Deformations

##### – Composition

- \* **Dartel flow**
  - **Flow field:** Specify the u\_rc1\*.nii flow field for that subject.
  - **Forward/Backwards:** This should be set to “Backward” to indicate a mapping from template to individual.
  - **Time Steps:** This is the number of time steps used by the final iterations of the Dartel registration (usually 64).
  - **Dartel template:** leave this field empty.
- \* **Imported \_sn.mat**
  - **Parameter File:** Select the spatial normalisation parameters that would spatially normalise the Template\_6.nii file.
  - **Voxel sizes:** These are set to “NaN” (not a number) by default, which would take the voxel sizes for the apriori/grey.nii file. Alternatively, you could specify your favourite voxel sizes for spatially normalised images.
  - **Bounding box:** Again, these are set to non-finite values by default, which results in the same bounding box as the apriori/grey.nii file. To specify your favourite bounding box, enter  $[x_{min}, y_{min}, z_{min}; x_{max}, y_{max}, z_{max}]$  (in units of mm, relative to the AC).

- **Output**

- \* **Pullback**

- **Apply to:** Specify the images for that subject that you would like spatially normalised.
    - **Output destination:** Specify where you want to write the images.
    - **Interpolation:** Specify the form of interpolation.
    - **Mask images:** Say whether you want to mask the images (see the Chapter on Realignment for more information here).
    - **Gaussian FWHM:** The images can be smoothed when they are written. If you do not want this, then enter 0 0 0.

The above procedure would be repeated for each subject in the study.

### 46.3 Warping Images to Existing Templates

If templates have already been created using Dartel, then it is possible to align other images with such templates. The images would first be imported in order to generate rc1\*.nii and rc2\*.nii files. The procedure is relatively straight-forward, and requires the *SPM→Tools→Dartel Tools→Run Dartel (existing Template)* option to be specified. Generally, the procedure would begin by registering with a smoother template, and end with a sharper one, with various intermediate templates between.

- **Run Dartel (existing Templates)**

- **Images**

- \* **Images:** Select the rc1\*.nii files.
    - \* **Images:** Select the corresponding rc2\*.nii files.

- **Settings:** Most settings would be kept at the default values, except for the specification of the templates. These are specified in within each of the *Settings→Outer Iterations→Outer Iteration→Template* fields. If the templates are Template\_\*.nii, then enter them in the order of Template\_1.nii, Template\_2.nii, ... Template\_6.nii.

Running this option is rather faster than *Run Dartel (create Template)*, as templates are not created. The output is in the form of a series of flow fields (u\_rc1\*.nii).

### 46.4 Warping one individual to match another

Sometimes the aim is to deform an image of one subject to match the shape of another. This can be achieved by running Dartel so that both images are matched with a common template, and composing the resulting spatial transformations. This can be achieved by aligning them both with a pre-existing template, but it is also possible to use the *Run Dartel (create Template)* option with the imported data of only two subjects. Once the flow fields (u\_rc1\*.nii files) have been estimated, then the resulting deformations can be composed using *SPM→Utils→Deformations*. If the objective is to warp A.nii to align with B.nii, then the procedure is set up by:

- **Deformations**

- **Composition**

- \* **Dartel flow**

- **Flow field:** Specify the u\_rc1A\_Template.nii flow field.
      - **Forward/Backwards:** Backward.
      - **Time Steps:** Usually 64.
      - **Dartel template:** leave this field empty.

- \* **Dartel flow**

- **Flow Field:** Specify the u\_rc1B\_Template.nii flow field.

- **Forward/Backwards:** Forward.
- **Time Steps:** Usually 64.
- **Dartel template:** leave this field empty.
- \* **Identity**
  - **Image to base Id on:** Specify B.nii in order to have the deformed image(s) written out at this resolution, and with the same orientations etc (ie so there is a voxel-for-voxel alignment, rather than having the images only aligned according to their “voxel-to-world” mappings).
- **Output**
  - \* **Pullback**
    - **Apply to:** Specify A.nii, and any other images for that subject that you would like warped to match B.nii. Note that these other images must be in alignment according to *Check Reg.*
    - **Output destination:** Specify where you want to write the images.
    - **Interpolation:** Specify the form of interpolation.
    - **Mask images:** Say whether you want to mask the images.
    - **Gaussian FWHM:** The images can be smoothed when they are written. If you do not want this, then enter 0 0 0.

Suppose the image of one subject has been manually labelled, then this option is useful for transferring the labels on to images of other subjects.

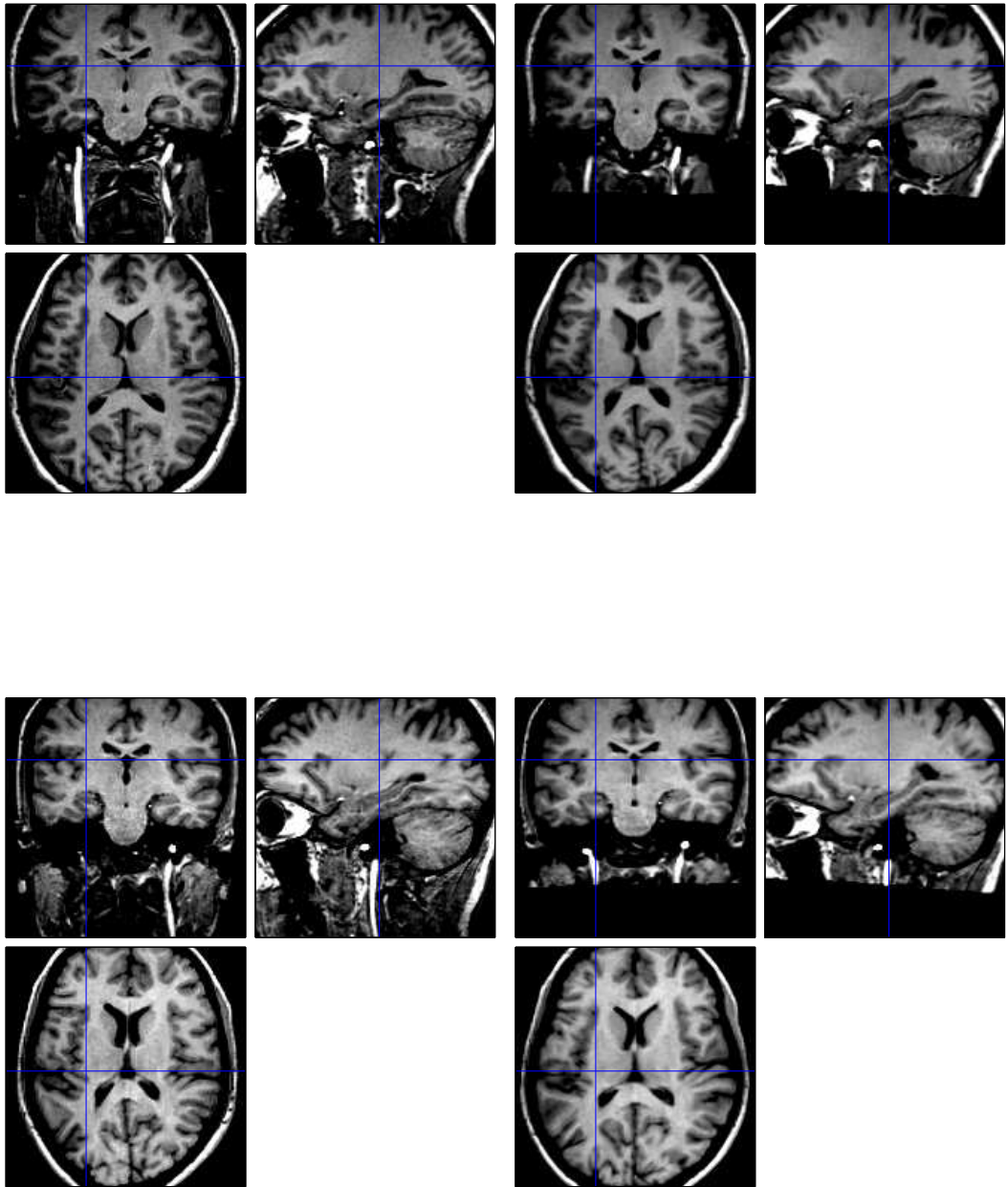


Figure 46.3: Composition of deformations to warp one individual to match another. Top-left: Original A.nii. Top-right: A.nii warped to match B.nii. Bottom-left: Original B.nii. Bottom-right: B.nii warped to match A.nii.



Part VIII

Batch Interface





# Chapter 47

## Batch tutorial

Details about the algorithms used for data processing are given in the other sections of this manual. This section explains how a sequence of processing steps can be run at once without MATLAB programming. SPM12 includes `matlabbatch`<sup>1</sup> which has been derived from the SPM5 batch system, but is also available as a separate package.

In `matlabbatch`, each data processing step is called “module”. There are e.g. modules for spatial processing of MRI data (realignment, normalisation, smoothing), statistics (fMRI or factorial design specification, model estimation, contrast specification). A batch describes which modules should be run on what kind of data and how these modules depend on each other.

Compared to running each processing step interactively, batches have a number of advantages:

**Documentation** Each batch can be saved as a MATLAB script. All parameters (including default settings) are included in this script. Thus, a saved batch contains a full description of the sequence of processing steps and the parameter settings used.

**Reproducibility** Batches can be saved, even if not all parameters have been set. For a multi-subject study, this allows to create template batches. These templates contain all settings which do not vary across subjects. For each subject, they can be loaded and only subject-specific parts need to be completed.

**Unattended execution** Instead of waiting for a processing step to complete before entering the results in the next one, all processing steps can be run in the specified order without any user interaction.

**Multiple batches** Multiple batches can be loaded and executed together.

**Error reporting** If a batch fails to complete, a standardised report will be given in the MATLAB command window. When running a batch from the GUI, this can be saved to an error `.mat` file.

### 47.1 Single subject

In this tutorial we will develop a batch for spatial processing and fMRI statistics of a single subject of the “Face” example dataset (see chapter 31). To follow this tutorial, it is not necessary to download the example dataset, except for the last step (entering subject dependent data).

To create a batch which can be re-used for multiple subjects in this study, it is necessary to collect/define

- study specific input data (e.g. MRI measurement parameters, time constants of the functional experiment, number of sessions),
- necessary processing steps,
- data flow between processing steps.

---

<sup>1</sup><http://sourceforge.net/projects/matlabbatch>

Subject specific input data (original functional and structural MRI data, subject specific experiment parameters) should be entered after the batch template has been saved.

### 47.1.1 Study specific input data

This dataset consists of fMRI data acquired in a single session and a structural MRI. See section 47.2 to learn how to deal efficiently with multi-session data. MRI parameters and experiment details are described in chapter 31.

### 47.1.2 Necessary processing steps

#### Helper modules

Some SPM modules produce graphics output which is captured in a PostScript file in the current working directory. Also, a new directory needs to be created for statistics. The “BasicIO” menu provides a collection of modules which are useful to organise a batch. We will need the following modules:

- Named directory selector
- Change directory
- Make directory

#### SPM processing

For a classical SPM analysis, the following processing steps are necessary:

- Realignment
- Slice timing correction
- Coregistration
- Segmentation
- Normalisation
- Smoothing
- fMRI design
- Model estimation
- Contrasts
- Results report

### 47.1.3 Add modules to the batch

The helper modules and the SPM processing modules can be assembled using the GUI. Click the “BATCH” button in the SPM Menu window. First, add the helper modules, followed by the SPM modules in the order listed above. Do not configure any details until you have selected all modules.

### 47.1.4 Configure subject-independent data

Now, go through the batch and configure all settings that are subject-independent (e.g. the name of the analysis directory, slice timing parameters) as described in chapter 31. Do not enter any data that is specific for a certain subject. The values that need to be entered are not repeated here, please refer to the corresponding sections in chapter 31.

The file `man/batch/face_single_subject_template_nodeps.m` contains the batch after all modules have been added and subject-independent data has been entered.

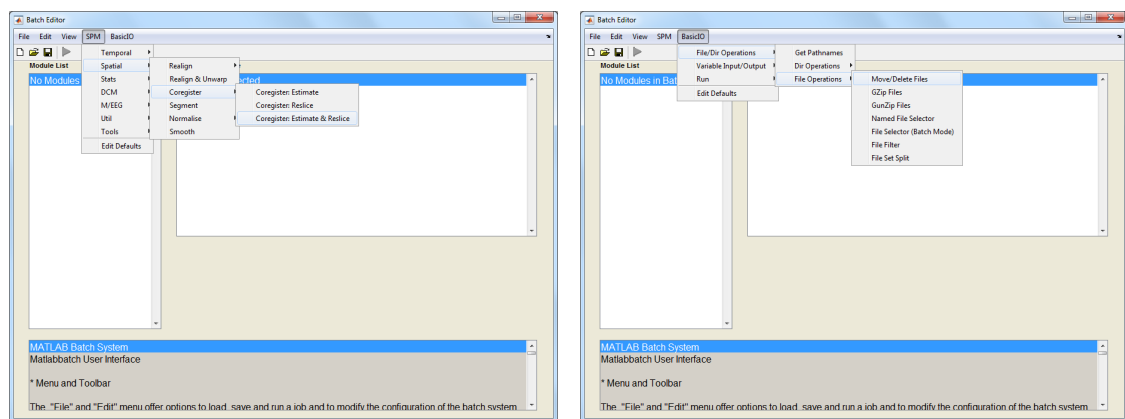


Figure 47.1: SPM and BasicIOapplication menus

### Named Directory Selector

**Input Name** Give this selection a name (e.g. “Subject directory”) - this name will be shown in the dependency list of this batch.

**Directories** Add a new directory selector, but do not enter a directory itself.

### Change Directory

Nothing to enter now.

### Make Directory

**New Directory Name** “categorical” - the name of the analysis directory. This directory will be created at batch run-time in the subject directory.

### Realign: Estimate & Reslice

**Data** Add a new “Session” item. Do not enter any files for this session now.

### Slice Timing

**Data** Add a new “Session” item. Do not enter any files for this session now.

**Timing options** Enter data for “Number of slices”, “TR”, “TA”, “Slice order”, “Reference slice”.

### Coreg: Estimate

Nothing to enter now.

### Segment

Nothing to enter now.

### Normalise: Write

**Data** Add a new “Subject”. Do not enter any files now.

**Writing Options** Adjust bounding box, voxel sizes, interpolation

### Smooth

**FWHM** Enter FWHM

**fMRI model specification**

Enter all data which is constant across subjects.

**Timing parameters** Enter values for “Units for design”, “Interscan interval”, “Microtime resolution”, “Microtime onset”

**Data & Design** Add a new “Session” item. Do not enter scans, conditions or regressors yet. They will be added as dependencies or subject specific inputs. If you want to make sure to remember this, you can highlight “Multiple conditions” and select “Clear Value” from the “Edit” menu. Do the same for “Multiple regressors”. This will mark both items with an <-X, indicating that something must be entered there.

**Factorial design** Enter the specification for both factors.

**Basis functions** Select the basis function and options you want to use.

**Model estimation**

Nothing to be entered yet for classical estimation.

**Contrast manager**

If you have selected the “Factorial design” option as described above, SPM will automatically create some contrasts for you. Here, you can create additional T- or F-contrasts. As an example, we will add an “Effects of interest” F-contrast.

**Contrast session** Add a new “F-contrast” item.

**Name** Enter a name for this contrast, e.g. “Effects of interest”.

**Contrast vectors** Add a new “Contrast vector” item. F-contrasts can have multiple rows. You can either enter a contrast matrix in an “F contrast vector” entry, or enter them row by row. To test for the effects of interest (1 basis function and 2 derivatives for each of the four conditions) enter `eye(12)` as F contrast vector.

**Replicate over sessions** This design does not have multiple sessions, so it is safe to say “No” here.

**Results report**

Reviewing individual results for a large number of subjects can be very time consuming. Results report will print results from selected contrasts to a PostScript file.

**Contrast(s)** Enter `Inf` to print a report for each of the defined contrasts.

**47.1.5 Data flow**

In chapter 31, each processing step was performed on its own. In most cases, output data was simply passed on from one module to the next. This scheme is illustrated in figure 47.2. Only the coloured items at the top of the flow chart are subject specific and need to be entered in the final batch. All arrow connections are subject-independent and can be specified in the batch template.

**Add dependencies**

Based on the data flow in figure 47.2, modules in the batch can now be connected. The batch containing all dependencies can be found in `man/batch/face_single_subject_template.m`.

Again, start editing at the top of the batch:

**Named Directory Selector**

Nothing to enter now.

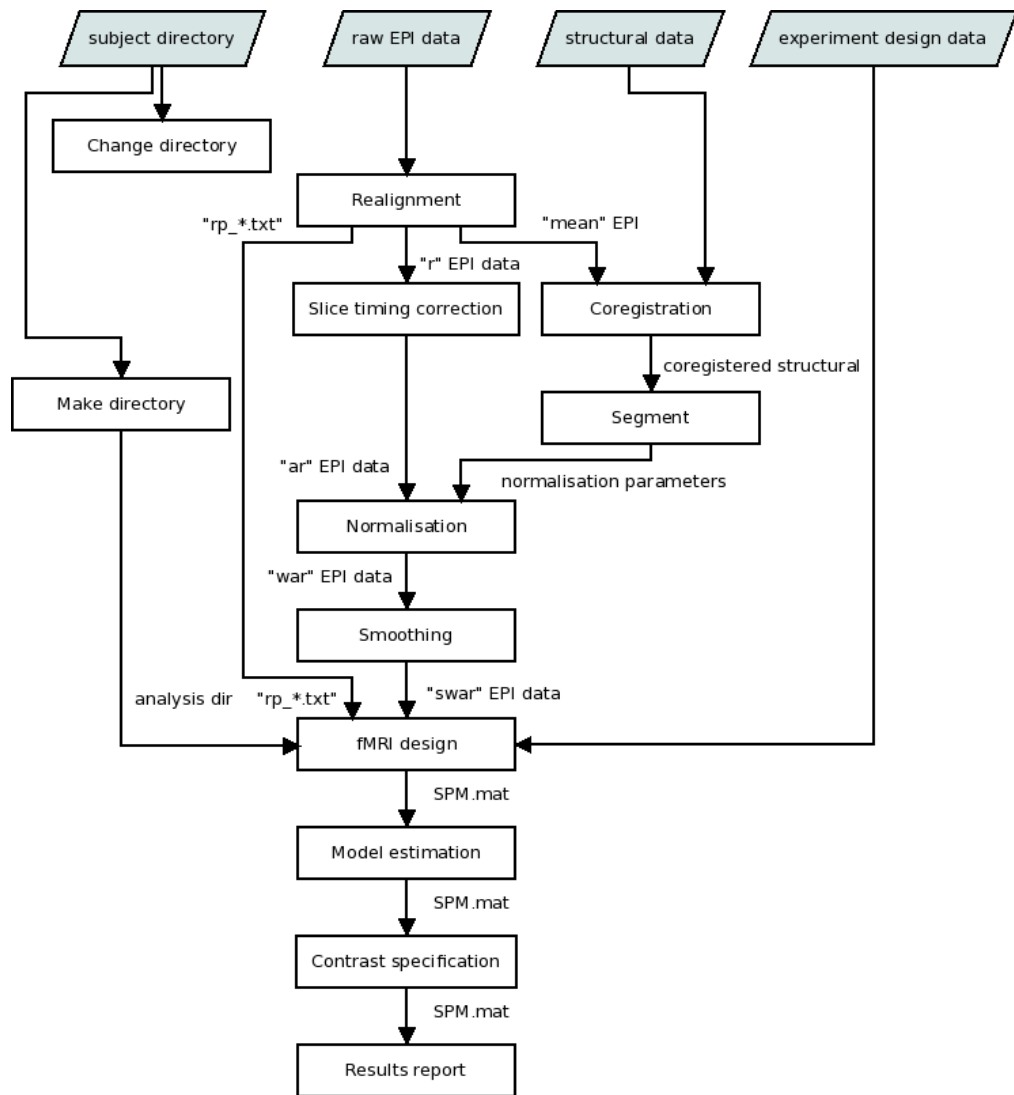


Figure 47.2: Flow chart for batch

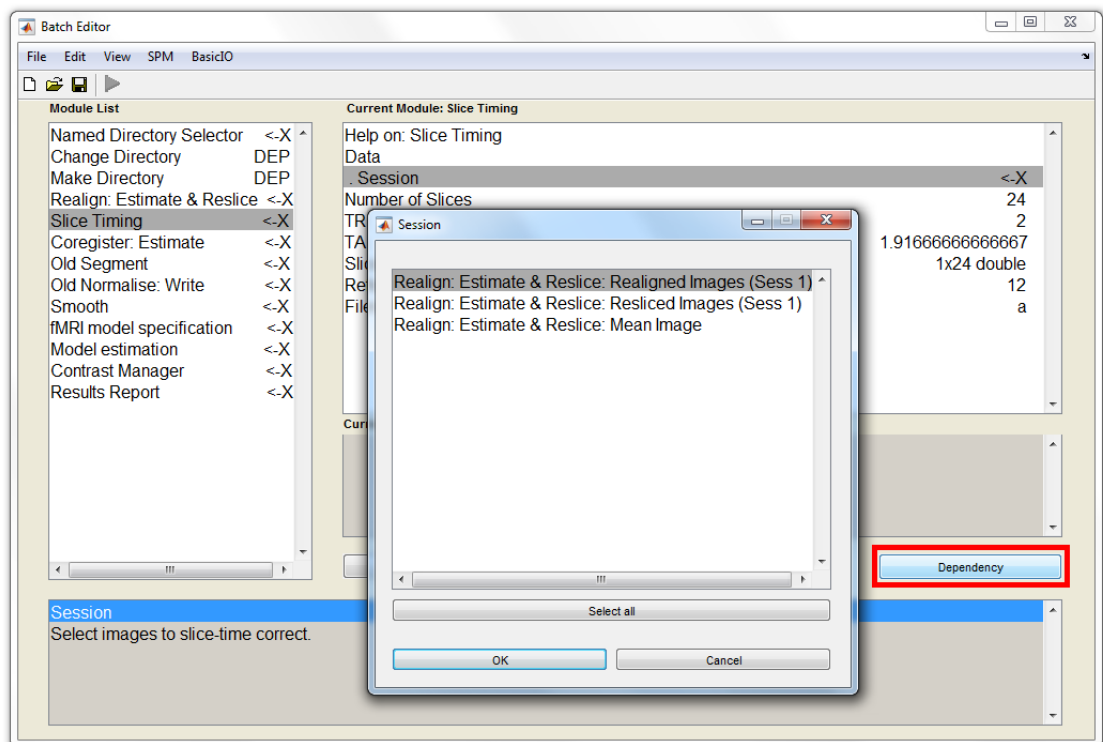


Figure 47.3: Dependency selection

### Change Directory

**Directory** Press “Dependency” and select “Subject directory(1)”. At run time, SPM will change to this directory before batch processing continues.

### Make Directory

**Parent Directory** Press “Dependency” and select “Subject directory(1)”. The “categorical” directory will be created in this directory.

### Realign: Estimate & Reslice

Nothing to enter now.

### Slice Timing

**Session** Press “Dependency” and select “Resliced Images (Sess 1)”.

### Coreg: Estimate

**Reference Image** Press “Dependency” and select “Mean Image”.

### Segment

**Data** Press “Dependency” and select “Coregistered Images”. At run time, this will resolve to the coregistered anatomical image.

### Normalise: Write

**Parameter File** Press “Dependency” and select “Norm Params File Subj→MNI (Subj 1)”.

**Images to Write** Press “Dependency” and select “Slice Timing Corr. Images (Sess 1)”.

**Smooth**

**Images to Smooth** Press “Dependency” and select “Normalised Images (Subj 1)”

**fMRI model specification**

**Directory** Press “Dependency” and select “Make Directory ‘categorical’”

**Scans** Press “Dependency” and select “Smoothed Images”. Note: this works because there is only one session in our experiment. In a multisession experiments, images from each session may be normalised and smoothed using the same parameters, but the smoothed images need to be split into sessions again. See section 47.2 how this can be done.

**Multiple regressors** Press “Dependency” and select “Realignment Param File (Sess 1)”.

**Model estimation**

**Select SPM.mat** Press “Dependency” and select “SPM.mat File (fMRI Design&Data)”.

**Contrast manager**

**Select SPM.mat** Press “Dependency” and select “SPM.mat File (Estimation)”.

**Results report**

**Select SPM.mat** Press “Dependency” and select “SPM.mat File (Contrasts)”.

**47.1.6 Entering subject-specific data**

Now, only 4 modules should have open inputs left (marked with <-X). These inputs correspond to data which vary over the subjects in your study:

**Named Directory Selector** Subject directory

**Realign: Estimate & Reslice** Raw EPI data for the fMRT session

**Coreg: Estimate** Anatomical image to be coregistered to mean EPI

**fMRI model specification** Names, conditions and onsets of your experimental conditions, specified in a multiple conditions .mat file.

Using the GUI, you can now perform these steps for each subject:

1. load the template batch
2. enter subject-specific data
3. save batch under a subject specific name.

After that, all batches for all subjects can be loaded and run at once.

This process can be automated using some basic MATLAB scripting. See section 47.2.3 for details.

**47.2 Advanced features****47.2.1 Multiple sessions**

If an fMRI experiment has multiple sessions, some processing steps need to take this into account (slice timing correction, realignment, fMRI design), while others can work on all sessions at once (normalisation, smoothing).

Two modules in BasicIO help to solve this problem:

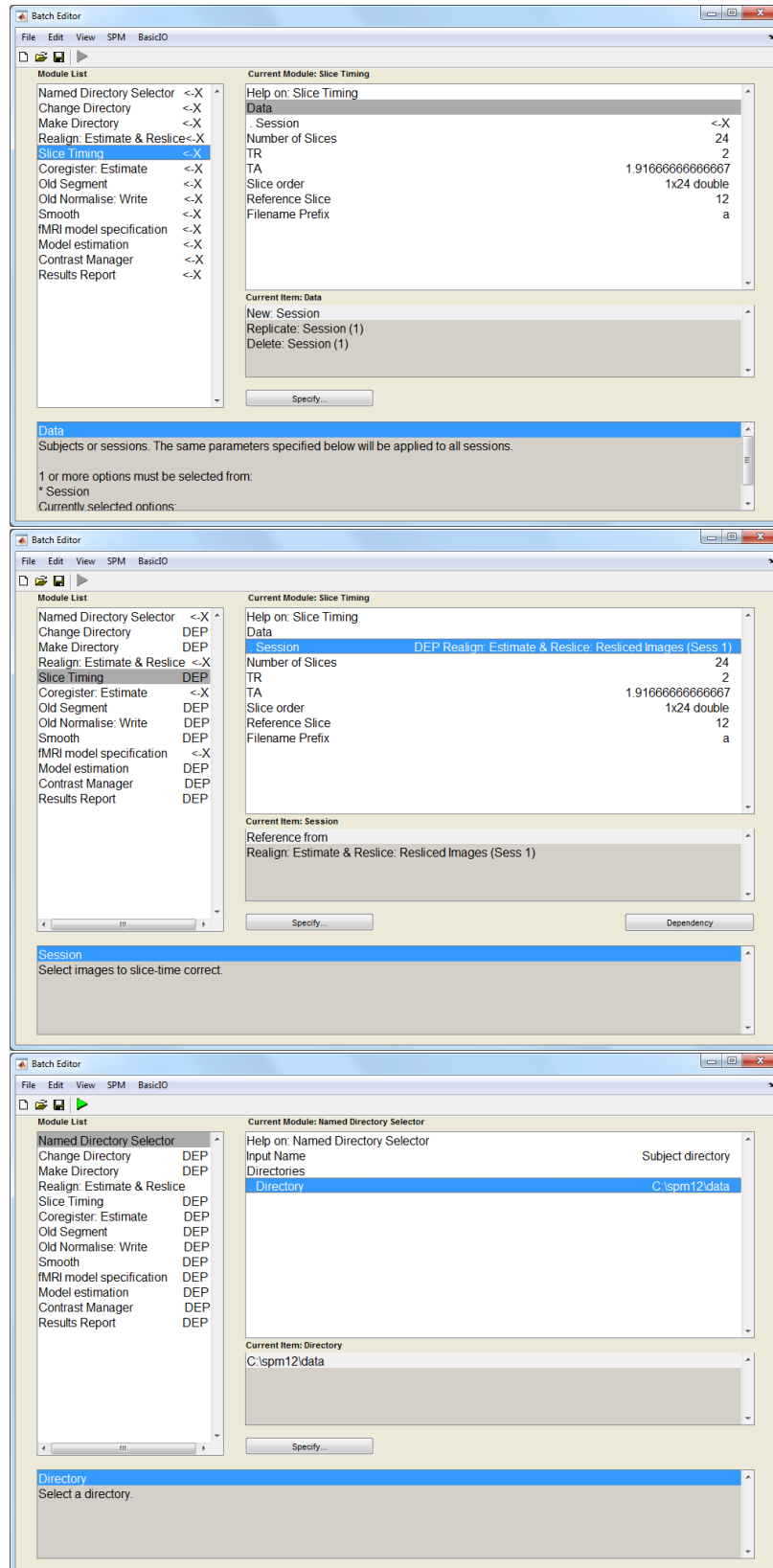


Figure 47.4: All stages of batch entry



**Named File Selector** Files can be entered here session by session. Note that this file selector selects all files (not restricted to images) by default. To select only images, set the filter string to something like `.*nii$` or `.*img$`.

**File Set Split** This module splits a list of files based on an index vector. Named file selector provides such an index vector to split the concatenation of all selected images into individual sessions again.

### 47.2.2 Processing multiple subjects in GUI

There are different ways to process multiple subjects in the batch editor:

- Add the necessary processing steps when creating the job.
- Create a per-subject template, save it and load it multiple times (i.e. in the file selector, add the same file multiple times to the list of selected files).
- Use “Run Batch Jobs” from “BasicIO”

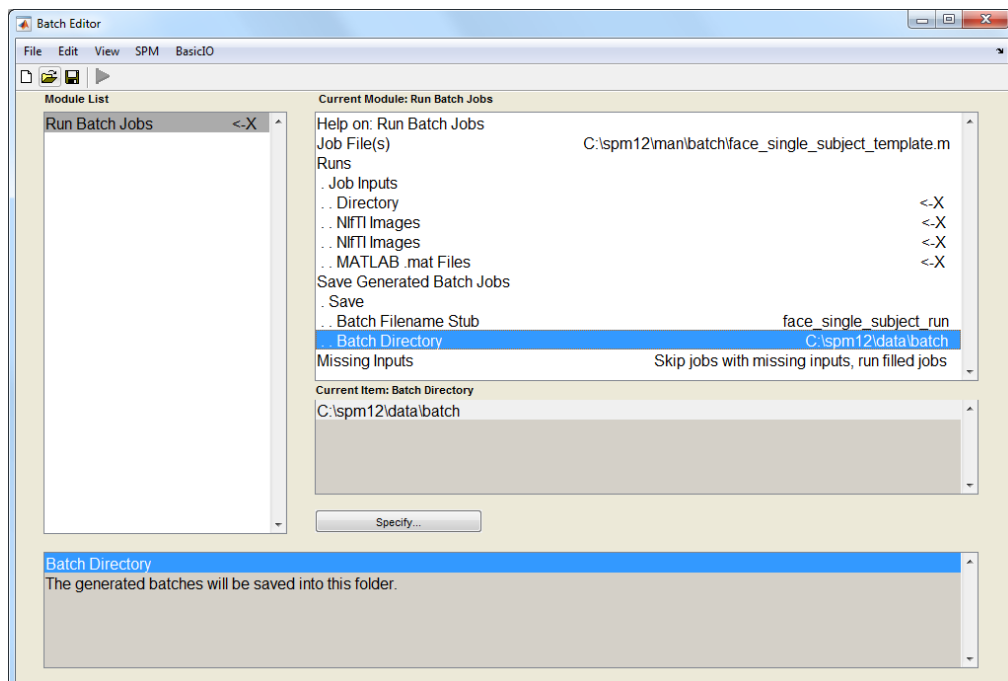


Figure 47.5: Using “Run Batch Jobs”

In all cases, the data for all subjects has to be entered through the GUI, and computation will be done for all subjects at once after all data is entered. There is an example job `face_multi_subject_template.m` that demonstrates the usage of “Run Batch Jobs” to run the single subject template job described above. Note that the order and type of inputs in the single subject template is important. Also, consistency checks are limited. If inconsistent data is entered, the job will fail to execute and return an error message.

To run this job for multiple subjects, simply repeat the “Runs” item as many times as necessary and fill in the required data.

### 47.2.3 Command line interface

The command line interface is especially useful to run multiple jobs at once without user interaction, e.g. to process multiple subjects or to combine separate processing steps. There is a “high-level” interface using `spm_jobman`, which combines “low-level” callbacks to `cfg_util`.

### SPM startup in command line mode

During normal startup, SPM performs important initialisation steps. Without initialisation, SPM and its batch system will not function properly. Consequently, an initialisation sequence needs to be run before any batch job can be submitted.

MATLAB has several command line options to start without its GUI (`-nodesktop`) or even without any graphics output to a screen (`-nodisplay`). See MATLAB documentation for details.

To run SPM in `-nodisplay` mode, the file `spm_defaults.m` has to be modified. The line `defaults.cmdline = 0;` must be changed to `defaults.cmdline = true;`. In command line mode, SPM will not open any figure window except the “Graphics” window.

Within MATLAB, the following commands are sufficient to set up SPM

1. `spm('defaults', MODALITY)` where MODALITY has to be replaced by the desired modality (e.g. 'fmri')
2. `spm_jobman('initcfg')`

After executing these commands, any SPM functions and batch jobs can be run in the same MATLAB session.

### Complete and run a pre-specified job

`spm_jobman('run', job[, input1, input2 ...])`

This interface takes a job and asks for the input to any open configuration items one after another. If a list of appropriate inputs is supplied, these will be filled in. After all inputs are filled, the job will be run. Note that only items without a pre-set value will be filled (marked with `<-X` in the GUI). To force a item to to be filled, use “Edit:Clear Value” in the GUI or set its value to '`<UNDEFINED>`' in the harvested job.

The job argument is very flexible, it can e.g. be a job variable, the name of a script creating a job variable, even a cell list of any mixture of variables and scripts. All job snippets found will be concatenated into a single job, the missing inputs will be filled and the resulting job will be run.

The batch system can generate a script skeleton for any loaded job. From the batch GUI, this feature is accessible via “File:Save Batch and Script”. This skeleton consists of a commented list of necessary inputs, a `for` loop to enter inputs for multiple runs or subjects and the code to initialise and run the job. An example is available in `face_single_subject_script.m`:

```
% List of open inputs
% Named Directory Selector: Directory - cfg_files
% Realign: Estimate & Reslice: Session - cfg_files
% Coreg: Estimate: Source Image - cfg_files
% fMRI model specification: Multiple conditions - cfg_files
nrun = X; % enter the number of runs here
jobfile = {fullfile(spm('dir'),'man','batch','face_single_subject_template.m')};
jobs = repmat(jobfile, 1, nrun);
inputs = cell(4, nrun);
for crun = 1:nrun
    % Named Directory Selector: Directory - cfg_files
    inputs{1, crun} = MATLAB_CODE_TO_FILL_INPUT;
    % Realign: Estimate & Reslice: Session - cfg_files
    inputs{2, crun} = MATLAB_CODE_TO_FILL_INPUT;
    % Coreg: Estimate: Source Image - cfg_files
    inputs{3, crun} = MATLAB_CODE_TO_FILL_INPUT;
    % fMRI model specification: Multiple conditions - cfg_files
    inputs{4, crun} = MATLAB_CODE_TO_FILL_INPUT;
end
spm('defaults','fmri');
spm_jobman('run',jobs,inputs{:});
```

The skeleton needs to be adapted to the actual data layout by adding MATLAB code which specifies the number of runs and the input data in the `for` loop.

Another example script and batch is available for the multimodal dataset, called `multimodal_fmri_script.m` and `multimodal_fmri_template.m`.

#### 47.2.4 Modifying a saved job

In some cases, instead of using the serial interface it may be more appropriate to modify the fields of a saved or harvested job. By default, jobs are saved as MATLAB `.mat` files, but they can also be saved as `.m` files. These files contain a number of MATLAB commands, which will create a variable `matlabbatch`. The commands can be modified to set different values, add or remove options.



# Chapter 48

## Developer's guide

### 48.1 SPM and Matlabbatch code organisation

This is a short overview describing code organisation and interfaces between SPM and the batch system.

#### 48.1.1 Code organisation

Most features are implemented in:

- `fullfile(spm('dir'),'matlabbatch')`: core batch system.
- `fullfile(spm('dir'),'config')`: SPM config files.
- `spm_jobman.m` and `spm_select.m`: wrappers to Matlabbatch.

Some assignments to configuration items are guarded by validity checks. Usually, there will be a warning issued if a wrong value is supplied. Special care needs to be taken for `.prog`, `.vfiles`, `.vout`, `.check` functions or function handles. The functions referenced here must be on MATLAB path before they are assigned to one of these fields. For toolboxes, this implies that toolbox paths must be added at the top of the configuration file.

#### 48.1.2 Interfaces between SPM and Matlabbatch

**Configuration files:** Configuration items are defined as objects. Structs of type `<type>` in SPM5 are represented as objects of class `cfg_<type>`. There is a class `cfg_exbranch` which is used for branches that have a `.prog` field.

**Dependencies:** Dependencies require computations to return a single output argument (e.g. a cell, struct). Parts of this output argument can be passed on to new inputs at run-time.

**Interface to the batch system:**

- `cfg_util` Configuration management, job management, job execution,
- `cfg_serial` A utility to fill missing inputs and run a job (optionally with a GUI input function),
- `cfg_ui` Graphical User Interface.

### 48.2 Configuration Code Details

Configuration code is split into two files per configuration:

`spm_cfg_*.m` Configuration classes, `.check`, `.vout` subfunctions

`spm_run_*.m` Run-time code, takes job structure as input and returns output structure as specified in `.vout`.

In a few cases (where there was no additional magic in the code), run-time code has been integrated into the main SPM code. This may be useful to run test batches without using the configuration/batch system.

### 48.2.1 Virtual Outputs

Virtual outputs are described by arrays of `cfg_dep` objects. These objects contain a “source” and a “target” part. Functions may have more than one virtual output (e.g. one output per session, a collection of results variables). One `cfg_dep` object has to be created for each output.

Only two fields in the “source” part need to be set in a `.vout` callback:

**sname** A display name for this output. This will appear in the dependencies list and should describe the contents of this dependency.

**src\_output** A subscript reference that can be used to address this output in the variable returned at run-time.

**tgt\_spec (optional)** A description on what kind of inputs this output should be displayed as dependency. This is not very convenient yet, the `match` and `cfg_findspec` methods are very restrictive in the kind of expressions that are allowed.

The `.vout` callback will be evaluated once the configuration system thinks that enough information about the *structure* of the outputs is available. This condition is met, once all in-tree nodes `cfg_(ex)branch`, `cfg_choice`, `cfg_repeat` have the required number of child nodes.

The `.vout` callback is called with a job structure as input, but its code *should not rely* on the evaluation of any contents of this structure (or at least provide a fallback). The contents of the leaf nodes may not be set or may contain a dependency object instead of a value during evaluation of `.vout`.

The “target” part will be filled by the configuration classes, the `src_exbranch` field is set in `cfg_util`.

### 48.2.2 SPM Startup

The top level configuration file for SPM is `spm_cfg.m`. It collects SPM core configuration files and does toolbox autodetection. If a toolbox directory contains `*_cfg_*.m` files, they will be loaded.

### 48.2.3 Defaults Settings

In `Matlabbatch`, there are different ways to set defaults:

1. in the configuration file itself,
2. in a defaults file, which has a structure similar to a harvested job,
3. using a `.def` field for leaf items.

Defaults set using option **1** or **2** will only be updated at SPM/`matlabbatch` startup. Defaults set using option **3** will be set once a new job is started. These defaults take precedence over the other defaults.

In core SPM, these defaults refer to `spm_get_defaults`, which accesses `spm_defaults`. Toolboxes may use their own callback functions.

Toolboxes should set their defaults using the `.def` fields, using a mechanism similar to `spm_get_defaults`. This allows for flexibility without interfering with SPMs own defaults.

## 48.3 Utilities

### 48.3.1 Batch Utilities

Matlabbatch is designed to support multiple applications. A standard application “BasicIO” is enabled by default. Among other options, it contains file/file selection manipulation utilities which can be used as a dependency source if multiple functions require the same set of files as an input argument. For debugging purposes, “Pass Output to Workspace” can be used to assign outputs of a computation to a workspace variable.

The `cfg_configui` folder contains an application which describes all configuration items in terms of configuration items. It is not enabled by default, but can be added to the batch system using `cfg_util('addapp'...)`. This utility can be used to generate a batch configuration file with the batch system itself.

### 48.3.2 MATLAB Code Generation

The `gencode` utility generates MATLAB `.m` file code for any kind of MATLAB variable. This is used to save batch files as well as to generate configuration code.

### 48.3.3 Configuration Management

The backend utility to manage the configuration data is `cfg_util`. It provides callbacks to add application configurations, and to load, modify, save or run jobs. These callbacks are used by two frontends: `cfg_ui` is a MATLAB GUI, while `cfg_serial` can be used both as a GUI and in script mode. In script mode, it will fill in job inputs from an argument list. This allows to run predefined jobs with e.g. subject dependent inputs without knowing the exact details of the job structure.





Part IX  
Bibliography



# Bibliography

- [1] J. Andersson, J. Ashburner, and K.J. Friston. A global estimator unbiased by local changes. *NeuroImage*, 13(6):1193–1206, 2001.
- [2] J. Andersson, C. Hutton, J. Ashburner, R. Turner, and K.J. Friston. Modelling geometric deformations in EPI time series. *NeuroImage*, 13(5):903–919, 2001.
- [3] J. Ashburner. A fast diffeomorphic image registration algorithm. *NeuroImage*, 38(1):95–113, 2007.
- [4] J. Ashburner and K.J. Friston. The role of registration and spatial normalization in detecting activations in functional imaging. *Clinical MRI/Developments in MR*, 7(1):26–28, 1997.
- [5] J. Ashburner and K.J. Friston. Nonlinear spatial normalization using basis functions. *Human Brain Mapping*, 7(4):254–266, 1999.
- [6] J. Ashburner and K.J. Friston. Voxel-based morphometry – the methods. *NeuroImage*, 11:805–821, 2000.
- [7] J. Ashburner and K.J. Friston. Why voxel-based morphometry should be used. *NeuroImage*, 14(6):1238–1243, 2001.
- [8] J. Ashburner and K.J. Friston. Unified segmentation. *NeuroImage*, 26:839–851, 2005.
- [9] J. Ashburner and S. Klöppel. Multivariate models of inter-subject anatomical variability. *NeuroImage*, 56(2):422–439, 2011.
- [10] J. Ashburner, P. Neelin, D. L. Collins, A. C. Evans, and K.J. Friston. Incorporating prior knowledge into image registration. *NeuroImage*, 6:344–352, 1997.
- [11] S. Baillet, J.C. Mosher, and R.M. Leahy. Electromagnetic brain mapping. *IEEE Sign. Proc. Mag.*, 18:14–30, 2001.
- [12] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pat. Anal. and Mach. Intel.*, 142:239–256, 1992.
- [13] C. Buchel and K.J. Friston. Modulation of connectivity in visual pathways by attention: Cortical interactions evaluated with structural equation modelling and fMRI. *Cerebral Cortex*, 7:768–778, 1997.
- [14] C. Buchel, A.P. Holmes, G. Rees, and K.J. Friston. Characterizing stimulus-response functions using nonlinear regressors in parametric fMRI experiments. *NeuroImage*, 8:140–148, 1998.
- [15] C. Buchel, O. Josephs, G. Rees, R. Turner, and C. FrithK.J. Friston. The functional anatomy of attention to visual motion. a functional mri study. *Brain*, 121:1281–1294, 1998.
- [16] C. Buchel, R.J.S. Wise, C.J. Mummary, J.B. Poline, and K.J. Friston. Nonlinear regression in parametric activation studies. *NeuroImage*, 4:60–66, 1996.
- [17] C.C. Chen, R.N.A. Henson, K.E. Stephan, J. Kilner, and K.J. Friston. Forward and backward connections in the brain: A dcm study of functional asymmetries. *NeuroImage*, 45(2):453–462, 2009.

- [18] C.C. Chen, S.J. Kiebel, and K.J. Friston. Dynamic causal modelling of induced responses. *NeuroImage*, 41(4):1293–1312, 2008.
- [19] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multi-modality image registration based on information theory. In Y. Bizais, C. Barillot, and R. Di Paola, editors, *Proc. Information Processing in Medical Imaging*, pages 263–274, Dordrecht, The Netherlands, 1995. Kluwer Academic Publishers.
- [20] R. W. Cox and A. Jesmanowicz. Real-time 3D image registration for functional MRI. *Magnetic Resonance In Medicine*, 42:1014–1018, 1999.
- [21] A.M. Dale and M. Sereno. Improved localization of cortical activity by combining EEG and MEG with MRI surface reconstruction: a linear approach. *J. Cognit. Neurosci.*, 5:162–176, 1993.
- [22] J. Daunizeau, S.J. Kiebel, and K.J. Friston. Dynamic causal modelling of distributed electromagnetic responses. *NeuroImage*, 2009.
- [23] O. David, S.J. Kiebel, L. Harrison, J. Mattout, J. Kilner, and K.J. Friston. Dynamic causal modelling of evoked responses in EEG and MEG. *NeuroImage*, 30:1255–1272, 2006.
- [24] W. F. Eddy, M. Fitzgerald, and D. C. Noll. Improved image registration by using Fourier interpolation. *Magnetic Resonance in Medicine*, 36:923–931, 1996.
- [25] Matthias Fastenrath, K.J. Friston, and S.J. Kiebel. Dynamical causal modelling for m/eeg: Spatial and temporal symmetry constraints. *NeuroImage*, 2008.
- [26] Vladimir Fonov, Alan C Evans, Kelly Botteron, C Robert Almli, Robert C McKinsty, and D Louis Collins. Unbiased average age-appropriate atlases for pediatric studies. *NeuroImage*, 54(1):313–327, 2011.
- [27] K.J. Friston. Functional and effective connectivity in neuroimaging: A synthesis. *Human Brain Mapping*, 2:56–78, 1995.
- [28] K.J. Friston, J. Ashburner, S.J. Kiebel, T.E. Nichols, and W.D. Penny, editors. *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. Academic Press, 2007.
- [29] K.J. Friston, A. Bastos, V. Litvak, K.E. Stephan, P. Fries, and R. Moran. Dcm for complex-valued data: Cross-spectra, coherence and phase-delays. *NeuroImage*, 59(1):439–455, 2012.
- [30] K.J. Friston, C. Buchel, G.R. Fink, J. Morris, E. Rolls, and R. Dolan. Psychophysiological and modulatory interactions in neuroimaging. *NeuroImage*, 6:218–229, 1997.
- [31] K.J. Friston, C. Frith, R.S.J. Frackowiak, and R. Turner. Characterizing dynamic brain responses with fMRI: A multivariate approach. *NeuroImage*, 2:166–172, 1995.
- [32] K.J. Friston, D.E. Glaser, R.N.A. Henson, S.J. Kiebel, C. Phillips, and J. Ashburner. Classical and Bayesian inference in neuroimaging: Applications. *NeuroImage*, 16:484–512, 2002.
- [33] K.J. Friston, L. Harrison, J. Daunizeau, S.J. Kiebel, C. Phillips, N. Trujillo-Bareto, R.N.A. Henson, G. Flandin, and J. Mattout. Multiple sparse priors for the m/eeg inverse problem. *NeuroImage*, 39(3):1104–1120, 2008.
- [34] K.J. Friston, L. Harrison, and W.D. Penny. Dynamic Causal Modelling. *NeuroImage*, 19(4):1273–1302, 2003.
- [35] K.J. Friston, R.N.A. Henson, C. Phillips, and J. Mattout. Bayesian estimation of evoked and induced responses. *Human Brain Mapping*, 27:722–735, 2005.
- [36] K.J. Friston, J. Kahan, B. Biswal, and A. Razi. A DCM for resting state fMRI. *NeuroImage*, 94:396–407, 2014.
- [37] K.J. Friston, J. Mattout, N. Trujillo-Bareto, J. Ashburner, and W.D. Penny. Variational free energy and the laplace approximation. *NeuroImage*, 34(1):220–234, 2007.

- [38] K.J. Friston, A. Mechelli, R. Turner, and C.J. Price. Nonlinear responses in fMRI: The Balloon model, Volterra kernels and other hemodynamics. *NeuroImage*, 12:466–477, 2000.
- [39] K.J. Friston and W.D. Penny. Posterior probability maps and SPMs. *NeuroImage*, 19(3):1240–1249, 2003.
- [40] K.J. Friston, W.D. Penny, and D.E. Glaser. Conjunction revisited. *NeuroImage*, 25(3):661–667, 2005.
- [41] K.J. Friston, W.D. Penny, C. Phillips, S.J. Kiebel, G. Hinton, and J. Ashburner. Classical and Bayesian inference in neuroimaging: Theory. *NeuroImage*, 16:465–483, 2002.
- [42] K.J. Friston, K.E. Stephan, Baojuan Li, and J. Daunizeau. Generalised filtering. *Mathematical Problems in Engineering*, 621670, 2010.
- [43] K.J. Friston, K.E. Stephan, T.E. Lund, A. Morcom, and S.J. Kiebel. Mixed-effects and fMRI studies. *NeuroImage*, 24:244–252, 2005.
- [44] K.J. Friston, N. Trujillo-Bareto, and J. Daunizeau. DEM: A variational treatment of dynamic systems. *NeuroImage*, 41(3):849–885, 2008.
- [45] M.I. Garrido, K.J. Friston, K.E. Stephan, S.J. Kiebel, T. Baldeweg, and J. Kilner. The functional anatomy of the MMN: A DCM study of the roving paradigm. *NeuroImage*, 42(2):936–944, 2008.
- [46] M.I. Garrido, J. Kilner, S.J. Kiebel, and K.J. Friston. Evoked brain responses are generated by feedback loops. *PNAS*, 104(52):20961–20966, 2007.
- [47] M.I. Garrido, J. Kilner, S.J. Kiebel, K.E. Stephan, and K.J. Friston. Dynamic causal modelling of evoked potentials: A reproducibility study. *NeuroImage*, 36:571–580, 2007.
- [48] D.R. Gitelman, W.D. Penny, J. Ashburner, and K.J. Friston. Modeling regional and psychophysiological interactions in fMRI: the importance of hemodynamic deconvolution. *NeuroImage*, 19:200–207, 2003.
- [49] D.E. Glaser. Variance Components. In R.S.J. Frackowiak, K.J. Friston, C. Frith, R. Dolan, K.J. Friston, C.J. Price, S. Zeki, J. Ashburner, and W.D. Penny, editors, *Human Brain Function*. Academic Press, 2nd edition, 2003.
- [50] R.N.A. Henson. Analysis of fMRI time series. In R.S.J. Frackowiak, K.J. Friston, C. Frith, R. Dolan, K.J. Friston, C.J. Price, S. Zeki, J. Ashburner, and W.D. Penny, editors, *Human Brain Function*. Academic Press, 2nd edition, 2003.
- [51] R.N.A. Henson, E. Mouchlianitis, and K.J. Friston. Meg and eeg data fusion: Simultaneous localisation of face-evoked responses. *NeuroImage*, 2009.
- [52] R.N.A. Henson and W.D. Penny. ANOVAs and SPM. Technical report, Wellcome Department of Imaging Neuroscience, 2003.
- [53] R.N.A. Henson, C.J. Price, M.D. Rugg, R. Turner, and K.J. Friston. Detecting latency differences in event-related bold responses: application to words versus non-words and initial versus repeated face presentations. *NeuroImage*, 15(1):83–97, 2002.
- [54] R.N.A. Henson, M.D. Rugg, and K.J. Friston. The choice of basis functions in event-related fMRI. *NeuroImage*, 13(6):149, June 2001. Supplement 1.
- [55] R.N.A. Henson, T. Shallice, M.L. Gorno-Tempini, and R. Dolan. Face repetition effects in implicit and explicit memory tests as measured by fMRI. *Cerebral Cortex*, 12:178–186, 2002.
- [56] C. Hutton, A. Bork, O. Josephs, R. Deichmann, J. Ashburner, and R. Turner. Image distortion correction in fMRI: A quantitative evaluation. *NeuroImage*, 16:217–240, 2002.

- [57] C. Hutton, R. Deichmann, R. Turner, and J. L. R. Andersson. Combined correction for geometric distortion and its interaction with head motion in fMRI. In *Proc. ISMRM 12*, Kyoto, Japan, 2004.
- [58] M. Jenkinson. Fast, automated, N-dimensional phase-unwrapping algorithm. *Magnetic Resonance in Medicine*, 49:193–197, 2003.
- [59] P. Jezzard and R. S. Balaban. Correction for geometric distortions in echoplanar images from B0 field variations. *Magnetic Resonance in Medicine*, 34:65–73, 1995.
- [60] J. Kahan and T. Foltynie. Understanding dcm: Ten simple rules for the clinician. *NeuroImage*, 83:542–549, 2013.
- [61] S.J. Kiebel. The general linear model. In R.S.J. Frackowiak, K.J. Friston, C. Frith, R. Dolan, K.J. Friston, C.J. Price, S. Zeki, J. Ashburner, and W.D. Penny, editors, *Human Brain Function*. Academic Press, 2nd edition, 2003.
- [62] S.J. Kiebel, J. Daunizeau, C. Phillips, and K.J. Friston. Variational bayesian inversion of the equivalent current dipole model in eeg/meg. *NeuroImage*, 39(2):728–741, 2008.
- [63] S.J. Kiebel, O. David, and K.J. Friston. Dynamic causal modelling of evoked responses in EEG/MEG with lead-field parameterization. *NeuroImage*, 30:1273–1284, 2006.
- [64] S.J. Kiebel, M.I. Garrido, and K.J. Friston. Dynamic causal modelling of evoked responses: The role of intrinsic connections. *NeuroImage*, 36:332–345, 2007.
- [65] S.J. Kiebel, M.I. Garrido, R. Moran, and K.J. Friston. Dynamic causal modelling for eeg and meg. *Cognitive Neurodynamics*, 2(2):121–136, 2008.
- [66] S.J. Kiebel, S. Klöppel, N. Weiskopf, and K.J. Friston. Dynamic causal modeling: A generative model of slice timing in fMRI. *NeuroImage*, 34:1487–1496, 2007.
- [67] A. P. Leff, T. M. Schofield, K. E. Stephan, J. T. Crinion, K. J. Friston, and C. J. Price. The cortical dynamics of intelligible speech. *J. Neurosci.*, 28:13209–13215, 2008.
- [68] Baojuan Li, J. Daunizeau, K.E. Stephan, W.D. Penny, D. Hu, and K.J. Friston. Generalised filtering and stochastic DCM for fMRI. *NeuroImage*, 58(2):442–457, 2011.
- [69] V. Litvak and K.J. Friston. Electromagnetic source reconstruction for group studies. *NeuroImage*, 42(4):1490–1498, 2008.
- [70] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Seutens. Multimodality image registration by maximisation of mutual information. *IEEE Transactions on Medical Imaging*, 16:187–197, 1997.
- [71] A.C. Marreiros, J. Daunizeau, S.J. Kiebel, and K.J. Friston. Population dynamics: Variance and the sigmoid activation function. *NeuroImage*, 42(1):147–157, 2008.
- [72] J. Mattout, C. Phillips, W.D. Penny, M. Rugg, and K.J. Friston. Meg source localization under multiple constraints: an extended Bayesian framework. *NeuroImage*, 30(3):753–767, 2006.
- [73] A. Mechelli, C.J. Price, K.J. Friston, and J. Ashburner. Voxel-based morphometry of the human brain: Methods and applications. *Current Medical Imaging Reviews*, pages 105–113, 2005.
- [74] R. Moran, S.J. Kiebel, N. Rombach, W.T. O’Connor, K.J. Murphy, R.B. Reilly, and K.J. Friston. Bayesian estimation of synaptic physiology from the spectral responses of neural masses. *NeuroImage*, 42(1):272–284, 2008.
- [75] R. Moran, S.J. Kiebel, K.E. Stephan, R.B. Reilly, J. Daunizeau, and K.J. Friston. A neural mass model of spectral responses in electrophysiology. *NeuroImage*, 37(3):706–720, 2007.

- [76] R. Moran, K.E. Stephan, R. Dolan, and K.J. Friston. Neural fields, spectral responses and lateral connections. *NeuroImage*, 55(4):1694–1708, 2011.
- [77] R. Moran, K.E. Stephan, F. Jung, R. Graf, R. Dolan, K.J. Friston, and M. Tittgemeyer. Dcm for steady state responses: a case study of anaesthesia dept. In *Proc Soc for Neuroscience*, 2009.
- [78] R. Moran, K.E. Stephan, T. Seidenbecher, H.C. Pape, R. Dolan, and K.J. Friston. Dynamic causal models of steady-state responses. *NeuroImage*, 44(3):796–811, 2009.
- [79] T.E. Nichols, M. Brett, J. Andersson, T. Wager, and J.B. Poline. Valid conjunction inference with the minimum statistic. *NeuroImage*, 25:653–660, 2005.
- [80] W.D. Penny and G. Flandin. Bayesian analysis of single-subject fMRI: SPM implementation. Technical report, Wellcome Department of Imaging Neuroscience, 2005.
- [81] W.D. Penny, G. Flandin, and N. Trujillo-Bareto. Bayesian Comparison of Spatially Regularised General Linear Models. *Human Brain Mapping*, 28:275–293, 2005.
- [82] W.D. Penny, A.P. Holmes, and K.J. Friston. Random effects analysis. In R.S.J. Frackowiak, K.J. Friston, C. Frith, R. Dolan, K.J. Friston, C.J. Price, S. Zeki, J. Ashburner, and W.D. Penny, editors, *Human Brain Function*. Academic Press, 2nd edition, 2003.
- [83] W.D. Penny, S.J. Kiebel, and K.J. Friston. Variational Bayesian Inference for fMRI time series. *NeuroImage*, 19(3):727–741, 2003.
- [84] W.D. Penny, V. Litvak, L. Fuentemilla, E. Duzel, and K.J. Friston. Dynamic Causal Models for phase coupling. *Journal of Neuroscience Methods*, 183(1):19–30, 2009.
- [85] W.D. Penny, K.E. Stephan, J. Daunizeau, M.J. Rosa, K.J. Friston, T.M. Schofield, and A.P. Leff. Comparing families of dynamic causal models. *PLoS Comput Biol*, 6(3):e1000709, 2010.
- [86] W.D. Penny, K.E. Stephan, A. Mechelli, and K.J. Friston. Comparing Dynamic Causal Models. *NeuroImage*, 22(3):1157–1172, 2004.
- [87] W.D. Penny, N. Trujillo-Bareto, and K.J. Friston. Bayesian fMRI time series analysis with spatial priors. *NeuroImage*, 24(2):350–362, 2005.
- [88] C. Phillips, J. Mattout, M.D. Rugg, P. Maquet, and K.J. Friston. An empirical Bayesian solution to the source reconstruction problem in EEG. *NeuroImage*, 24:997–1011, 2005.
- [89] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C (Second Edition)*. Cambridge, Cambridge, 1992.
- [90] A. Razi, J. Kahan, G. Rees, and K.J. Friston. Construct validation of DCM for resting state fMRI. *NeuroImage*, 106:1–14, 2015.
- [91] K.E. Stephan, W.D. Penny, J. Daunizeau, R. Moran, and K.J. Friston. Bayesian model selection for group studies. *NeuroImage*, 46(3):1004–10174, 2009.
- [92] C. Studholme, D. L. G. Hill, and D. J. Hawkes. An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognition*, 32:71–86, 1999.
- [93] P. Thévenaz, T. Blu, and M. Unser. Interpolation revisited. *IEEE Transactions on Medical Imaging*, 19(7):739–758, 2000.
- [94] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing: Part I – theory. *IEEE Transactions on Signal Processing*, 41(2):821–833, 1993.
- [95] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing: Part II – efficient design and applications. *IEEE Transactions on Signal Processing*, 41(2):834–848, 1993.
- [96] A.M. Bastos, W.M. Usrey, R.A. Adams, G.R. Mangun, P. Fries, and K.J. Friston. Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711, 2012.

- [97] W. M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. Multi-modal volume registration by maximisation of mutual information. *Medical Image Analysis*, 1(1):35–51, 1996.
- [98] I.C. Wright, P.K. McGuire, J.B. Poline, J.M. Traverre, R.M. Murray, C. Frith, R.S.J. Frackowiak, and K.J. Friston. A voxel-based method for the statistical analysis of gray and white matter density applied to schizophrenia. *NeuroImage*, 2:244–252, 1995.