

# SPM8 Manual

---

The FIL Methods Group

John Ashburner (Chapters 1-7,18-26,34)  
Guillaume Flandin (LaTeX/Subversion)  
Rik Henson (Chapters 29, 30 and 33)  
Stefan Kiebel (Chapters 11-12,15-17)  
James Kilner (Chapter 13)  
Jeremie Mattout (Chapter 14)  
Will Penny (Chapters 8-10,28-32)  
Klaas Stephan (Chapter 32)  
Chloe Hutton (Chapter 27)  
Volkmar Glauche (Chapter 35)



# Contents

<b>I Temporal processing</b>	<b>13</b>
<b>1 Slice Timing</b>	<b>15</b>
1.1 Data . . . . .	16
1.1.1 Session . . . . .	16
1.2 Number of Slices . . . . .	16
1.3 TR . . . . .	16
1.4 TA . . . . .	16
1.5 Slice order . . . . .	16
1.6 Reference Slice . . . . .	16
1.7 Filename Prefix . . . . .	17
<b>II Spatial processing</b>	<b>19</b>
<b>2 Realign</b>	<b>21</b>
2.1 Realign: Estimate . . . . .	21
2.1.1 Data . . . . .	21
2.1.2 Estimation Options . . . . .	22
2.2 Realign: Reslice . . . . .	23
2.2.1 Images . . . . .	23
2.2.2 Reslice Options . . . . .	23
2.3 Realign: Estimate & Reslice . . . . .	24
2.3.1 Data . . . . .	24
2.3.2 Estimation Options . . . . .	24
2.3.3 Reslice Options . . . . .	25
<b>3 Realign &amp; Unwarp</b>	<b>27</b>
3.1 Data . . . . .	30
3.1.1 Session . . . . .	30
3.2 Estimation Options . . . . .	30
3.2.1 Quality . . . . .	30
3.2.2 Separation . . . . .	30
3.2.3 Smoothing (FWHM) . . . . .	30
3.2.4 Num Passes . . . . .	30
3.2.5 Interpolation . . . . .	31
3.2.6 Wrapping . . . . .	31
3.2.7 Weighting . . . . .	31
3.3 Unwarp Estimation Options . . . . .	31
3.3.1 Basis Functions . . . . .	31
3.3.2 Regularisation . . . . .	31
3.3.3 Reg. Factor . . . . .	31
3.3.4 Jacobian deformations . . . . .	31
3.3.5 First-order effects . . . . .	32
3.3.6 Second-order effects . . . . .	32
3.3.7 Smoothing for unwarp (FWHM) . . . . .	32

3.3.8	Re-estimate movement params . . . . .	32
3.3.9	Number of Iterations . . . . .	32
3.3.10	Taylor expansion point . . . . .	32
3.4	Unwarp Reslicing Options . . . . .	32
3.4.1	Reslices images (unwarp)? . . . . .	32
3.4.2	Interpolation . . . . .	33
3.4.3	Wrapping . . . . .	33
3.4.4	Masking . . . . .	33
3.4.5	Filename Prefix . . . . .	33
<b>4</b>	<b>Coreg</b>	<b>35</b>
4.1	Coreg: Estimate . . . . .	35
4.1.1	Reference Image . . . . .	35
4.1.2	Source Image . . . . .	36
4.1.3	Other Images . . . . .	36
4.1.4	Estimation Options . . . . .	36
4.2	Coreg: Reslice . . . . .	36
4.2.1	Image Defining Space . . . . .	36
4.2.2	Images to Reslice . . . . .	36
4.2.3	Reslice Options . . . . .	36
4.3	Coreg: Estimate & Reslice . . . . .	37
4.3.1	Reference Image . . . . .	37
4.3.2	Source Image . . . . .	37
4.3.3	Other Images . . . . .	37
4.3.4	Estimation Options . . . . .	37
4.3.5	Reslice Options . . . . .	38
<b>5</b>	<b>Segment</b>	<b>39</b>
5.1	Data . . . . .	40
5.2	Output Files . . . . .	40
5.2.1	Grey Matter . . . . .	43
5.2.2	White Matter . . . . .	43
5.2.3	Cerebro-Spinal Fluid . . . . .	43
5.2.4	Bias Corrected . . . . .	43
5.2.5	Clean up any partitions . . . . .	43
5.3	Custom . . . . .	43
5.3.1	Tissue probability maps . . . . .	43
5.3.2	Gaussians per class . . . . .	44
5.3.3	Affine Regularisation . . . . .	44
5.3.4	Warping Regularisation . . . . .	44
5.3.5	Warp Frequency Cutoff . . . . .	45
5.3.6	Bias regularisation . . . . .	45
5.3.7	Bias FWHM . . . . .	45
5.3.8	Sampling distance . . . . .	45
5.3.9	Masking image . . . . .	45
<b>6</b>	<b>Normalise</b>	<b>47</b>
6.1	Normalise: Estimate . . . . .	48
6.1.1	Data . . . . .	48
6.1.2	Estimation Options . . . . .	48
6.2	Normalise: Write . . . . .	49
6.2.1	Data . . . . .	49
6.2.2	Writing Options . . . . .	50
6.3	Normalise: Estimate & Write . . . . .	50
6.3.1	Data . . . . .	50
6.3.2	Estimation Options . . . . .	51
6.3.3	Writing Options . . . . .	52

<b>7 Smooth</b>	<b>53</b>
7.1 Images to Smooth . . . . .	53
7.2 FWHM . . . . .	53
7.3 Data Type . . . . .	53
7.4 Filename Prefix . . . . .	53
<b>III fMRI Statistics</b>	<b>55</b>
<b>8 fMRI model specification</b>	<b>57</b>
8.1 Timing parameters . . . . .	57
8.1.1 Units for design . . . . .	59
8.1.2 Interscan interval . . . . .	59
8.1.3 Microtime resolution . . . . .	59
8.1.4 Microtime onset . . . . .	59
8.2 Data & Design . . . . .	59
8.2.1 Subject/Session . . . . .	59
8.3 Factorial design . . . . .	62
8.3.1 Factor . . . . .	63
8.4 Basis Functions . . . . .	63
8.4.1 Canonical HRF . . . . .	63
8.4.2 Other basis sets . . . . .	63
8.5 Model Interactions (Volterra) . . . . .	64
8.6 Directory . . . . .	64
8.7 Global normalisation . . . . .	64
8.8 Explicit mask . . . . .	64
8.9 Serial correlations . . . . .	65
8.10 Reviewing your design . . . . .	65
<b>9 fMRI model estimation</b>	<b>67</b>
9.1 Select SPM.mat . . . . .	67
9.2 Method . . . . .	67
9.2.1 Classical . . . . .	67
9.2.2 Bayesian 1st-level . . . . .	69
9.2.3 Bayesian 2nd-level . . . . .	72
9.3 Output files . . . . .	72
9.3.1 Classical 1st-level . . . . .	72
9.3.2 Bayesian 1st-level . . . . .	72
9.4 Model comparison . . . . .	73
<b>10 Factorial design specification</b>	<b>75</b>
10.1 Directory . . . . .	76
10.2 Design . . . . .	76
10.2.1 One-sample t-test . . . . .	76
10.2.2 Two-sample t-test . . . . .	76
10.2.3 Paired t-test . . . . .	77
10.2.4 Multiple regression . . . . .	78
10.2.5 Full factorial . . . . .	78
10.2.6 Flexible factorial . . . . .	80
10.3 Covariates . . . . .	82
10.3.1 Covariate . . . . .	82
10.4 Masking . . . . .	82
10.4.1 Threshold masking . . . . .	82
10.4.2 Implicit Mask . . . . .	83
10.4.3 Explicit Mask . . . . .	83
10.5 Global calculation . . . . .	83
10.5.1 Omit . . . . .	83

10.5.2 User . . . . .	83
10.5.3 Mean . . . . .	83
10.6 Global normalisation . . . . .	84
10.6.1 Overall grand mean scaling . . . . .	84
10.6.2 Normalisation . . . . .	84
<b>IV EEG/MEG</b>	<b>87</b>
<b>11 SPM for EEG/MEG overview</b>	<b>89</b>
<b>12 EEG/MEG preprocessing — Tutorial</b>	<b>91</b>
12.1 ERP analysis . . . . .	91
12.1.1 Convert . . . . .	91
12.1.2 Epoch . . . . .	91
12.1.3 Filter . . . . .	93
12.1.4 Downsample . . . . .	93
12.1.5 Artefacts . . . . .	93
12.1.6 Averaging . . . . .	93
12.2 Other useful functions . . . . .	93
12.2.1 Time-Frequency . . . . .	93
12.2.2 Average TF . . . . .	94
<b>13 EEG/MEG preprocessing — Reference</b>	<b>95</b>
13.1 Conversion of data . . . . .	95
13.1.1 Converting data yourself . . . . .	95
13.2 Converting CNT or BDF files . . . . .	96
13.2.1 Syntax . . . . .	96
13.3 BDF data . . . . .	96
13.3.1 Syntax . . . . .	96
13.4 CNT data . . . . .	97
13.4.1 Syntax . . . . .	97
13.5 The MEEG SPM format . . . . .	97
13.5.1 channels . . . . .	98
13.5.2 events . . . . .	98
13.5.3 filter . . . . .	99
13.5.4 threshold . . . . .	99
13.6 Reading of data . . . . .	99
13.6.1 Syntax . . . . .	99
13.7 The channel template file . . . . .	100
13.7.1 Structure . . . . .	100
13.7.2 Creating your own channel template file . . . . .	100
13.8 Epoching the data . . . . .	101
13.8.1 Syntax . . . . .	101
13.9 Filtering the data . . . . .	101
13.9.1 Syntax . . . . .	101
13.10 Artefact detection and rejection . . . . .	102
13.10.1 Syntax . . . . .	102
13.11 Downsampling . . . . .	103
13.11.1 Syntax . . . . .	103
13.12 Rereferencing . . . . .	103
13.12.1 Syntax . . . . .	103
13.13 Grand mean . . . . .	104
13.13.1 Syntax . . . . .	104
13.14 Merge . . . . .	104
13.14.1 Syntax . . . . .	104
13.15 Time-frequency decomposition . . . . .	104

13.15.1 Syntax . . . . .	105
13.16 Averaging . . . . .	105
13.16.1 Syntax . . . . .	105
13.17 Linear combinations of epochs . . . . .	105
13.17.1 Syntax . . . . .	105
13.18 Mixing of channels . . . . .	106
13.18.1 Syntax . . . . .	106
13.19 Weighting of the time-series . . . . .	106
13.19.1 Syntax . . . . .	106
13.20 Displaying data . . . . .	107
<b>14 Spatial projections</b>	<b>109</b>
14.1 Distributed linear models . . . . .	109
14.2 Equivalent current dipoles . . . . .	109
14.3 Interpolation on scalp . . . . .	109
<b>15 3D source reconstruction: Imaging approach</b>	<b>111</b>
15.1 Introduction . . . . .	111
15.2 Data structure . . . . .	112
15.3 Source space modeling ( <i>mesh</i> ) . . . . .	112
15.4 Data Registration ( <i>datareg</i> ) . . . . .	113
15.5 Forward computation ( <i>forward</i> ) . . . . .	113
15.6 Inverse reconstruction ( <i>inverse</i> ) . . . . .	114
<b>16 M/EEG modelling and statistics</b>	<b>115</b>
16.1 Preliminary remarks . . . . .	115
16.2 How epoched time-series are analysed in SPM . . . . .	115
16.3 1st level . . . . .	116
16.3.1 The aim . . . . .	116
16.3.2 Start . . . . .	116
16.3.3 All options . . . . .	116
16.3.4 How many factors? . . . . .	117
16.3.5 Factor names and # of levels for factors . . . . .	117
16.3.6 Select design component . . . . .	117
16.3.7 Data . . . . .	117
16.3.8 'Estimation' . . . . .	117
16.3.9 Results . . . . .	117
16.3.10 Display . . . . .	118
16.3.11 ERP/ERF . . . . .	118
16.3.12 Multiple subjects . . . . .	118
16.4 2nd level models . . . . .	118
16.4.1 All options . . . . .	118
16.4.2 Factors . . . . .	118
16.4.3 Design partitions and design components . . . . .	118
16.4.4 Covariance components . . . . .	119
16.4.5 Data . . . . .	119
16.4.6 Estimation and Results . . . . .	119
<b>17 Equivalent current Dipole fitting</b>	<b>121</b>
17.1 Necessary data . . . . .	121
17.2 Model building . . . . .	121
17.3 Dipole fitting . . . . .	122
17.4 Result display . . . . .	123
17.5 Preparing the *.pol files . . . . .	123

<b>18 Dynamic Causal Modelling for evoked responses</b>	<b>125</b>
18.1 Calling DCM for ERP/ERF . . . . .	125
18.2 cd, load and save . . . . .	126
18.3 Data selection . . . . .	126
18.4 Spatial model specification . . . . .	126
18.5 Connectivity model specification . . . . .	127
18.6 Estimation . . . . .	127
18.7 Results . . . . .	127
<b>V Utilities</b>	<b>129</b>
<b>19 Display Image</b>	<b>131</b>
19.1 Image to Display . . . . .	132
<b>20 Check Registration</b>	<b>135</b>
20.1 Images to Display . . . . .	135
<b>21 Image Calculator</b>	<b>137</b>
21.1 Input Images . . . . .	137
21.2 Output Filename . . . . .	137
21.3 Output Directory . . . . .	137
21.4 Expression . . . . .	138
21.5 Options . . . . .	138
21.5.1 Data Matrix . . . . .	138
21.5.2 Masking . . . . .	138
21.5.3 Interpolation . . . . .	138
21.5.4 Data Type . . . . .	138
<b>22 DICOM Import</b>	<b>139</b>
22.1 DICOM files . . . . .	139
22.2 Directory structure for converted files . . . . .	139
22.3 Output directory . . . . .	139
22.4 Conversion options . . . . .	140
22.4.1 Output image format . . . . .	140
22.4.2 Use ICEDims in filename . . . . .	140
<b>23 MINC Import</b>	<b>141</b>
23.1 MINC files . . . . .	141
23.2 Options . . . . .	141
23.2.1 Data Type . . . . .	141
23.2.2 NIFTI Type . . . . .	141
<b>24 ECAT Import</b>	<b>143</b>
24.1 ECAT files . . . . .	143
24.2 Options . . . . .	143
24.2.1 NIFTI Type . . . . .	143
<b>25 Deformations</b>	<b>145</b>
25.1 Composition . . . . .	145
25.1.1 Imported _sn.mat . . . . .	145
25.1.2 DARTEL flow . . . . .	146
25.1.3 Deformation Field . . . . .	146
25.1.4 Identity . . . . .	146
25.1.5 Inverse . . . . .	146
25.1.6 Composition . . . . .	152
25.2 Save as . . . . .	159
25.3 Apply to . . . . .	159

25.4 Interpolation . . . . .	159
<b>VI Tools</b>	<b>161</b>
<b>26 High-Dimensional Warping</b>	<b>163</b>
26.1 Subjects . . . . .	163
26.1.1 Subject . . . . .	163
26.2 Bias Correction Options . . . . .	164
26.2.1 Iterations . . . . .	164
26.2.2 Bias FWHM . . . . .	164
26.2.3 Bias regularisation . . . . .	164
26.2.4 Levenberg-Marquardt regularisation . . . . .	164
26.3 Warping Options . . . . .	164
26.3.1 Iterations . . . . .	164
26.3.2 Warping regularisation . . . . .	164
<b>27 DARTEL Tools</b>	<b>165</b>
27.1 Initial Import . . . . .	166
27.1.1 Parameter Files . . . . .	166
27.1.2 Output Directory . . . . .	166
27.1.3 Bounding box . . . . .	166
27.1.4 Voxel size . . . . .	167
27.1.5 Image option . . . . .	167
27.1.6 Grey Matter . . . . .	167
27.1.7 White Matter . . . . .	167
27.1.8 CSF . . . . .	167
27.2 Run DARTEL (create Templates) . . . . .	167
27.2.1 Images . . . . .	167
27.2.2 Settings . . . . .	167
27.3 Run DARTEL (existing Templates) . . . . .	169
27.3.1 Images . . . . .	170
27.3.2 Settings . . . . .	170
27.4 Create Warped . . . . .	171
27.4.1 Flow fields . . . . .	171
27.4.2 Images . . . . .	171
27.4.3 Modulation . . . . .	171
27.4.4 Time Steps . . . . .	171
27.4.5 Interpolation . . . . .	171
27.5 Jacobian determinants . . . . .	172
27.5.1 Flow fields . . . . .	172
27.5.2 Time Steps . . . . .	172
27.6 Create Inverse Warped . . . . .	172
27.6.1 Flow fields . . . . .	172
27.6.2 Images . . . . .	172
27.6.3 Time Steps . . . . .	172
27.6.4 Interpolation . . . . .	172
27.7 Kernel Utilities . . . . .	172
27.7.1 Generate Residuals . . . . .	173
27.7.2 Kernel from Resids . . . . .	173
27.7.3 Kernel from Flows . . . . .	174

<b>28 FieldMap Toolbox</b>	<b>175</b>
28.1 Introduction . . . . .	175
28.1.1 Latest News . . . . .	175
28.2 Creating Field Maps Using the FieldMap GUI . . . . .	176
28.2.1 Create field map in Hz . . . . .	176
28.2.2 Create voxel displacement map (VDM) and unwarp EPI . . . . .	178
28.3 Using the spm_defaults file . . . . .	179
28.4 Using the SPM5 User Interface . . . . .	181
28.5 Using the FieldMap in Batch scripts . . . . .	181
28.6 Using the VDM file with Unwarp . . . . .	181
28.7 Appendices . . . . .	182
28.7.1 Processing Hz field maps . . . . .	182
28.7.2 Converting Hz field map to VDM . . . . .	182
28.7.3 Matching field map data to EPI data . . . . .	183
<b>VII Data sets and examples</b>	<b>185</b>
<b>29 Auditory fMRI data</b>	<b>187</b>
29.1 Spatial pre-processing . . . . .	187
29.1.1 Realignment . . . . .	187
29.1.2 Coregistration . . . . .	189
29.1.3 Segmentation . . . . .	192
29.1.4 Normalize . . . . .	194
29.1.5 Smoothing . . . . .	194
29.2 Model specification, review and estimation . . . . .	196
29.2.1 Estimate . . . . .	196
29.3 Inference . . . . .	199
29.3.1 Contrast manager . . . . .	199
29.3.2 Masking . . . . .	200
29.3.3 Thresholds . . . . .	200
29.3.4 Files . . . . .	200
29.3.5 Maximum Intensity Projections . . . . .	201
29.3.6 Design matrix . . . . .	201
29.3.7 Statistical tables . . . . .	201
29.3.8 Plotting responses at a voxel . . . . .	203
29.3.9 Overlays . . . . .	204
29.3.10 Miscellaneous . . . . .	206
29.4 Bayesian analysis . . . . .	206
29.4.1 Specification . . . . .	206
29.4.2 Estimation . . . . .	207
29.4.3 Inference . . . . .	207
<b>30 Face data</b>	<b>211</b>
30.1 Spatial pre-processing . . . . .	211
30.1.1 Display . . . . .	211
30.1.2 Realignment . . . . .	211
30.1.3 Slice timing correction . . . . .	215
30.1.4 Coregistration . . . . .	215
30.1.5 Segmentation . . . . .	217
30.1.6 Normalize . . . . .	218
30.1.7 Smoothing . . . . .	219
30.2 Modelling categorical responses . . . . .	221
30.2.1 Estimate . . . . .	223
30.2.2 Inference for categorical design . . . . .	223
30.2.3 Statistical tables . . . . .	223
30.2.4 F-contrasts . . . . .	225

30.2.5 F-contrasts for testing effects of movement . . . . .	229
30.3 Modelling parametric responses . . . . .	229
30.3.1 Estimate . . . . .	231
30.3.2 Plotting parametric responses . . . . .	231
30.4 Bayesian analysis . . . . .	234
30.4.1 Specification . . . . .	234
30.4.2 Estimation . . . . .	235
30.4.3 Inference . . . . .	235
<b>31 Face group data</b>	<b>239</b>
31.1 Introduction . . . . .	239
31.2 Data . . . . .	239
31.3 Canonical HRF . . . . .	240
31.4 Informed basis set . . . . .	242
31.4.1 Nonsphericity . . . . .	244
31.4.2 Informed Results . . . . .	244
31.4.3 T- and F-contrasts . . . . .	247
31.5 FIR basis set . . . . .	251
31.5.1 Nonsphericity again . . . . .	252
31.5.2 FIR Results . . . . .	253
<b>32 Verbal Fluency PET data</b>	<b>259</b>
32.1 Introduction . . . . .	259
32.2 Single subject . . . . .	259
32.3 Multiple subjects . . . . .	260
32.3.1 Subject and Condition design . . . . .	262
32.3.2 Subject and Time design . . . . .	262
32.3.3 Subject by Condition design . . . . .	264
32.3.4 Contrast manager . . . . .	266
32.3.5 Masking and thresholds . . . . .	268
32.3.6 MIPs and results tables . . . . .	269
32.3.7 Small volume correction . . . . .	271
32.3.8 Extracting data from regions . . . . .	271
32.3.9 Inclusive Masking . . . . .	273
32.3.10 Conjunctions . . . . .	273
<b>33 Dynamic Causal Modeling for fMRI</b>	<b>277</b>
33.1 Theoretical background . . . . .	277
33.2 Bayesian model selection . . . . .	280
33.3 Practical example . . . . .	282
33.3.1 Defining the GLM . . . . .	282
33.3.2 Extracting time series . . . . .	282
33.3.3 Specifying and estimating the DCM . . . . .	283
33.3.4 Comparing models . . . . .	285
<b>34 Multimodal face-evoked responses</b>	<b>289</b>
34.1 Overview . . . . .	289
34.2 Paradigm and Data . . . . .	289
34.2.1 Structural MRI . . . . .	289
34.2.2 EEG data . . . . .	290
34.2.3 MEG data . . . . .	291
34.2.4 fMRI data . . . . .	292
34.3 Getting Started . . . . .	292
34.4 EEG analysis . . . . .	292
34.4.1 Preprocessing the EEG data . . . . .	292
34.4.2 Basic ERPs . . . . .	293
34.4.3 3D SPMs (Sensor Maps over Time) . . . . .	293

34.4.4 3D "imaging" reconstruction . . . . .	300
34.5 MEG analysis . . . . .	304
34.5.1 Preprocessing the MEG data . . . . .	304
34.5.2 Time-Frequency Analysis . . . . .	307
34.5.3 2D Time-Frequency SPMs . . . . .	308
34.5.4 "Imaging" reconstruction of differential power . . . . .	310
34.6 fMRI analysis . . . . .	315
34.6.1 Preprocessing the fMRI data . . . . .	315
34.6.2 Statistical analysis of fMRI data . . . . .	315
34.7 References . . . . .	319
<b>35 Using DARTEL</b>	<b>321</b>
35.1 Using DARTEL for VBM . . . . .	321
35.2 Using DARTEL to Spatially Normalise to MNI Space . . . . .	326
35.2.1 Affine transform of DARTEL template to MNI space . . . . .	326
35.2.2 Combining deformations . . . . .	328
35.3 Warping Images to Existing Templates . . . . .	329
35.4 Warping one individual to match another . . . . .	329
<b>VIII Batching and User Interface</b>	<b>333</b>
<b>36 Batch interface</b>	<b>335</b>
36.1 Batch tutorial . . . . .	335
36.1.1 Study specific input data . . . . .	336
36.1.2 Necessary processing steps . . . . .	336
36.1.3 Add modules to the batch . . . . .	336
36.1.4 Configure subject-independent data . . . . .	337
36.1.5 Data flow . . . . .	339
36.1.6 Entering subject-specific data . . . . .	341
36.2 Advanced features . . . . .	342
36.2.1 Multiple sessions . . . . .	342
36.2.2 Command line interface . . . . .	342
36.3 SPM5 to matlabbatch transition guide . . . . .	342
36.3.1 Code Reorganisation . . . . .	344
36.3.2 Interfaces between SPM and Matlabbatch . . . . .	344
36.4 Configuration Code Details . . . . .	344
36.4.1 Virtual Outputs . . . . .	345
36.4.2 SPM Startup . . . . .	345
36.4.3 Defaults Settings . . . . .	345
36.4.4 Toolbox Migration . . . . .	345
36.5 Utilities . . . . .	345
36.5.1 Batch Utilities . . . . .	345
36.5.2 Matlab Code Generation . . . . .	346
36.5.3 Configuration Management . . . . .	346

# **Part I**

## **Temporal processing**



# Chapter 1

## Slice Timing

### Contents

---

<b>1.1</b>	<b>Data</b>	16
1.1.1	Session	16
<b>1.2</b>	<b>Number of Slices</b>	16
<b>1.3</b>	<b>TR</b>	16
<b>1.4</b>	<b>TA</b>	16
<b>1.5</b>	<b>Slice order</b>	16
<b>1.6</b>	<b>Reference Slice</b>	16
<b>1.7</b>	<b>Filename Prefix</b>	17

---

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 physio-logical signal change was present at frequencies higher than our typical Nyquist (0.25 HZ).

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 acquisition correct.

## 1.2 Number of Slices

Enter the number of slices

## 1.3 TR

Enter the TR in seconds

## 1.4 TA

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

## 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]
```

## 1.6 Reference Slice

Enter the reference slice

## 1.7 Filename Prefix

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



## Part II

# Spatial processing



# Chapter 2

## Realign

### Contents

---

<b>2.1</b>	<b>Realign: Estimate</b>	<b>21</b>
2.1.1	Data	21
2.1.2	Estimation Options	22
<b>2.2</b>	<b>Realign: Reslice</b>	<b>23</b>
2.2.1	Images	23
2.2.2	Reslice Options	23
<b>2.3</b>	<b>Realign: Estimate &amp; Reslice</b>	<b>24</b>
2.3.1	Data	24
2.3.2	Estimation Options	24
2.3.3	Reslice Options	25

---

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 [22]. 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). 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 [22].

#### 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 give 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 give 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 better interpolation, but they are slower because they use more neighbouring voxels [62, 63, 64].

### 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. 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. Bilinear Interpolation is probably OK for PET, but not so suitable for fMRI because higher degree interpolation generally gives better results [62, 63, 64]. Although higher degree methods provide better interpolation, but they are slower because they use more neighbouring voxels. Fourier Interpolation [20, 17] 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 [22]. 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) [5]. 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 [62, 63, 64].

### 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. 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. Bilinear Interpolation is probably OK for PET, but not so suitable for fMRI because higher degree interpolation generally gives better results [62, 63, 64]. Although higher degree methods provide better interpolation, but they are slower because they use more neighbouring voxels. Fourier Interpolation [20, 17] 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>30</b>
3.1.1 Session	30
<b>3.2 Estimation Options</b>	<b>30</b>
3.2.1 Quality	30
3.2.2 Separation	30
3.2.3 Smoothing (FWHM)	30
3.2.4 Num Passes	30
3.2.5 Interpolation	31
3.2.6 Wrapping	31
3.2.7 Weighting	31
<b>3.3 Unwarp Estimation Options</b>	<b>31</b>
3.3.1 Basis Functions	31
3.3.2 Regularisation	31
3.3.3 Reg. Factor	31
3.3.4 Jacobian deformations	31
3.3.5 First-order effects	32
3.3.6 Second-order effects	32
3.3.7 Smoothing for unwarp (FWHM)	32
3.3.8 Re-estimate movement params	32
3.3.9 Number of Iterations	32
3.3.10 Taylor expansion point	32
<b>3.4 Unwarp Reslicing Options</b>	<b>32</b>
3.4.1 Reslices images (unwarp)?	32
3.4.2 Interpolation	33
3.4.3 Wrapping	33
3.4.4 Masking	33
3.4.5 Filename Prefix	33

---

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). ".mat" files are written for each of the input images. 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.

The paper [2] is unfortunately a bit old now and describes none of the newer features. Hopefully we'll have a second paper out any decade now.

See also 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

unwarped (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 thought 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 you can say "sod Unwarp" (but not to our faces!).

## 3.1 Data

Data sessions to unwarped.

### 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 give 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 give 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 being written in a different space.

Nearest Neighbour - Fastest, but not normally recommended.

Bilinear Interpolation - OK for PET, or realigned fMRI.

B-spline Interpolation/\* [62] - Better quality (but slower) interpolation, especially with higher degree splines. Do not use B-splines when there is any region of NaN or Inf in the images.

### 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:

$$\begin{bmatrix} 4 & 4 \\ 4 & 5 \\ 5 & 5 \end{bmatrix}$$

### 3.3.7 Smoothing for unwarped (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 Reslices 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 - Fastest, but not normally recommended.

Bilinear Interpolation - OK for PET, or realigned fMRI. B-spline Interpolation[62]

- Better quality (but slower) interpolation, especially with higher degree splines. Do not use B-splines when there is any region of NaN or Inf in the images.

### 3.4.3 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).

### 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

## Coreg

### Contents

---

<b>4.1</b>	<b>Coreg: Estimate</b>	<b>35</b>
4.1.1	Reference Image	35
4.1.2	Source Image	36
4.1.3	Other Images	36
4.1.4	Estimation Options	36
<b>4.2</b>	<b>Coreg: Reslice</b>	<b>36</b>
4.2.1	Image Defining Space	36
4.2.2	Images to Reslice	36
4.2.3	Reslice Options	36
<b>4.3</b>	<b>Coreg: Estimate &amp; Reslice</b>	<b>37</b>
4.3.1	Reference Image	37
4.3.2	Source Image	37
4.3.3	Other Images	37
4.3.4	Estimation Options	37
4.3.5	Reslice Options	38

---

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 Coreg: Estimate

The registration method used here is based on work by Collignon et al [16]. 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 [59].

#### Objective Function

Registration involves finding parameters that either maximise or minimise some objective function. For inter-modal registration, use Mutual Information [16, 65]\*, Normalised Mutual Information [61], or Entropy Correlation Coefficient [47]. 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 Coreg: 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). Bilinear 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 [62, 63, 64] 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 Coreg: Estimate & Reslice

The registration method used here is based on work by Collignon et al [16]. 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 [59].

### Objective Function

Registration involves finding parameters that either maximise or minimise some objective function. For inter-modal registration, use Mutual Information [16, 65]\*/, Normalised Mutual Information [61], or Entropy Correlation Coefficient [47]. 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). Bilinear 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 [62, 63, 64] 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</b>	<b>Data</b>	<b>40</b>
<b>5.2</b>	<b>Output Files</b>	<b>40</b>
5.2.1	Grey Matter	43
5.2.2	White Matter	43
5.2.3	Cerebro-Spinal Fluid	43
5.2.4	Bias Corrected	43
5.2.5	Clean up any partitions	43
<b>5.3</b>	<b>Custom</b>	<b>43</b>
5.3.1	Tissue probability maps	43
5.3.2	Gaussians per class	44
5.3.3	Affine Regularisation	44
5.3.4	Warping Regularisation	44
5.3.5	Warp Frequency Cutoff	45
5.3.6	Bias regularisation	45
5.3.7	Bias FWHM	45
5.3.8	Sampling distance	45
5.3.9	Masking image	45

---

Segment, bias correct and spatially normalise - all in the same model [9]. This function can be used for bias correcting, spatially normalising or segmenting your 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 [67, 50, 7, 8]. 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.

## 5.1 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.

## 5.2 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, although this may need some empirical evaluations.

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 5.1). You can also produce spatially normalised versions - both with ( $mwc^*$ ) and without ( $wc^*$ ) modulation (see Figure 5.2). 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 (both un-modulated and modulated). 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:

$$\begin{pmatrix} \frac{dx'}{dx} & \frac{dx'}{dy} & \frac{dx'}{dz} \\ \frac{dy'}{dx} & \frac{dy'}{dy} & \frac{dy'}{dz} \\ \frac{dz'}{dx} & \frac{dz'}{dy} & \frac{dz'}{dz} \end{pmatrix}$$

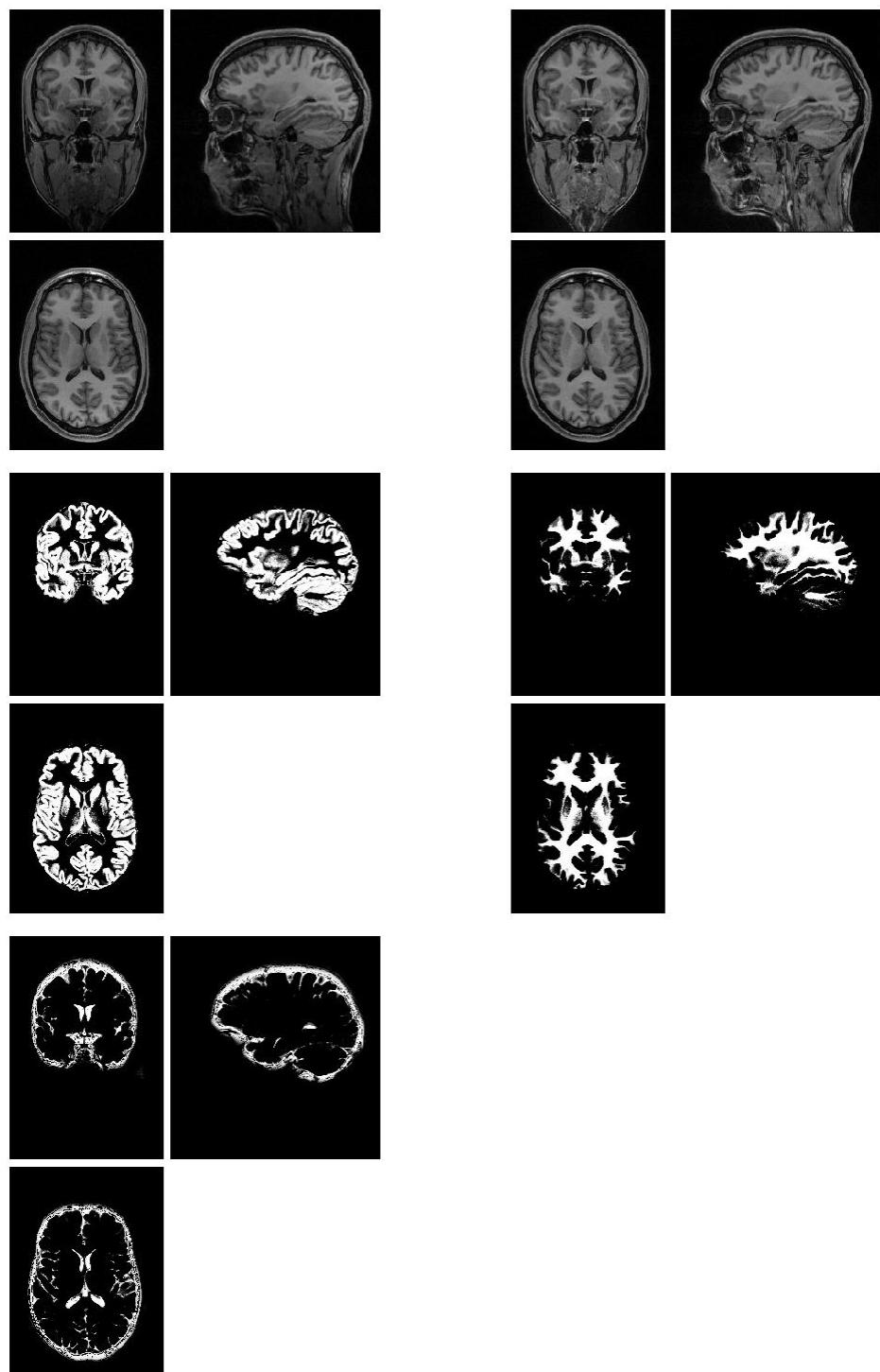


Figure 5.1: Segmentation results. These are the results that can be obtained in the original space of the image (i.e. the results that are not spatially normalised). Top left: original image ( $X.\text{img}$ ). Top right: bias corrected image ( $mX.\text{img}$ ). Middle and bottom rows: segmented grey matter ( $c1X.\text{img}$ ), white matter ( $c2X.\text{img}$ ) and CSF ( $c3X.\text{img}$ ).



Figure 5.2: 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`).

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 5.2).

### 5.2.1 Grey Matter

Options to produce grey matter images: c1\*.img, wc1\*.img and mwc1\*.img.

### 5.2.2 White Matter

Options to produce white matter images: c2\*.img, wc2\*.img and mwc2\*.img.

### 5.2.3 Cerebro-Spinal Fluid

Options to produce CSF images: c3\*.img, wc3\*.img and mwc3\*.img.

### 5.2.4 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.

### 5.2.5 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.

## 5.3 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.

### 5.3.1 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.

### 5.3.2 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.

### 5.3.3 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.

### 5.3.4 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.

### 5.3.5 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.

### 5.3.6 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.

### 5.3.7 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.

### 5.3.8 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.

### 5.3.9 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.



# Chapter 6

## Normalise

### Contents

---

<b>6.1</b>	<b>Normalise: Estimate</b>	<b>48</b>
6.1.1	Data	48
6.1.2	Estimation Options	48
<b>6.2</b>	<b>Normalise: Write</b>	<b>49</b>
6.2.1	Data	49
6.2.2	Writing Options	50
<b>6.3</b>	<b>Normalise: Estimate &amp; Write</b>	<b>50</b>
6.3.1	Data	50
6.3.2	Estimation Options	51
6.3.3	Writing Options	52

---

This 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.

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 [6] . The default options result in each of the deformation fields being described

by 1176parameters, 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 primarily use is for stereotactic normalisation to facilitate inter-subject averaging and precise characterisation of functional anatomy [5] . 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 \*.img scans are written to the same subdirectory as the original \*.img, prefixed with a 'w' (i.e. w\*.img). The details of the transformations are displayed in the results window, and the parameters are saved in the "\*\_sn.mat" file.

## 6.1 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'.

### 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.

**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.

### 6.1.2 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.

## 6.2 Normalise: Write

Allows previously estimated warps (stored in imagename'.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.

**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.

### 6.2.2 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.

Nearest Neighbour: - Fastest, but not normally recommended.

Bilinear Interpolation: - OK for PET, or realigned fMRI.

B-spline Interpolation: - Better quality (but slower) interpolation [62], especially with higher degree splines. Do not use B-splines when there is any region of NaN or Inf in the images.

#### 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'.

## 6.3 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.

### 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.

**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.

### 6.3.2 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.

#### 6.3.3 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.

Nearest Neighbour: - Fastest, but not normally recommended.

Bilinear Interpolation: - OK for PET, or realigned fMRI.

B-spline Interpolation: - Better quality (but slower) interpolation [62], especially with higher degree splines. Do not use B-splines when there is any region of NaN or Inf in the images.

##### 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 7

## Smooth

### Contents

---

<b>7.1</b>	<b>Images to Smooth</b>	53
<b>7.2</b>	<b>FWHM</b>	53
<b>7.3</b>	<b>Data Type</b>	53
<b>7.4</b>	<b>Filename Prefix</b>	53

---

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 \*.img and are prefixed with a 's' (i.e. s\*.img). 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 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.

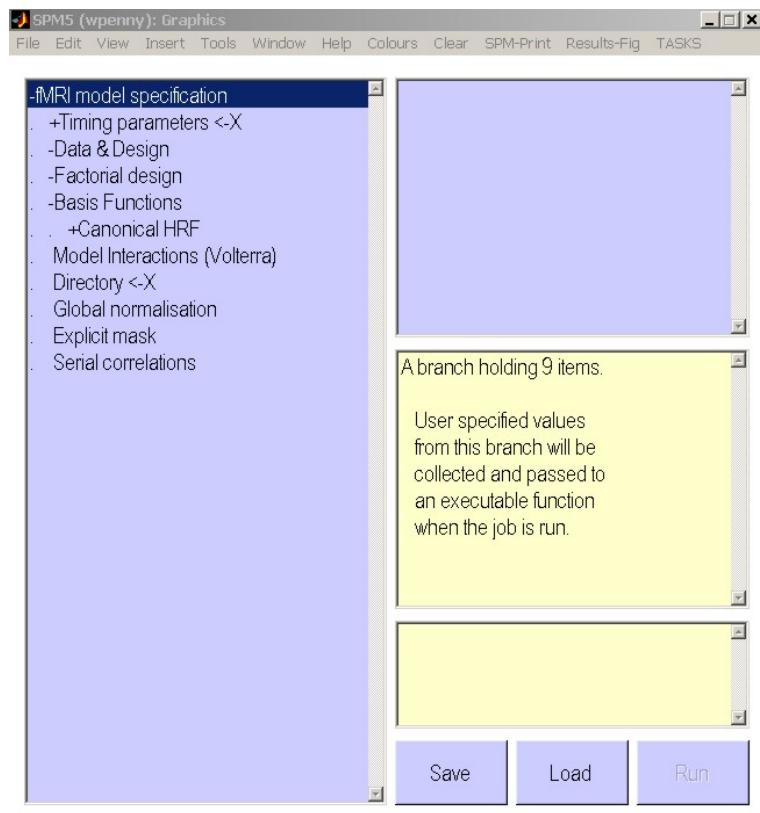


Figure 8.1: After starting SPM in fMRI mode, pressing the ‘Specify 1st-level’ button, and then double-clicking on the ‘+fMRI model specification’ text, the SPM graphics window should appear as above. The options under ‘fMRI model specification’ can be examined by clicking on them. A single click will bring up some help text in the lower subwindow (not shown in the above graphic). A double-click on options prefixed by a ‘+’ will allow you to specify options at a greater level of detail. 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, t0, is the first time-bin at which the regressors are resampled to coincide with data acquisition. If t0 = 1 then the regressors will be appropriate for the first slice. If you want to temporally realign the regressors so that they match responses in the middle slice then make t0 = 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 t0 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, t0=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.

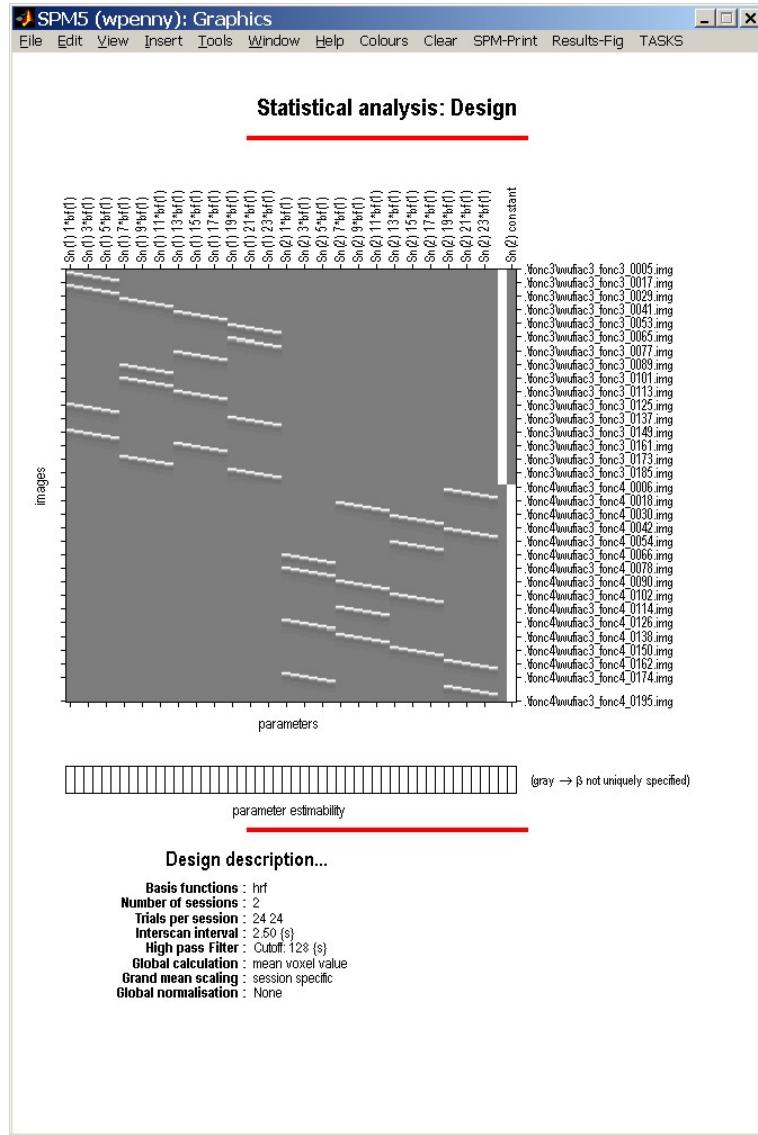


Figure 8.2: *Design matrix for fMRI data from two sessions. There are 24 experimental conditions for each session. The last two columns model the average activity in each session, giving a total of 50 regressors. There are 191 fMRI scans for each session. The overall design matrix therefore has 382 rows and 50 columns.*

## 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 (in seconds). 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.

**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 [15, 14] for further details of parametric modulations.

---

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

### 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 mutliple 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 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 [32] 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 [34].

### 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 [31] 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 [26] 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 [29, 55] 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.

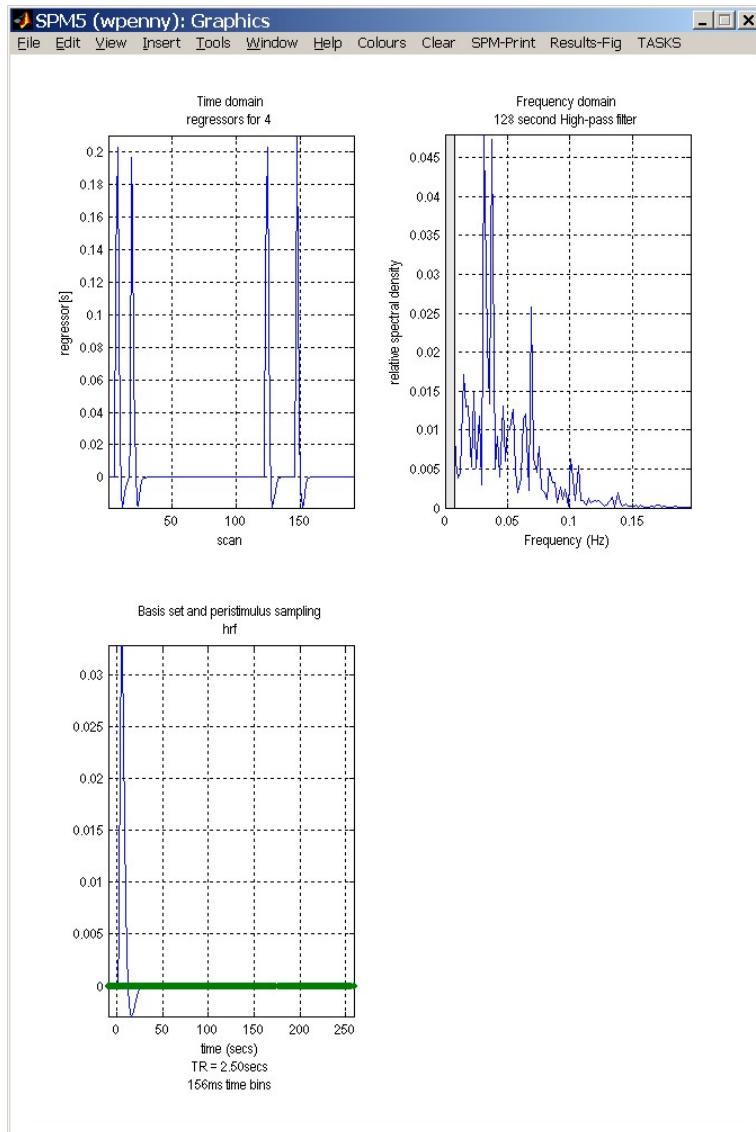


Figure 8.3: After pressing ‘Review’, selecting the pull-down ‘Design’ menu, Explore->Session, and selecting the regressor you wish to look at, you should get a plot similar to the one above. The top row shows time and frequency domain plots of the time-series corresponding to this regressor. In this 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 (see the peak at about 0.04Hz).

# 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 that is new to 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 [29, 23] 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.

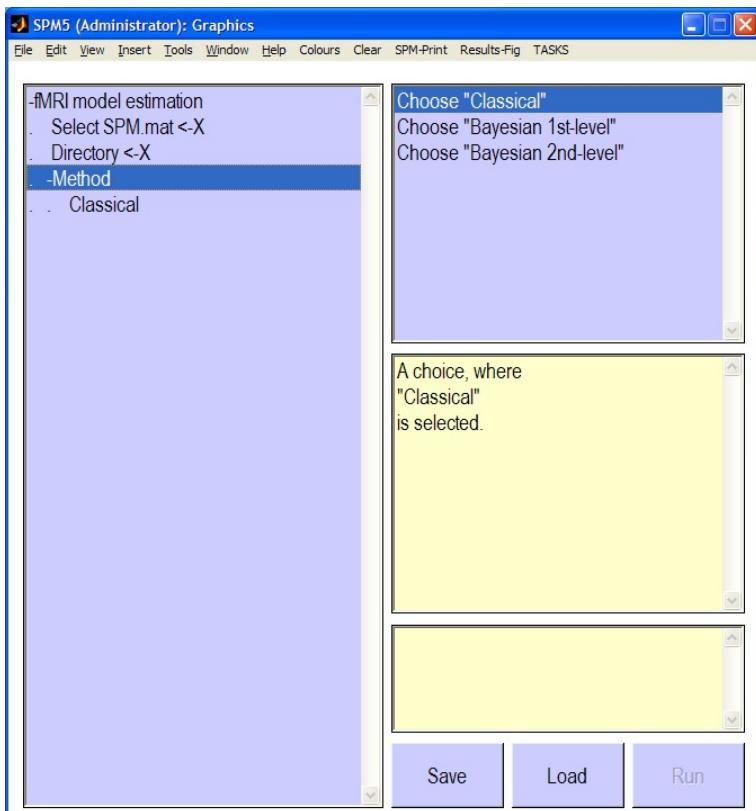


Figure 9.1: After starting SPM in fMRI mode, pressing the ‘Estimate’ button, and then double-clicking on the ‘+fMRI model estimation’ text, the SPM graphics window should appear as above. The options under ‘-fMRI model estimation’ can be examined by clicking on them. A single click will bring up some help text in the lower subwindow (not shown in the above graphic). A double-click on options prefixed by a ‘+’ will allow you to specify options at a greater level of detail. 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.

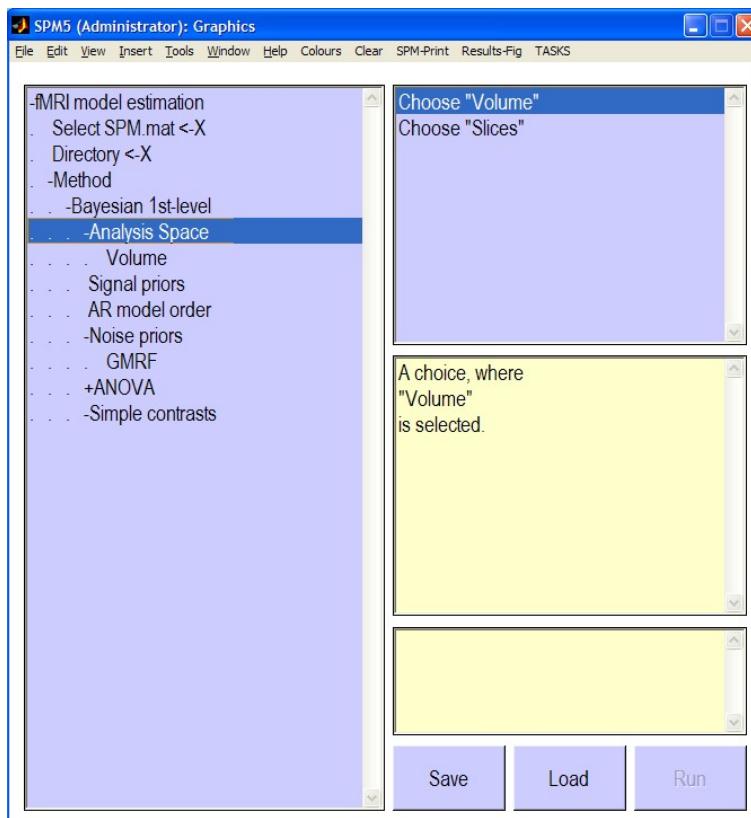


Figure 9.2: After choosing Bayesian 1st-level under ‘Method’ and then double-clicking on the ‘+Bayesian 1st-level’ text, the SPM graphics window should appear as above. Each of the options shown above is described in this chapter.

### 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 [55, 57, 52, 53].

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) [27].

#### Analysis Space

Because estimation can be time consuming, an option is provided to analyse selected slices rather than the whole volume.

**Volume** You have selected the Volume option. SPM will analyse fMRI time series in all slices of each volume.

**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 your slice maps to be visible.

### Signal priors

- [GMRF] Gaussian Markov Random Field. 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.
- [LORETA] Low Resolution Tomography Prior. This spatial prior is very similar to the GMRF prior and is a standard choice for MEG/EEG source localisation algorithms. It does, however, have undesirable edge effects.
- [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 (see also [27]), 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) algorithm but where the weights are different at each voxel. This reflects the different noise correlations 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 [55].

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 Independent and Identically Distributed (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 three noise prior options.

- [GMRF] Gaussian Markov Random Field. 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.
- [LORETA] Low Resolution Tomography Prior. See comments on LORETA priors for regression coefficients.
- [Tissue-type] This provides an estimation of AR coefficients at each voxel that 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 [53].

## ANOVA

Perform 1st or 2nd level Analysis of Variance.

**First level** This is implemented using Bayesian model comparison as described in [53]. 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 another.

### 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. This uses an approximate reconstruction based on a Taylor series expansion described in [52]. 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 another.

### 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 different 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. It is described in detail in [27].

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.

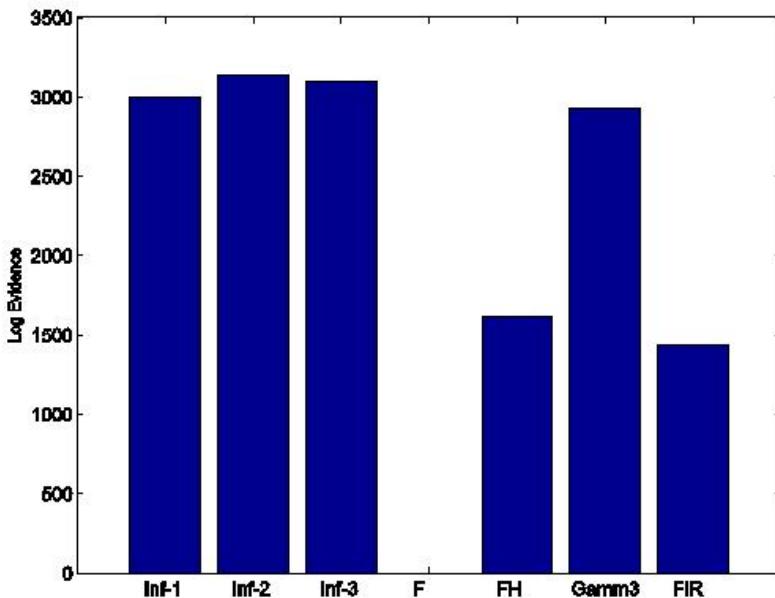


Figure 9.3: This plot shows the model evidence for a number of different hemodynamic basis sets: *Inf1* - Canonical HRF, *Inf2* - Canonical plus temporal derivative, *Inf3* - Canonical plus temporal and dispersion derivatives, *F* - Fourier, *FH* - Fourier with a Hanning Window, *Gamm3* - 3 Gamma basis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

## 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 [53] 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.



# Chapter 10

## Factorial design specification

### Contents

---

<b>10.1</b>	<b>Directory</b>	76
<b>10.2</b>	<b>Design</b>	76
10.2.1	One-sample t-test	76
10.2.2	Two-sample t-test	76
10.2.3	Paired t-test	77
10.2.4	Multiple regression	78
10.2.5	Full factorial	78
10.2.6	Flexible factorial	80
<b>10.3</b>	<b>Covariates</b>	82
10.3.1	Covariate	82
<b>10.4</b>	<b>Masking</b>	82
10.4.1	Threshold masking	82
10.4.2	Implicit Mask	83
10.4.3	Explicit Mask	83
<b>10.5</b>	<b>Global calculation</b>	83
10.5.1	Omit	83
10.5.2	User	83
10.5.3	Mean	83
<b>10.6</b>	<b>Global normalisation</b>	84
10.6.1	Overall grand mean scaling	84
10.6.2	Normalisation	84

---

This interface is used for setting up analyses of PET data. It is also used for '2nd level' or 'random effects' analysis which allow one to make a population inference. First level models can be used to produce appropriate summary data, which can then be used as raw data for a 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.

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.

A separate interface handles design configuration for 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

#### Scans

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

### 10.2.2 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 only used for PET data.

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 only used for PET data.

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

##### Pairs

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

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

##### 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 only used for PET data.

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 only used for PET data.

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

#### 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

### 10.2.5 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 only used for PET data.

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 only used for PET data.

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 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.

### 10.2.6 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 nscan x 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 can 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 can type in simply [1 2 1 2] to specify eg. just the 'Task' factor level. You do not need to eg. for the 4th subject enter the matrix [1 4; 2 4; 1 4; 2 4].

**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 only used for PET data.

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 only used for PET data.

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 row 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. Unlike SPM94/5/6, where the design was partitioned into effects of interest and nuisance effects for the computation of adjusted data and the F-statistic (which was used to thresh out voxels where there appeared to be no effects of interest), SPM5 does not partition the design in this way. The only remaining distinction between effects of interest (including covariates) and nuisance effects is their location in the design matrix, which we have retained for continuity. Pre-specified design matrix partitions can be entered.

### 10.3.1 Covariate

Add a new covariate to your experimental design

#### Vector

Vector of 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

The appropriate centering option is usually the one that corresponds to the interaction chosen, and ensures that main effects of the interacting factor aren't affected by the covariate. You are advised to choose this option, unless you have other modelling considerations.

## 10.4 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.

### 10.4.1 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

### 10.4.2 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.

### 10.4.3 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.

## 10.5 Global calculation

This option is only used for PET data.

There are three methods for estimating global effects (1) Omit (asssuming 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)

### 10.5.1 Omit

Omit

### 10.5.2 User

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

#### Global values

Enter the vector of global values

### 10.5.3 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.

## 10.6 Global normalisation

This option is only used for PET data.

Global nuisance effects are usually accounted for either by scaling the images so that they all have the same 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.

### 10.6.1 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.

## 10.6.2 Normalisation

Global nuisance effects are usually accounted for either by scaling the images so that they all have the same 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 EEG/MEG overview

Unlike previous versions, SPM5 provides for the analysis of EEG and MEG data. The initial main motivation for this big leap came from the insight that any integration of modalities like fMRI and EEG should be based on a common theoretical and practical basis. Historically, research into fMRI and EEG/MEG models and analysis has been quite divorced. The same is partially true for the EEG and MEG field. It is our hope that SPM5 provides a common analysis ground for modellers and experimentalists of both the PET/fMRI and EEG/MEG fields.

SPM for EEG/MEG was primarily developed for the analysis of epoched data. This is because we are mostly interested in experiments which perturb the system with a designed stimulus. The analysis of continuous data is typically performed for experiments without designed stimuli like in sleep or epilepsy research. Note that although SPM5 does not provide for an analysis of continuous data per se, many SPM5 routines can be used for these data.

SPM for EEG/MEG can be partitioned into four parts. These are (i) preprocessing, (ii) projection to voxel-space/source reconstruction, (iii) statistical analysis, and (iv) Dynamical Causal Modelling (DCM).

The preprocessing functions provide for simple operations that are standard in other software packages.

The projection to voxel-space is a critical step. When using a source reconstruction, it takes the analysis to brain space. But even when using a simple projection to some 2D-sensor plane, we can then use SPM functions and concepts that were developed for voxel-based data. The projection to a 2D-plane is performed using a simple interpolation and is mostly equivalent to widely used sensor-based analyses. The source reconstruction to brain space is based on models that assume many distributed dipoles in brain space. Solutions to these models typically show dispersed activity and are well-suited for SPM mass-univariate analysis approach.

The statistical analysis is one of the strong points of SPM. PET/fMRI users already familiar with the graphical representation of general linear models won't have difficulties to use SPM for the analysis of EEG/MEG data. SPM5 provides for a comprehensive range of classical linear models that can be used to model the data. These models are basically the same as used in the EEG/MEG field for random effects analyses of multiple subjects. In SPM, we assume that the data in voxel-space is a sampled version of some continuous Gaussian random field. This allows us to use Random field theory (GFT) for the correction of multiple comparisons to control family-wise error over voxels. The GFT approach has the advantage that super-threshold maxima are assessed for their significance, whereas conventional approaches have to specify a-priori the locations of expected activations.

Dynamical Causal Modelling (DCM) is a departure from the mass-univariate approach and is an extension of the SPM software package. DCM for ERP/ERFs is a generative model for evoked responses. The observed data is modelled as the spatiotemporal expression of a small hierarchical network that responds to a stimulus. Differences between evoked responses due to different stimuli are modelled as modulations of the coupling between specific areas. Importantly,

this approach is based on a neurobiologically grounded model. This allows us to obtain parameter estimates that have some physiological interpretation.

The following chapters will go through all the EEG/MEG related functionality of SPM5. All users will probably find the tutorial useful for a quick start. A further detailed description of the preprocessing functions is given in chapter 13. The 3D-source reconstruction is described in chapter 15 and some dipole fitting technique in chapter 17. In chapter 16, we guide you through the modelling of M/EEG data at the first and second level of a hierarchical linear model. Finally, in chapter 18, we describe the graphical user interface for dynamical causal modelling for evoked responses, i.e. event-related potentials (ERPs) and event-related fields (ERFs).

# Chapter 12

## EEG/MEG preprocessing — Tutorial

This tutorial will give a users guide to the pre-processing sections of SPM M/EEG. We will use an example data set collected on a 128 active electrode Biosemi EEG system. This data set is available from the SPM website. The data was recorded continuously and had three event types (event identifiers 1,2 and 4) These event types indicated the type of visual stimulus presented.

### 12.1 ERP analysis

#### 12.1.1 Convert

Convert reads the EEG data file and writes the data in a format that is used for SPM analysis. The SPM format has two components a \*.mat and \*.dat file. The \*.mat file contains the data structure D and the \*.dat is the M/EEG data.

After clicking on Convert you will be asked to select the format of your EEG or MEG data. For the example data set we select BDF. Next select the data file. For the example data we select the EEGexample.bdf. Next select the data template file from the EEG template file. This file contains a template of electrode positions in 2D. For the example data set select the bdf\_setup.mat.

NB the questions then asked depend upon the data format selected. Here we will only address the questions for the BDF format of the example data set.

For BDF files the HEOG, VEOG and any other additional recordings are saved in the EXG channels. Here we attribute labels to these recordings. For the example data set we recorded HEOG (EXG 3 4), VEOG (EXG 4 5) and recorded from the earlobes (EXG 1 2) to allow re-referencing offline. No other additional data were recorded. Therefore we enter the following:

The Convert function writes an EEGexample.mat and an EEGexample.dat file in the current directory.

#### 12.1.2 Epoch

To epoch the data click on epoching. Select the EEG mat file to epoch. Choose the peri-stimulus time window. Choose the events to epoch. Possible events will be listed in the Matlab command window. If you do not have any events in you converted data file you can input them in by reading in a new event list.

For the example data set exampleEEG.mat was selected and the following parameters were used:

Having epoched the data one could filter and downsample the epoched data as above. Epoching writes a \*.mat and \*.dat file prefixed by a 'e\_-'.

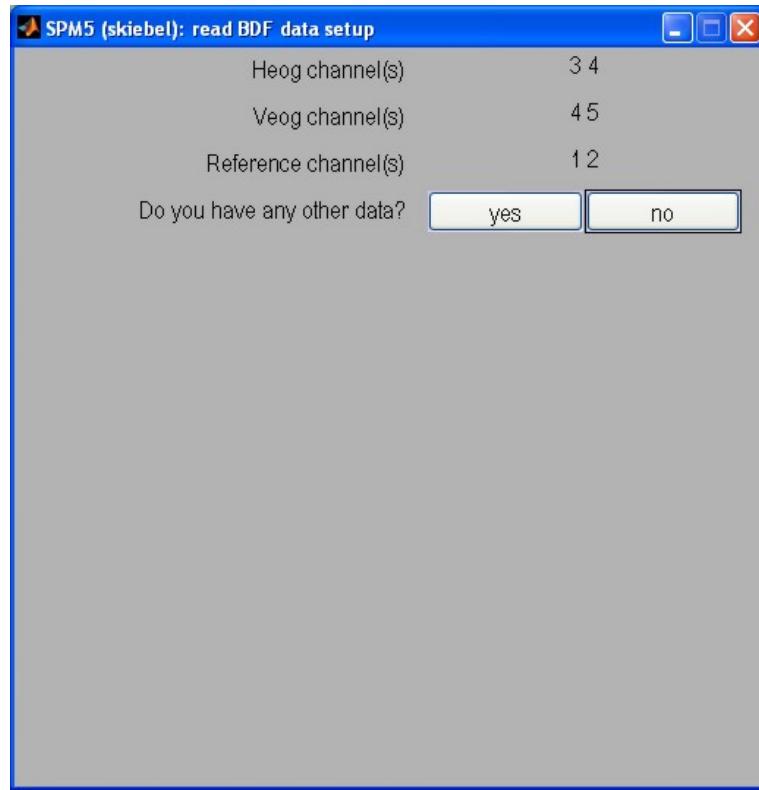


Figure 12.1: *Specifying the options for converting bdf-files*

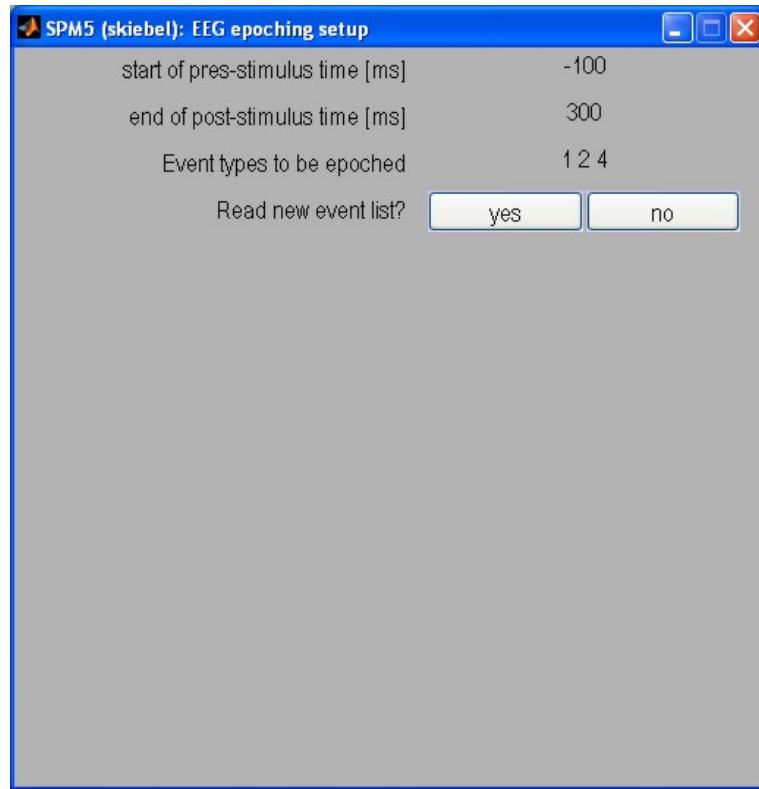


Figure 12.2: *Specifying the options for epoching the example data*

### 12.1.3 Filter

Filter filters the data using a Butterworth filter. This can be applied either before or after epoching (see 1.3). Here it is applied after.

After clicking Filter select the \*.mat file produced by Convert. For the example data set select e\_EEGexample.mat.

Next select the filter type, either lowpass or bandpass. If lowpass is selected enter the cut-off frequency. If bandpass is selected enter the two frequencies that specify the band of interest.

The Filter function writes a \*.mat and \*.dat file prefixed by a 'f'.

For the example data set e\_exampleEEG.mat was selected and a bandpass filter was used from 0.1-45 Hz.

### 12.1.4 Downsample

To downsample the data select downsample from the 'Other' pull-down menu. Select the data file to downsample. Enter the new sampling rate. For the example dataset we downsampled to 100 Hz. Downsample writes a \*.mat and \*.dat file prefixed by a 'd'.

For the example data set fe\_exampleEEG.mat was selected.

### 12.1.5 Artifacts

Two different methods of artefact removal are implemented in SPM5. One is a simple thresholding method. The other uses a robust averaging methodology to weight each time point by a function of the residuals.

To remove artefacts click on Artifacts. Select the epoched data file to analyse. If you know of bad trials or electrodes that you noted during acquisition or found using another methodology you can read them in using the 'read own artefact list'. If you want to use robust averaging click yes to the following question if not click no. To threshold channels click yes and then enter the threshold you wish to use. The thresholding has two passes. One to find bad electrodes and the second to find bad trials. If robust averaging was selected the second pass will apply the robust averaging approach but a first pass could use a thresholding method to find the bad electrodes prior to robust averaging.

Artifacts writes a \*.mat and \*.dat file prefixed by an 'a'.

For the example data set dfe\_exampleEEG.mat was selected. No artefact list was used and robust averaging was selected with a threshold of 100 v. For the robust averaging the default parameters were used.

### 12.1.6 Averaging

To produce a mean ERP click on averaging and select the EEG mat file that you wish to average. This will automatically average either ignoring the bad channels and trials or it will apply the weighting matrix calculated from robust averaging. For the example data set adfe\_exampleEEG.mat was selected.

Averaging writes a \*.mat and \*.dat file prefixed by an 'm'.

## 12.2 Other useful functions

### 12.2.1 Time-Frequency

In SPM5 it is possible to apply a time-frequency analysis to the data. This can either be applied to the averaged ERP to produce time-frequency plot of evoked oscillations. Or it can be applied to each trial and then averaged across trials to produce time-frequency plot of evoked oscillations

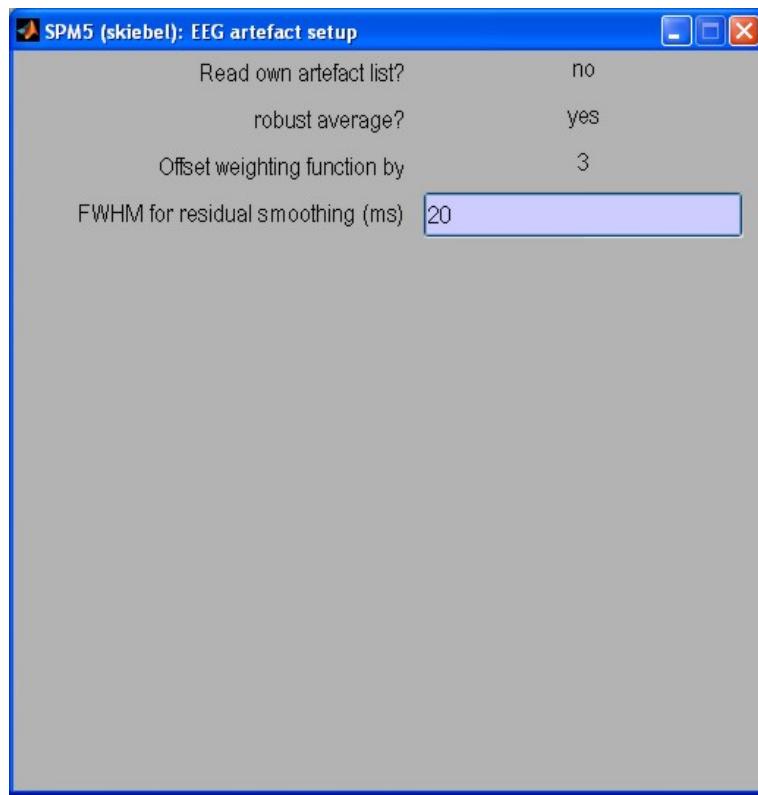


Figure 12.3: *Specifying the options for artefact detection in the example data*

and induced oscillations. To produce time-frequency plots select time-frequency from the other pull-down menu. Select the EEG mat file to analyse. Enter a vector containing the frequencies that you wish to analyse. You are then given the option to baseline correct the time-frequency data. Next enter the Morlet wavelet factor that you wish to use. The default value is 7. Next you can select the channels you wish to analyse. The default is all channels.

Time-Frequency writes two \*.mat and \*.dat files. The first, the power, is prefixed by 't1'. The second, the phase, is prefixed by 't2'.

### 12.2.2 Average TF

To average time-frequency data sets across trials select average TF from the other pull down menu. Select the T1\*.mat EEG mat file to average. As with the Averaging function Average TF writes a \*.mat and \*.dat file prefixed by an 'm'.

# Chapter 13

## EEG/MEG preprocessing — Reference

In this chapter we will describe the purpose and syntax of all SPM/MEEG preprocessing functions. These functions can be called either from SPM’s graphical user interface (GUI) or from the matlab command line. For the command line, 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 has the advantage that the input does not need to follow a specific input argument order. If an obligatory input argument is missing, the function will invoke the GUI and ask the user for the missing argument. When using the GUI, a function is called without any input argument, i.e. SPM will ask for all input arguments. If using the command line, you can specify all arguments in advance and use SPM/MEEG functions in batch mode. We provided some sample batch script (*meeg\_preprocess*) in the *man/example\_scripts/* folder of the distribution.

### 13.1 Conversion of data

The conversion of data is necessary to read data from a native machine-dependent format to a matlab-based, common SPM format. This format stores the data in a \*.dat file and all other information in a \*.mat file. The contents of the \*.mat file is a single struct with fields that contain all information about the data (described further below). Currently, SPM can deal with a few formats (s. below).

#### 13.1.1 Converting data yourself

If your format is not one of these, you need to convert the data yourself. This might sound difficult, but it is actually easier than most people think. If things go wrong, the SPM developer team is usually quick to answer questions concerning help with the conversion of data. What we can’t do though is to provide a conversion routine for every M/EEG format. The reason is that there are many formats around, which also evolve over time. To support all these formats would be simply too much work for us.

To write a conversion routine, you (or a helpful colleague) need a minimum knowledge of MatLab. To make things easier and to provide you with a starting point, we wrote a generic conversion script that can be easily modified to work with your specific data. You find this script (*meeg\_read\_data*) in the *man/example\_scripts/* folder of the distribution.

There are three parts to this script. In the first part you provide SPM with information about your data, e.g. sample rate, number of conditions, etc. The second part will read the actual data into the MatLab workspace. To do this you’ll need to write a few lines of matlab. This can be easy, if your data is in ASCII format. It’s more difficult, when the data is still in its native (binary) format. In that case you must know the file specification. Alternatively, you might be fortunate and find free third-party MatLab-software, somewhere on the internet, that does this job for you! In the third part, the data and all information is converted to the SPM format.

The final step is to generate a channel template file. This is necessary to determine the coordinates of the channels in 2D-space. These coordinates are needed for viewing data and projection to voxel-space<sup>1</sup>. See below how such a channel template file can be generated. If you’re using a standard setup, it’s likely that we or someone else have already provided such a file.

## 13.2 Converting CNT or BDF files

The conversion routine can be started either by using the GUI (Convert) or by calling the function *spm\_eeg\_convert2mat*. This function is simply a wrapper function that calls the appropriate conversion function.

### 13.2.1 Syntax

*D* = *spm\_eeg\_converteeg2mat*(*S*)

#### Input

The input struct *S* is optional and has the following optional fields:

- |                  |   |
|------------------|---|
| <b>fmt</b>       | - string that determines type of input file. Currently, this string can be either 'CNT', 'BDF', 'EGI-txt' |
| <b>Mname</b>     | - char matrix of input file name(s)   |
| <b>Fchannels</b> | - String containing name of channel template file   |

#### Output

The output struct *D* contains the header struct of the converted file. This struct has been written to a \*.mat file of the same name as the converted file. The data has been written to a corresponding \*.dat file.

## 13.3 BDF data

The Biosemi Data format (BDF) can be converted with the function *spm\_eeg\_rdata\_bdf*. There is no explicit reference electrode, because the Biosemi system uses reference-free measurements. Nearly all information is contained in the raw \*.bdf file. The only information that is not in the file is the actual usage of the 8 external channels. Typically these are used for EOG and some reference measurement. These information must be supplied to SPM. Keep in mind that a false declaration of external channels can severely degrade the quality of your data.

The conversion routine can be started by calling the function *spm\_eeg\_rdata\_bdf*.

### 13.3.1 Syntax

*D* = *spm\_eeg\_rdata\_bdf*(*S*)

#### Input

The input struct *S* is optional and has the following optional fields:

---

<sup>1</sup>If you don’t want to look at your data, or project to 2D voxel-space, you can actually proceed without this channel template file.

<b>Fdata</b>	-	filename of bdf-file
<b>Fchannels</b>	-	String containing name of channel template file
<b>Cheog</b>	-	One or two indices of external channels used for HEOG. Valid indices lie between 1 and 8.
<b>Cveog</b>	-	indices (1 - 8) of external channels used for VEOG. Valid indices lie between 1 and 8.
<b>Creference</b>	-	indices (1 - 8) of external channels used for reference. Valid indices lie between 1 and 8.

### Output

The output struct  $D$  contains the header struct of the converted file. This struct has been written to a \*.mat file of the same name as the converted file. The data has been written to a corresponding \*.dat file.

## 13.4 CNT data

The neuroscan CNT format can be converted with the function *spm\_eeg\_rdata*. Nearly all information is contained in the raw \*.cnt file. The only information that is not in there is about the reference used. SPM will ask you explicitly about the reference. However, if you don't want to re-reference your data at a later stage, you don't need to supply information about the reference electrode. The same is true, if you want to re-reference, but don't want to transform the reference channel again to an EEG channel. For example, this is the case when the reference were the earlobes.

The conversion routine can be started by calling the function *spm\_eeg\_rdata*.

### 13.4.1 Syntax

$$D = \text{spm\_eeg\_rdata}(S)$$

### Input

The input struct  $S$  is optional and has the following optional fields:

<b>Fdata</b>	-	filename of CNT-file
<b>Fchannels</b>	-	String containing name of channel template file
<b>reference</b>	-	name of reference channel. If you want to make this channel an EEG channel at a later re-referencing, you need to supply the exact name of the channel. If you just want to store the reference name (e.g. earlobes), just enter any descriptive text.

### Output

The output struct  $D$  contains the header struct of the converted file. This struct has been written to a \*.mat file of the same name as the converted file. The data has been written to a corresponding \*.dat file.

## 13.5 The MEEG SPM format

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. Note that the data should always be read using the routine *spm\_eeg\_ldata*, see section 13.6. In the following, we will first describe all single-element entries, and then all entries that are itself structs.

<b>Nchannels</b>	- The number of channels. This number also includes channels like EOG or other external channels
<b>Nevents</b>	- The number of epochs
<b>Nsamples</b>	- The number of time bins in one epoch
<b>Radc</b>	- The sampling rate measured in Hertz
<b>fnamedat</b>	- The name of the *.dat file without leading path
<b>fname</b>	- The name of the *.mat file without leading path
<b>path</b>	- The path to the directory where the *.mat and *.dat file are stored
<b>datatype</b>	- The datatype with which the data in the *.dat file is stored. Possible datatypes are 'int16' and 'float'
<b>data</b>	- This is a <code>spm_file_array</code> struct that contains the memory mapped data. For epoch data this is effectively a three-dimensional array of the dimensions <i>Nchannels</i> × <i>Nsamples</i> × <i>Nevents</i> .
<b>scale</b>	- A matrix with internally used scaling values for the memory mapping of data. For documentation, see the directory <code>file_array</code> in the SPM main directory.
<b>modality</b>	- A string that is (currently) either 'EEG' or 'MEG' and describes the type of data in the file
<b>units</b>	- A string that determines the units of the data in tex-format, e.g. $\mu\text{V}$ for micro V

### 13.5.1 channels

The substruct `channels` contains all channel-related information.

<b>ctf</b>	- The name of a channel template file (CTF) without leading path. It is assumed that the CTF is located in the EEGtemplates sub-directory of the SPM5 main directory. This file contains standard channel names for a given setup and their coordinates on a 2D plane. When converting a file to the SPM-format, a link is made to a CTF. Identification of channels in the data file is via the channel names. (See also sec. 13.7)
<b>Bad</b>	- An index vector of bad channels
<b>name</b>	- A cell vector of channel names
<b>eeg</b>	- The indices of actual EEG channels. For example, these exclude the EOG channels.
<b>order</b>	- An index vector that maps from the data order of channels to the corresponding channel in the CTF.
<b>heog</b>	- The channel index of the HEOG channel.
<b>veog</b>	- The channel index of the VEOG channel.
<b>reference</b>	- If available, this is an index of the reference channel in the order of the CTF. Otherwise this is 0.
<b>ref_name</b>	- If available, the name of the reference channel. This actually does not need to be a valid channel name, but is just used as a reminder for the user (e.g. 'earlobes').
<b>thresholded</b>	- A cell vector that contain channel indices (in data order) of epochs with data surpassing some threshold. This is usually generated by the <code>spm_eeg_artefact</code> function.

### 13.5.2 events

The substruct `events` contains information related to the epochs.

<b>code</b>	- A vector which contains event numbers for each event. These were read during the conversion from the event channel of the raw data.
<b>time</b>	- A vector which contains the timing of stimulus presentation for each event (measured in time bins). This is used for epoching the data.
<b>start</b>	- The number of time bins before onset of the stimulus
<b>stop</b>	- The number of time bins after onset of the stimulus
<b>Ntypes</b>	- The number of different event types
<b>reject</b>	- A vector which for each event a 0 or 1's indicating whether this trial was rejected or not.
<b>repl</b>	- A vector with the number of single trials which were used for each event-type by the averaging function.

### 13.5.3 filter

The substruct *filter* contains information about filtering the data. This struct is usually filled by the function *spm\_eeg\_filter*.

<b>type</b>	- The name of the used filter
<b>band</b>	- 'lowpass' or 'bandpass'
<b>PHz</b>	- The cutoff of the filter in Hertz
<b>para</b>	- A cell vector with filter parameters. See the matlab function <i>filter</i> for a description of what these parameters are.

### 13.5.4 threshold

The substruct *threshold* contains information about thresholding the data. This struct is usually filled by the function *spm\_eeg\_artefact*.

<b>External_list</b>	- Indicator (0/1) whether external information was used which trials were artefactual or clean.
<b>threshold</b>	- The threshold used in microVolt

## 13.6 Reading of data

Once the data is in SPM-format, it can be read into matlab. This should be done using the routine *spm\_eeg\_ldata*. (Note: If you only work with the GUI, you won't need to call this function.) The routine will mainly do two things. First, it will load the header struct in the \*.mat-file. Secondly, it will memory map the data in the \*.dat file to a field in this struct. The memory mapped data can be addressed like a matrix which is convenient for accessing the data in a random access way. However, a word of caution: If you write new values to the D.data-matrix, the matrix is not only changed in the matlab variable (in memory), but also physically on the hard disk.

This function can only be called via matlab command line.

### 13.6.1 Syntax

$$D = \text{spm\_eeg\_ldata}(P)$$

#### Input

The input string *P* is optional and contains the file name of the \*.mat file.

#### Output

The output struct *D* contains all header information about the data. The data are memory mapped and can be accessed as the field *data*.

## 13.7 The channel template file

The channel template file is SPM’s way of connecting acquired channel data to a spatial location. The locations of channels are typically not contained in the MEEG raw data file. A channel template file contains channel names for a given setup and their locations in some coordinate system. All channel template files are contained in the subdirectory *EEGtemplates* in the SPM5-directory. During the initial conversion of data each channel is identified by its name and mapped to the corresponding channel location contained in the channel template file. If a channel’s name is not contained in the user-specified channel template file, a warning is issued. Many warnings usually mean that the wrong channel template file for a specific setup was selected. Note that even if the mapping from channels to their locations were not identified correctly, it is still possible to perform preprocessing operations (e.g. epoching, filtering, etc.) on the converted data. However, the channels’ locations are needed for display and mapping to voxel space.

Currently, the channel template files in SPM5b all map into some standard 2D space on the scalp. This is useful for mapping multiple subjects’ data to a standard space and performing SPM analyses in 2D scalp space. Future updates of SPM5b will supply channel template files that map to a 3D sensor space, which is critical for 3D source reconstruction. This 3D space can be some standard space which might be useful for MEG data and EEG data acquired with a cap. Alternatively, one can also use digitized sensor positions as locations, e.g. acquired with a Polymus system.

### 13.7.1 Structure

A channel template file (CTF) is a mat-file that contains four variables:

- |                  |  |
|------------------|--|
| <b>Nchannels</b> | - The number of channels known to the CTF  |
| <b>Cnames</b>    | - A cell vector of channel names. Each cell can contain either a string or a cell vector of strings. The latter allows to have 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. |
| <b>Cpos</b>      | - 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 $\geq 0.95$ .   |
| <b>Rxy</b>       | - A factor that determines the width of the display plots to their height when displaying the data. Standard is 1.5.   |

### 13.7.2 Creating your own channel template file

The channel template file is important for using SPM’s full functionality for MEEG data. The channel template files contained in the *EEGtemplates* directory are the ones that we or our collaborators found useful. Other groups will need different channel template files, because they might have different setups, i.e. they use different channel names and/or different channel coordinates. Note that if a specific setup is just a subset of channels of an existing setup, the CTF of the full setup can be used.

If a new channel template file is needed, this can be simply created by saving the variables *Nchannels*, *Cnames*, *Cpos* and *Rxy* to a new channel template file. Typically this would be done by running a script that creates these four variables and saves them to a file. The creative bit is to list the actual coordinates of the channels on a 2D plane. We found two feasible ways for doing this. The first is to note that many electrode setups consist of electrodes sitting on concentric rings equidistant to other electrodes on each ring. Such a setup can be programmed as a script which places electrodes on each of these rings. A second way is that at least some producers of EEG caps provide coordinates for specific setups in 3D space. For example, have a look at [http://www.easycap.de/easycap/e/downloads/electrode\\_sites\\_coordinates.htm](http://www.easycap.de/easycap/e/downloads/electrode_sites_coordinates.htm). The projection to 2D coordinates could be done by first using Matlab’s *sph2cart* function to transform to Cartesian coordinates. This is followed by applying the subfunction *CartToFlat* of *spm\_eeg\_DrawSV* (SPM5 DipoleFit toolbox) to the Cartesian coordinates. We provided an ex-

ample script (*make\_Easycap\_montage1*) in the *man/example\_scripts/* folder of the distribution to illustrate this process.

## 13.8 Epoching the data

Epoching cuts out little chunks of data and saves them as 'single trials'. For each stimulus onset, the epoched trial starts at some user-specified pre-stimulus time and end at some post-stimulus time, e.g. from -100 to 400 milliseconds in peri-stimulus time. The epoched data is also baseline-corrected, i.e. the mean of the pre-stimulus time is subtracted from the whole trial. The resulting event codes are the same as saved in the \*.mat file. One can re-code events by supplying a vector of event codes.

The epoching routine can be started either by using the GUI (Epoching) or by calling the function *spm\_eeg\_epochs*.

### 13.8.1 Syntax

$D = \text{spm\_eeg\_epochs}(S)$

#### Input

The input struct  $S$  is optional and has the following optional fields:

<b>D</b>	- filename of MEEG mat file
<b>events</b>	- a struct containing the following fields
<b>start</b>	- pre-stimulus start of epoch[ms]
<b>stop</b>	- post-stimulus end of epoch[ms]
<b>types</b>	- vector of event types to extract
<b>Inewlist</b>	- indicate (0/1) to use new list of event codes
<b>Ec</b>	- vector of new event codes

#### Output

The output struct  $D$  contains the header struct of the epoched file. This struct has been written to a \*.mat file of the same name as the input file, but prepended with *e\_*. The data has been written to a corresponding \*.dat file.

## 13.9 Filtering the data

Continuous or epoched data can be filtered with a low- 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. SPM's filter function *spm\_eeg\_filter* uses matlab's signal processing toolbox. If you don't have this toolbox, you cannot filter your data using SPM.

The filter routine can be started either by using the GUI (Filter) or by calling the function *spm\_eeg\_filter*.

### 13.9.1 Syntax

$D = \text{spm\_eeg\_filter}(S)$

#### Input

The input struct  $S$  is optional and has the following optional fields:

<b>D</b>	- filename of MEEG mat file
<b>filter</b>	- a struct containing the following fields
<b>type</b>	- type of filter, currently must be 'butterworth'
<b>band</b>	- a string, 'lowpass' or 'bandpass'
<b>PHz</b>	- one (lowpass) or two (bandpass) cut-offs [Hz]

### Output

The output struct *D* contains the header struct of the filtered file. This struct has been written to a \*.mat file of the same name as the input file, but prepended with *f*. The data has been written to a corresponding \*.dat file.

## 13.10 Artefact detection and rejection

Some trials are likely to not only contain neuronal signals of interest, but also signal from other sources like eye movements or muscular activity. These signal components are referred to as artefacts. In SPM, we use only two simple automatic artefact detection schemes. The first is thresholding the data and the second is robust averaging. One can also choose to detect artefacts manually by visualizing each trial (see below). Another option is to use a more sophisticated artefact detection approach (implemented by some other software) and supply that information to SPM.

Thresholding the data is done in two passes. In the first pass, SPM detects all instances for which the threshold was passed by the absolute value for a channel and single trial. If a channel has more than a certain percentage of artefactual trials, it is defined as a bad channel. In a second pass the thresholding is repeated, but without taking into account bad channels. A trial for which the absolute data surpasses the threshold in some channel (excluding bad channels) is considered artefactual.

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 artefact routine can be started either by using the GUI (Artefacts) or by calling the function *spm\_eeg\_artefact*.

### 13.10.1 Syntax

$$D = \text{spm\_eeg\_artefact}(S)$$

### Input

The input struct *S* is optional and has the following optional fields:

<b>D</b>	- filename of MEEG mat file								
<b>thresholds</b>	- a struct containing the following fields								
<b>External_list</b>	- indicate (0/1) to use external artefact list								
<b>out_list</b>	- index vector of artefactual trials								
<b>in_list</b>	- index vector of clean trials								
<b>Check_Threshold</b>	- indicate (0/1) whether to threshold channels								
<b>threshold</b>	- threshold to use: can be either a scalar which is the threshold for all channels, or a vector of channel-wise thresholds								
<b>artefact</b>	<table> <tr> <td><b>in_list</b></td><td>- index vector of clean trials</td></tr> <tr> <td><b>weighted</b></td><td>- a struct containing the following fields</td></tr> <tr> <td><b>wtrials</b></td><td>- indicate (0/1) whether to use robust averaging</td></tr> <tr> <td></td><td>- indicate (0/1) whether to use robust averaging across trials</td></tr> </table>	<b>in_list</b>	- index vector of clean trials	<b>weighted</b>	- a struct containing the following fields	<b>wtrials</b>	- indicate (0/1) whether to use robust averaging		- indicate (0/1) whether to use robust averaging across trials
<b>in_list</b>	- index vector of clean trials								
<b>weighted</b>	- a struct containing the following fields								
<b>wtrials</b>	- indicate (0/1) whether to use robust averaging								
	- indicate (0/1) whether to use robust averaging across trials								

### Output

The output struct  $D$  contains the header struct of the artefact-detected file. This struct has been written to a \*.mat file of the same name as the input file, but prepended with  $a$ . The data has been written to a corresponding \*.dat file.

## 13.11 Downsampling

The data can be downsampled to any sample rate. This is useful if the data was acquired at a higher sampling rate than one needs for making inferences about low-frequency components. SPM's downsampling routine uses the matlab function *resample*, which is part of matlab's signal processing toolbox. If you don't have this toolbox, you cannot downsample your data using SPM.

The downsampling routine can be started either by using the GUI (Other/downsample) or by calling the function *spm\_eeg\_downsample*.

### 13.11.1 Syntax

$D = \text{spm\_eeg\_downsample}(S)$

### Input

The input struct  $S$  is optional and has the following optional fields:

- |                 |                                  |
|-----------------|----------------------------------|
| <b>D</b>        | - filename of MEEG mat file      |
| <b>Radc_new</b> | - the new sampling rate in Hertz |

### Output

The output struct  $D$  contains the header struct of the downsampled file. This struct has been written to a \*.mat file of the same name as the input file, but prepended with  $d$ . The data has been written to a corresponding \*.dat file.

## 13.12 Rereferencing

When you acquired data to a certain reference, you can simply re-reference the data to another channel or to the average over a set of channels. Bad channels are excluded from an average reference. If there was only a single reference channel before, one can add it again to the data. The rereferencing routine displays the indices of all channels of the data as a help to decide which indices to select as a new reference.

The rereferencing routine can be started either by using the GUI (Other/rerefence) or by calling the function *spm\_eeg\_rerefence*.

### 13.12.1 Syntax

$D = \text{spm\_eeg\_rerefence}(S)$

### Input

The input struct  $S$  is optional and has the following optional fields:

- |               |  |
|---------------|--|
| <b>D</b>      | - filename of MEEG mat file                |
| <b>newref</b> | - a struct containing the following fields |

### Output

The output struct  $D$  contains the header struct of the rereferenced file. This struct has been written to a \*.mat file of the same name as the input file, but prepended with  $R$ . The data has been written to a corresponding \*.dat file.

## 13.13 Grand mean

The grand mean is usually understood as the average of ERPs 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. The averaged file will be written into the same directory as the first selected file.

The grand mean routine can be started either by using the GUI (Other/grand mean) or by calling the function *spm\_eeg\_grandmean*.

### 13.13.1 Syntax

$D = \text{spm\_eeg\_grandmean}(S)$

#### Input

The input struct  $S$  is optional and has the following optional fields:

**P** - filenames of M/EEG mat files (char matrix)

#### Output

The output struct  $D$  contains the header struct of the averaged file. This struct has been written to a \*.mat file of the same name as the input file, but prepended with *g*. The data has been written to a corresponding \*.dat file.

## 13.14 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. 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 merge routine can be started either by using the GUI (Other/merge) or by calling the function *spm\_eeg\_merge*.

### 13.14.1 Syntax

$D = \text{spm\_eeg\_merge}(S)$

#### Input

The input struct  $S$  is optional and has the following optional fields:

**P** - filenames of MEEG mat files (char matrix)

#### Output

The output struct  $D$  contains the header struct of the merged file. This struct has been written to a \*.mat file of the same name as the input file, but prepended with *c*. The data has been written to a corresponding \*.dat file.

## 13.15 Time-frequency decomposition

The time-frequency decomposition is performed by using a continuous Morlet wavelet transform. The result is written as two result files, one contains the instantaneous power and the other the phase estimates. One can select the channels and frequencies for which power and phase should be estimated. Optionally, one can apply a baseline correction to the power estimates, i.e. the mean power of the pre-stimulus time is subtracted from the power estimates.

The time-frequency decomposition routine can be started either by using the GUI (Other/time-frequency) or by calling the function *spm\_eeg\_tf*.

### 13.15.1 Syntax

$D = \text{spm\_eeg\_tf}(S)$

#### Input

The input struct  $S$  is optional and has the following optional fields:

<b>D</b>	- filename of MEEG mat file
<b>frequencies</b>	- vector of frequencies [Hertz] at which decomposition is performed
<b>rm_baseline</b>	- indicate (0/1) whether baseline should be subtracted
<b>Sbaseline</b>	- start and stop of baseline (in time bins)
<b>channels</b>	- indices of channels for which to perform time-frequency decomposition
<b>Mfactor</b>	- the so called Morlet wavelet factor, defaults to 7.

#### Output

The output struct  $D$  contains the header struct of the phase information. This struct has been written to a \*.mat file of the same name as the input file, but prepended with *t2*\_. The data has been written to a corresponding \*.dat file. The power data has been written to a file prepended with *t1*\_.

## 13.16 Averaging

Averaging of the single trial data is the crucial step to obtain the ERP. By default, when averaging single trial data, single trials are averaged within trial type. Power data of single trials (see sec. 13.15) can also be averaged.

The averaging routine can be started either by using the GUI (average) or by calling the function *spm\_eeg\_average*.

### 13.16.1 Syntax

$D = \text{spm\_eeg\_average}(S)$

#### Input

The input struct  $S$  is optional and has the following optional fields:

<b>D</b>	- filename of MEEG mat file
----------	-----------------------------

## 13.17 Linear combinations of epochs

As an extension to the averaging functionality, SPM can also be used to compute linear combinations of single trials or epochs. For example, you might be interested in computing the difference between two ERPs. This can be done by calling the function *spm\_eeg\_weight\_epochs*.

### 13.17.1 Syntax

$D = \text{spm\_eeg\_weight\_epochs}(S)$

**Input**

The input struct  $S$  is optional and has the following optional fields:

- D** - filename of MEEG mat file
- c** - a weight (contrast) matrix with dimensions  $N_{contrasts} \times N_{epochs}$ . Each row of  $c$  contains one contrast vector. For a simple difference between two ERPs use  $[-1 1]$ .

**Output**

The output struct  $D$  contains the header struct of the averaged file. This struct has been written to a \*.mat file of the same name as the input file, but prepended with  $m$ . The data has been written to a corresponding \*.dat file.

## 13.18 Mixing of channels

SPM can also be used to compute the mixing of channels by a square matrix. For example, we found this useful for computing a weighting of the data with an independent component analysis (ICA) mixing matrix. You can do this by calling the function *spm\_eeg\_weight\_channels*.

### 13.18.1 Syntax

$D = \text{spm\_eeg\_weight\_channels}(S)$

**Input**

The input struct  $S$  is optional and has the following optional fields:

- D** - filename of MEEG mat file
- W** - a mixing matrix with dimensions  $N_{channels} \times N_{channels}$ .  
Hint: If you call the function without arguments, prepare a variable that contains this matrix.

**Output**

The output struct  $D$  contains the header struct of the averaged file. This struct has been written to a \*.mat file of the same name as the input file, but prepended with  $w$ . The data has been written to a corresponding \*.dat file.

## 13.19 Weighting of the time-series

You can use SPM to multiply your data, over peri-stimulus time, by some weighting function. For example, we found this useful for removing stimulus-related artefacts due to an electrical impulse at peri-stimulus time 0. The weighting would be a function over peri-stimulus time consisting of 1s everywhere, except for time 0, where you would remove data by putting in a 0. You can do this by calling the function *spm\_eeg\_weight\_time*.

### 13.19.1 Syntax

$D = \text{spm\_eeg\_weight\_time}(S)$

**Input**

The input struct  $S$  is optional and has the following optional fields:

- D** - filename of MEEG mat file  
**weight** - a weighting function (vector) with length peri-stimulus time points. Hint: If you call the function without arguments, prepare a variable that contains this vector.

### Output

The output struct *D* contains the header struct of the averaged file. This struct has been written to a \*.mat file of the same name as the input file, but prepended with *w*. The data has been written to a corresponding \*.dat file.

## 13.20 Displaying data

SPM can be used to display epoched data. The viewer is called by choosing the EEG/MEG modality and clicking on the *M/EEG* entry in the *Display* menu. After selecting an epoched M/EEG file in SPM format, the viewer displays all channels of the first trial or trial type. The position of the channels is taken from the channel template file (see above).

Navigation through trials or trial types is either by the trial slider or by the trial listbox. The scaling of the displayed data (for EEG:  $\mu$ Volt, for MEG: in  $10^2$  femto Tesla) can be changed by using the scaling slider. Up to four trials or trial types can be plotted at the same time by using the shift or ctrl-button while selecting files with the left mouse button in the listbox.

Single trials can be classified as either artefactual or clean by pressing the *Reject* button. This information can be saved to the \*.mat file by pressing the *Save* button.

A left-clicking on a single channel plot will plot the time-series of this channel in much more detail in a pop-up figure. Another left-click on the (small) channel plot will close the pop-up figure again.

The topography at a specific time point can be displayed either in 2D or in 3D by clicking on the *Topography* button and selecting a peri-stimulus time and choosing between 2D/3D. In this display, bad channels will not be interpolated, but no data is plotted in the location of the bad channel.

The set of displayed channels can be changed by clicking on the *channel* button. This is useful for (i) faster plotting of single trial data by choosing less channels to display and (ii) having larger plots in the display tool. In the channel select tool, you can click on the channel to select or deselect this channel. Alternatively, you can also use the listbox with shift/ctrl to select or deselect channels. Channel selection can be saved and loaded to mat-files. Pressing *Ok* will confirm your selected channels and update your display.



# Chapter 14

## Spatial projections

After pre-processing the M/EEG data, the data is projected to voxel space. This is a critical departure from most standard analyses for ERP data which typically analyse data in channel space. The main motivation to project data to voxel space is our overall goal to integrate M/EEG with other modalities like functional magnetic resonance imaging (fMRI). A minimum requirement for integration is that both kinds of data are in the same space. Naturally, this space is the 3D-brain space, because it is here where neuronal sources are active and cause observed M/EEG activity in the sensors.

Projection to brain space, i.e. source reconstruction, is usually done using one of two approaches. The first is based on the assumption that the potential fields measured on the scalp can be explained by a few equivalent current dipoles (ECD). Although this is a very parsimonious model and widely used in the M/EEG community, this is not the approach we are going to use for mass-univariate analysis in SPM. Rather we use a second approach, which models the observed sensor data by hundreds to thousands of dipoles in brain space. Such a model has many more parameters than the sparse dipole model, but parameters can be estimated efficiently using Bayesian techniques with informed functional and anatomical priors. The solutions from these distributed models lead to reconstructed sources that can be extended and don't need to be focal like in the ECD approach.

However, note that we do not exclude ECD models from the SPM software. Although we won't use ECD models for the mass-univariate analysis, we are actively pursuing dynamical causal modelling based on spatial forward models that use ECD representations (see chapter ??). Furthermore, there is a toolbox available for ECD reconstruction using classical techniques (see below). One can use this toolbox to derive dipolar solutions for multiple subjects and analyse the reconstructed source activity outside of SPM (i.e. the mass-univariate approach).

In cases when one is not interested in source reconstruction, one can instead project to the 2D scalp surface. This procedure basically leads to a conventional sensor-based analysis.

### 14.1 Distributed linear models

### 14.2 Equivalent current dipoles

### 14.3 Interpolation on scalp

The interpolation of sensor data in 2D voxel space is mostly equivalent to a sensor-based analysis. The critical difference to conventional analysis approaches lies hidden in the treatment of the factor *space*. In the ERP community, the factor space is traditionally considered a factor with only a couple of levels. Each level is usually an average over a region on the scalp, e.g. left frontal electrodes. This allows to test for specific contrasts in a step-down fashion and is appropriate for setups that measure up to ca. 32 channels. For example, one first establishes overall significance for some global F-test and then tests differences or interactions. While this procedure has its established place in the literature, it has the disadvantage that one has to specify some contrast over channels before testing. With high-density acquisitions, it is not obvious which of the many

possible averages will lead to the most sensitive test. An alternative is to analyse the data in a mass-univariate fashion over space and use GFT to assess significance, adjusted for multiple comparisons, of topological features like maxima. This approach has the advantage that one doesn't have to specify expected locations of activations prior to the test. This principle was recently demonstrated by Kilner et al. (2005) for EEG power data.

Note that when using this mass-univariate approach, it is not possible to test for region  $\times$  condition interactions. By this we mean that one tests for differences between voxels. For this one needs multivariate models that take correlations over space directly into account. In conventional ERP analyses testing for region  $\times$  condition interactions is a standard procedure. So why can't you do this in SPM? Although this issue sounds like a technical subtlety, the main assumption behind such a test are vulnerable: When comparing effects between voxels, one effectively assumes that the underlying causes in different regions have the same effect on the scalp. This is not necessarily true.

# Chapter 15

## 3D source reconstruction: Imaging approach

Here is a brief help to the 3D reconstruction based on the Imaging approach. In the near future, this will be improved by including more theoretical details upon the different procedures as well as a practical tutorial that will guide the user through the SPM interface via the analysis of a sample dataset.

### 15.1 Introduction

This chapter focuses on the imaging (or distributed) method for doing EEG/MEG source reconstruction in SPM. Such an approach to spatial projection onto (3D) brain space consists in considering a large amount of dipolar sources all 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 ...), 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 enables mass-univariate analysis in SPM (see chapter...).

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 more than a decade now [18] and the proposed pipeline in SPM for '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 various nature; data-driven relevance estimation being made possible through Bayesian model comparison [29, 58, 49, 25].
- 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. In this chapter, we go through each of those steps that all need to be completed when proceeding with a full inverse analysis:

1. Source space modeling,

2. Data co-registration,
3. Forward computation,
4. Inverse reconstruction.

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

Everything which is described hereafter is a new feature in SPM and is accessible from SPM5 user-interface by choosing the 'EEG/MEG' application, '3D source reconstruction' and 'Imaging'.

## 15.2 Data structure

The Matlab structure 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.

## 15.3 Source space modeling (*mesh*)

The individual cortical mesh is obtained from a template mesh. Four Mesh sizes are available (3004, 4004, 5004 and 7204 vertices). If not yet obtained, the spatial normalization of the subject's T1 MRI into MNI space is performed (see *spm\_preproc.m* based on tissue probability maps). The inverse of that transformation is computed and applied to the template mesh to furnish the individual cortical mesh.

Individual meshes for the inner-skull and scalp surfaces are also computed from the individual T1 MRI. They are obtained by performing a binary mask of the the volumes delimited by the inner-skull and scalp surface respectively. Then, using an initial spherical mesh, a realistic-shaped mesh is obtained for each of the two tissues and further regularized via an erosion and growing procedure.

The meshing module includes the following functions:

- *spm\_eeg\_inv\_mesh\_ui.m*: run the user interface for this module,
- *spm\_eeg\_inv\_spatnorm.m*: normalize the T1 image if needed,
- *spm\_eeg\_inv\_meshing.m*: main function to produce Cortex, Inner-skull and Scalp meshes,
- *spm\_eeg\_inv\_getmasks.m*: produce masks of Inner-skull and Scalp,
- *spm\_eeg\_inv\_ErodeGrow.m*: erosion and growing procedure,
- *spm\_eeg\_inv\_getmeshes.m*: obtains the inner-skull and scalp meshes from correpsonding binary masks,

- *spm\_eeg\_inv\_CtrBin.m*
- *spm\_eeg\_inv\_TesBin.m*
- *spm\_eeg\_inv\_ElastM.m*
- *spm\_eeg\_inv\_checkmeshes.m*: displays the computed three meshes in the SPM main figure

## 15.4 Data Registration (*datareg*)

There are two possible ways of coregistering 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 tesselation). For EEG, the sensor locations can be used instead of the headshape. For MEG, the headshape is first coregistered into sMRI space; the same transformation is then applied to the sensors.

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.

The data-registration module includes the following functions:

- *spm\_eeg\_inv\_datareg\_ui.m*: run the user interface for this module,
- *spm\_eeg\_inv\_datareg.m*: main co-registration function,
- *spm\_eeg\_inv\_checkdatareg.m*: display meshes, sensor locations and fiducials in native MRI space to enable one checking the co-registration by eye.

## 15.5 Forward computation (*forward*)

Several methods are proposed, depending on the modality (EEG or MEG). All these approaches/functions are identical to the one initially developed and provided by the BrainSTorm package (Matlab open-source and free software: <http://neuroimage.usc.edu/brainstorm/>).

For EEG [21]:

1. single sphere (scalp surface),
2. three spheres (inner, outer skull and scalp surfaces),
3. three spheres (+ Berg correction),
4. overlapping spheres (one fitted sphere per sensor).

For MEG [36]:

1. single sphere,
2. overlapping spheres

The forward module includes the following functions:

1. *spm\_eeg\_inv\_forward\_ui.m*: run the user interface for this module,
2. *spm\_eeg\_inv\_BSTcreatefiles.m*: create the structure and required files and parameters to interface SPM and BrainSTorm,
3. *spm\_eeg\_inv\_BSTfwdsol.m*: compute the BrainSTorm forward solution, calling function *bst\_headmodeler.m*,
4. *spm\_eeg\_inv\_PCAgain*: compute the svd of the gain matrix.

## 15.6 Inverse reconstruction (*inverse*)

The reconstruction is 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.

The inverse module includes the following functions:

- *spm\_eeg\_inv\_inverse\_ui.m*: run the user interface for this module,
- *spm\_eeg\_inv\_inverse.m*: main function,
- *spm\_eeg\_inv\_evoked.m*: compute the evoked response,
- *spm\_eeg\_inv\_induced.m*: compute the evoked and/or induced power,
- *spm\_eeg\_inv\_msp.m*: Multivariate Source Prelocalisation [48].

# Chapter 16

# M/EEG modelling and statistics

After projection to 2D- or a 3D-space (source reconstruction), the data is in voxel-space and ready to be analysed. There are several ways how one can proceed. In this chapter, we will focus on analyzing epoched time-series data. These can be event-related responses (ERPs), event-related fields (ERFs) or single trials (M/EEG).

In the following, we will go through the various stages of modelling using typical examples to illustrate the procedures.

## 16.1 Preliminary remarks

All analyses can be done using either the graphical user interface (GUI) or a batch system (i.e. using scripts in the SPM2-fashion, s. below). The GUI has the advantage that one doesn't need matlab-knowledge to analyse data. The batch system has the advantage that it is a fast and efficient way of entering model and data. Its disadvantage is that some Matlab-Knowledge is required. However, with this distribution, we provide some template scripts to analyse (typical) data in batch mode. We assume that with slight modifications these scripts can be used for most analyses.

## 16.2 How epoched time-series are analysed in SPM

After preprocessing the data (i.e. epoching, filtering, etc...) and projection to voxel-space, we have discretely sampled versions of continuous fields [43]. These data can be analysed with a mass-univariate approach using results from Random Field theory (RFT) to adjust p-values for multiple comparisons [66]. The model used at each voxel is a general linear model [44]. Typically one wants to analyse multiple subjects' data acquired under multiple conditions. Given that each evoked-response has up to hundreds of time points, this is an awful lot of data at each voxel. The ideal way to analyse these data would be to specify a single hierarchical model (1st level: within-subject, 2nd level: over subjects) and estimate its parameters. However, this is computationally not feasible because of the length of the data vector at each voxel. Fortunately, such a 2-level model can usually be split up into two models: The 1st level and the 2nd level model. The input data to the 2nd model are contrasts of the 1st level model [44]. In all cases considered in this chapter, this 2-stage procedure gives exactly the same results as the 2-level model. The reason for this is that we are not really *modelling* the data at the 1st level, but simply forming weighted sums of the data, over time. For example, if we are interested in the N170 component, one could average the data from 150 to 190 milliseconds. This is exactly the approach used in conventional ERP analysis. This approach is not a model, because simply taking sums corresponds to using an identity matrix as design matrix. This procedure leaves no degrees of freedom for error estimation.

In summary, the SPM-approach is to form, at each voxel, weighted sums of the data, over time, at the 1st level. We refer to these weighted sums as contrast images. These form the input to the 2nd level, where one usually tests for differences between conditions or between groups (s. below). The second level models are usually the same as the ones one would use for functional

magnetic resonance imaging (fMRI). Importantly, these 2nd level models have enough degrees of freedom to estimate the error, i.e. statistics can be computed.

The output of such a 2nd level analysis is a voxel-volume (or map), where each voxel contains one statistical value. The associated p-value is adjusted for multiple comparisons [66]. This adjustment is important, because there are many other voxels or channels. One (disadvantageous) alternative to adjustment is to consider only pre-selected channels or averages over channels. This is why the adjustment is especially important for high-density measurements, because there are many channels to select from. We believe that it is generally too subjective to select channels for analysis a-priori. We see the GFT-adjustment as a good way of looking at the whole data without any prior selection. This has been already demonstrated for EEG data (in another context) by [46].

### 16.3 1st level

At the 1st level, we select periods or time points in peri-stimulus time that we would like to analyse. Critically, this choice must be made a-priori by you. The alternative would be to not treat peri-stimulus time as a factor, but as a dimension of a Random Field approach. This alternative approach is often used in time-frequency analysis of induced and evoked oscillations, where it seems sometimes difficult to specify areas of interest on the time-frequency plane a-priori [46].

In the present approach, time is a factor, and you have to form weighted-sums over peri-stimulus time to provide input to the 2nd level. Of course, you don't need to constrain yourself to a single contrast around a specific peri-stimulus time, but you can compute as many as you like. For example, to analyse multiple aspects of an ERP, it is not uncommon to form averages around several time-points of an ERP. At the 2nd level, these can be either analysed independently or within one model to make inferences about interactions between conditions and peri-stimulus time.

In the following, we will go through model specification and computation of contrast images. This guide is not written as a tutorial (i.e. detailed instructions for a given data set), but describes each design option and hopefully provides deeper background knowledge.

#### 16.3.1 The aim

The aim of the 1st level is to compute contrast images that provide the input to the 2nd level. We will describe this using the example of 2D-data, i.e. data that has not been source reconstructed but, for each peri-stimulus time point, has been projected to a 2D-plane (s. chapter 14).

#### 16.3.2 Start

Start SPM by the command '*spm eeg*' from the matlab command line. Press the *EEG/MEG* button. Your first choice is to either specify the model design or the input data. One always starts with the design. Currently, there are two design options: (i) *all options* and (ii) *ERP/ERF*. The latter option is a shortcut to quickly input an evoked responses study. We will first describe *all options* and then treat the *ERP/ERF* option as a special case.

#### 16.3.3 All options

You first have to answer the question whether this is a 1st level design. This determines whether SPM expects to model peri-stimulus time as a factor. Also, if one models first-level data, SPM will ask next for **one** M/EEG-matfile before the data was projected to voxel-space. The reason for this is that the voxel-images lost important information during the conversion. For example, all timing information were lost. With the nifti-images only, SPM doesn't know the peri-stimulus time of each data point. However, this information is critical as soon as you try to specify (later on) linear weights in terms of peri-stimulus time. So, when you select an M/EEG file, SPM will read timing information from this file. For an ERP-study, the M/EEG-file of the average (ERP) is a good choice.

### 16.3.4 How many factors?

This question starts off the design specification proper. SPM needs to know the number of factors which you want to model. At the 1st level, there are typically only factors *peri-stimulus time* and *condition*. If you like, you can further subdivide the condition-factor in its components. For instance, if you have a 2x2 factorial design, you may want to specify 3 factors: *factor1*, *factor2* and *peri-stimulus time*.

### 16.3.5 Factor names and # of levels for factors

For each factor, you now input its name, e.g. condition, and enter the number of levels. For instance, if you have 2 conditions, you enter 2. For peri-stimulus time, you enter the number of time points in your evoked responses. Important: You should call the peri-stimulus time factor 'time'. For the number of levels for this special factor, SPM defaults to the correct number of peri-stimulus time points. (Note that it is currently not possible to model only a subset of time points.)

### 16.3.6 Select design component

You have the choice between *Identity* and *Constant*. Your selected design components are combined (by Kronecker tensor product) to form the 1st level design matrix. This has also been described in [44]. For the 1st level, you simply choose for all factors *identity*. This completes model specification.

### 16.3.7 Data

For selecting data, press the *EEG/MEG* button again. After selecting the *SPM.mat* file, you are asked to select data for each factor. The order in which you input data depends on the order of how you named the individual factors. We recommend that you make the *peri-stimulus time* factor the last factor. After projection to voxel-space, the data are stored as 4-dimensional files with the third dimension  $z = 1$ . If you want to input all peri-stimulus time points for a given file, you have to select all volumes along the 4th dimension. This is done by setting the number '1' in the SPM-file selector (below the 'Filt' line) to '1:101', where '101' is the total number of peri-stimulus time points. Of course, you have to replace '101' by the number of time points of your data (or by any natural number bigger than that). This choice will make all time points selectable. Then right-click over the file names and *Select all*. Press *done* to confirm your choice. This completes data selection.

### 16.3.8 'Estimation'

Although there is actually nothing to estimate, clicking the *Estimation* button will prepare some internal structure for the results section. We kept this (otherwise redundant) estimation step to provide for greater similarity with other analyses using SPM.

### 16.3.9 Results

After clicking on *Results*, choose the appropriate *SPM* and the contrast manager will pop up. In contrast to a usual SPM study, we don't use the contrast manager to compute statistics, but contrasts only!

Click *Define new contrast...* and enter a name for your contrast. Then note a (new) button called *components* which is only visible for M/EEG models. Clicking this button opens the contrast components manager. This is simply a tool that exploits the knowledge about the factors which you have specified earlier. Knowing the factors and their levels makes it easy to split up a (long) contrast weight vector into a few components. For each contrast weight vector, each factor contributes one component. By using the Kronecker tensor product, these components can be combined into the resulting contrast weight vector. This is not only time-saving, but many people tend to find this approach more intuitive than the usual approach of figuring out the contrasts yourself. For instance, if you have specified two conditions, you might be interested in

their difference. Enter a  $[-1 \ 1]$  as contrast component. For the *time* factor, instead of entering one number for each time point, better click on the *Generate* button. Click on the 'Time' button and specify a rectangular averaging window by providing the start and end of this window (in milliseconds). Press *Compute*. You can see now in the contrast manager window that your contrast weights have been computed and are displayed above the identity (design) matrix. You can also specify the contrast weights as usual in the contrast box, but this would require to enter several hundreds to thousands of numbers. Press *ok* to proceed and compute the contrast.

### 16.3.10 Display

You can display the resulting contrast image by using the *Display* button.

### 16.3.11 ERP/ERF

You can shortcut some of the question and especially the data selection by choosing the *ERP/ERF* option (instead of *all options* when specifying a design. This option assumes that you have two factors, *condition* and *time*. There are less questions during design specification. When selecting data, you don't need to select all time point, but only the first! SPM will assume that you want to select all time points of the selected file. Using this option will otherwise result in the same model as described above.

### 16.3.12 Multiple subjects

For each of your subjects, you perform these operations in a separate 1st-level analysis. For each subject, you want to compute the same contrasts and use them as input to a model, where *subjects* is the repetition factor.

## 16.4 2nd level models

For 2nd level modelling, you can use different ways to specify a model. There is *Basic models* which was primarily developed for PET/fMRI but is equally appropriate for EEG/MEG data. These are suited best when the model is simple (like a 1-sample or 2-sample t-test). In our experience, most EEG/MEG models fall into this category of simple models. If models are more complicated, like, e.g., two groups with multiple subjects/conditions, we recommend using the *EEG/MEG* models.

### 16.4.1 All options

As above, go for *All options*. This time, press 'no' for the question 'Is this a first-level design'.

### 16.4.2 Factors

This includes all factors, even repetition factors. For example, at the 2nd level a 2x2 factorial design has 3 factors: *subject*, *factor1* and *factor2*.

### 16.4.3 Design partitions and design components

The way this modelling device constructs a design matrix is by using the Kronecker tensor product on the hierarchy of specified design components. However, some/many designs can't be constructed in this way. For example, the design matrix of a paired two sample-test consists of two merged partitions, each of which is a Kronecker tensor product of design components. For each partition, the factors and the levels are the same. The difference is in the choice of the design components for each factor under each partition. For example, for a paired two-sample-test, one has 2 factors (subjects and conditions) and 2 design partitions. For the 1st partition, choose *Constant* for subjects and *Identity* for conditions. For the 2nd partition, it's the other way around, i.e. *Identity* for subjects and *Constant* for conditions.

#### 16.4.4 Covariance components

Specification of the covariance components determines the error model [30]. For each factor, there are two questions: (i) Identical variance for factor *xxx*, and (ii) Independence for factor *xxx*. SPM constructs all the variance components from your answers. The first question pertains to the assumption whether each level of this factor has identical variance. The second questions asks whether the different levels for a given factor are correlated. Some examples: For a repetition factor like subjects, you should always answer both questions with yes. For a group factor, one would assume that the levels of this factor (the groups) have unequal variance structures, but are uncorrelated (i.e., (i) no, (ii) yes). For a condition factor, the choice is up to you. A very restrained model would follow from using (i) yes (ii) yes, whereas the most liberal model is given by (i) no (ii) no.

#### 16.4.5 Data

For each combination of factors, SPM asks you for the filenames of the data. Sometimes, this process can be more convient for you, when you have specified the factors in a specific order. For example, if you have two factors *subjects* and *condition*, the order (i) subjects, (ii) condition will ask for all images for each subject. This is convienient if you have stored the contrast images in their individual subject folder. This is the case, if you have computed 1st level contrasts following the approach described above. However, if, in an intermediate step, you have saved contrasts in condition-specific folders, the alternative order ((i) condition, (ii) subjects) is more appropriate.

#### 16.4.6 Estimation and Results

The estimation follows the usual scheme, i.e. for a classical estimation procedure we use exactly the same routine as for PET/fMRI data (i.e. maximum-likelihood estimators for the parameters and Restricted Maximum Likelihood for estimation of the variance parameters).

For specification of contrasts, you have the option to specify contrasts component-wise. This can be useful for complex designs, when it's no longer easy to work out the interaction contrasts.

For 2D data the statistical map is displayed instead of the usual glass brain. You can invoke all the usual functions that are also available for fMRI/PET data. An additional option is *channels* which let you visualise to which voxel each channel maps. You can select this option by right-clicking the button on the statistical map background. SPM asks you then for one of the original M/EEG-mat files to read the channel mapping.



# Chapter 17

## Equivalent current Dipole fitting

This little chapter demonstrates how to use the ECD (Equivalent Current Dipole) routines with the multimodal dataset available on the FIL website. The aim is to fit a single dipole on the N170 wave visible in the 3 conditions. I will briefly describe how to analyse the dataset. For more details about the implementation, please refer to the help bit of and comments in the routines themselves.

### 17.1 Necessary data

Before proceeding any further, we have to make sure that we have all the necessary data in the right format. We need

- the *amri.img/hdr* structural MRI of the subject. It will be used to build the head model and display the results in the subject's anatomical space.
- the *mae\_eeg.dat/mat* EEG data files. These are the fully processed data with one ERP per condition.
- the coordinates of the sensors, fiducial markers and scalp points (headshape) in 3 distinct *\*.mat* files.

In the dataset provided on the web, the raw *\*.pol* files are available. It is necessary to prepare these files to use them with the source reconstruction routines. This is a crucial step as the registration between the "EEG space" and "patient/image space" relies entirely on these files! To prepare these files, use the little script *create\_fid\_files.m* distributed with SPM5. A copy is also available at the end of this chapter.

Once we have all the files ready, we can proceed with the 3 main steps: building the model, fitting the dipole and displaying the results. To launch the GUI, press "3D source reconstruction" in the main window of SPM.

### 17.2 Model building

After selecting the data file *mae\_eeg.mat* and the method "ECD", the first step is building the meshes for the scalp and inner skull volume. This is done automatically through the "Meshes" button. Select the structural MRI to use (*amri.img* here) and wait...

This step takes some time as the MRI is normalised and segmented. The normalisation parameters are saved in the *amri\_vbm\_inv\_sn.mat* file and will be used later to map coordinates between the template and subject spaces. With the segmentation, the brain and scalp binary volumes are built (*amri\_iskull.img* and *amri\_oscaldp.img*). These are used to build the outer scalp and inner skull surface meshes. These are saved in the *model\_head\_amri.mat* file with other information. The scalp mesh is also saved in the file *amri\_scVert.mat*.

Once the head model is ready, we can co-register the EEG space with subject/image space. Use the "Data Reg." button and decide if the registration should be based on the fiducials only (which is quite approximate) or the fiducials and the scalp surface (which should be more precise). Then select the appropriate files: *fid\_eeg.mat*, *fid\_MRI.mat*, *headshape\_orig.mat*, *amri\_scVert.mat* and let the routine work.

To prepare the model for the forward solution, simply press "ForwardComp." and "individual" to use the subject's own MRI. The forward model uses a spherical approximation. The best fitting sphere are adjusted on the scalp surface and 2 other spheres are added to model the scalp and skull outer surfaces. Obviously the head is not spherical and there will be a mismatch between the scalp/brain surfaces and their respective spheres. We have used the idea proposed by Spinelli et al., 2000 [60], where the brain volume is warped into a sphere. This allows us to use an analytical formula to calculate the forward solution for each dipole location while preserving some anatomical characteristics: superficial (resp. deep) sources remain superficial (resp. deep) in the spherical head model.

At this last step, the electrodes are also introduced in the head model and positioned relative to the subject head, as in the MRI. The *model\_head\_amri.m* at contains the information about the fitted spheres and electrodes. Dipole fitting of the data is now possible.

### 17.3 Dipole fitting

By pressing the "Inverse Sol." you launch the dipole fitting procedure. A number of questions have to be answered in order to specify the kind of solution you want:

- "Condition to use", select which condition is used to fit the dipole(s). So far, it is not possible to fit multiple conditions (or linear combinations of them) at the same time. For example, for differences between conditions, you should pre-calculate this difference before trying to fit ECDs.
- "Time window", define the time window in ms on which the ECDs should be fitted. With the N170 demo data, a good window is 150 to 180.
- "Number of dipoles", this is the crucial question. How many dipoles should be used? It's up to you to decide... With the demo data, from the look of the EEG scalp map, 1 ECD should be enough.
- "Number of random seeds". In order to avoid being trapped in a local minimum during the optimisation process because of a peculiar starting point. The algorithm can be launched from multiple random starting 'seeds'. If they all converge to approximately the same solution, then we'll have most surely reached the local optimum.
- "Orientation of the dipoles". The location of the ECD will be constant throughout the time window but its orientation can be left free or be fixed as well. Leaving the orientation free allows the dipoles to rotate over time. To fix the orientation, we can use the (weighted according to the EEG power)) mean over the time window or use the orientation of the ECD fitting the time instant with maximum EEG power.
- "File name". File names are suggested but feel free to change it!

After fitting the N random seeds, the routine tries to group them in clusters of similar ECDs according to their location and signal variance explained. Eventually, these 'grouped' ECDs are displayed on the subject anatomy. The result of this clustering is saved in a mat file starting by *res\_* and finishing with the name you entered.

## 17.4 Result display

Results can be redisplayed with the routine *spm\_eeg\_inv\_ecd\_DrawDip.m*. The routine asks you to select the solution file you want to display and the MR image to be used.

## 17.5 Preparing the \*.pol files

```
fid_eeg = ([-0.0587687 6.79448 -0.00636311 ; ... 0.0352661 -6.78906 -0.00369206 ; ...
9.3675 0.0260009 0.00481311] + ... [-0.0328487 6.78991 0.00636288 ; ... 0.0563513
-6.79533 0.00369206 ; ... 9.45206 -0.0260009 -0.00481297])/2 ... * 10 ;
fid_mri = [-71.8 3.5 -58.8 ; ... 71.3 -6 -62.5 ; ... 0 90.6 -28.4] ;
sensors = load('sensors_noFid.pol','ASCII')*10; headshape = load('headshape_noFid.pol','-
ASCII')*10;
Rot = spm_matrix([0 0 0 0 0 -pi/2]); Rot = Rot(1:3,1:3); fid_eeg = (Rot*fid_eeg)';
sensors = (Rot*sensors)'; headshape = (Rot*headshape)';
save fid_eeg fid_eeg save fid_mri fid_mri save sensors_orig sensors save headshape_orig
headshape
```



## Chapter 18

# Dynamic Causal Modelling for evoked responses

Dynamic Causal Modelling for ERP/ERFs is described in [19] and [42], see also <http://www.fil.ion.ucl.ac.uk/spm/doc/biblio/>. We recommend reading these two communications as a starter, because it will enable you to better understand all the modelling options presented in this chapter.

In summary, the goal of DCM is to explain evoked responses as the output of an interacting network consisting of a few areas that receive an input stimulus. The difference between two evoked responses that receive comparable stimuli under two conditions is modelled as a modulation of some of the inter-areal connections [19]. This interpretation of the ERP makes hypotheses about connectivity directly testable. For example, one can ask, whether the difference between two evoked responses can be explained by some top-down modulation of early areas (Garrido et al., in preparation). Additionally, because DCM is framed in a Bayesian way, one can also compute model evidences. These can be used to compare alternative, equally plausible, models and decide which model is the better one [56].

DCM for ERP/ERFs takes the spatial forward model into account. To do this, we parameterise the lead field, i.e., the spatial projection of source activity to the sensors. In the present version, this is done by assuming that each area is modelled by one equivalent current dipole (ECD) [42]. In other words, DCM is used not only to solve for the connectivity but, simultaneously, also for the spatial parameters.

In the following, we will describe the graphical user interface (GUI). The GUI allows to comfortably specify all parameters of a model. If you want to specify lots of 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.

### 18.1 Calling DCM for ERP/ERF

Currently, the GUI is hidden away as a toolbox. You can find it under *Toolboxes* → *apierp*. The GUI is partitioned into 5 parts, going from the top to the bottom. The first part is about loading and saving DCMs. The second part is about selecting some data, the third is used to specify a spatial forward model, the fourth is for specifying the connectivity model, and the last allows you to estimate parameters and view results.

In general, you have to specify the data and model in a specific order. The first stage is data selection, the second the spatial model, followed by the connectivity model specification. This order is necessary, because there are dependencies among the three parts that would be hard to resolve without this order. However, at any time, you can switch forth and back between model parts. Also, within each part, you can specify information in any order you like.

## 18.2 cd, load and save

At the top of the GUI, you can *cd* to a new working directory, load existing DCMs or save a new one. In general, you can *save* and *load* during model specification at any time.

## 18.3 Data selection

In this part, you select the data. These can be either event-related potentials or fields (i.e. a data matrix (channels  $\times$  peri-stimulus time), averaged over single trials). Currently, you can only analyse one or two evoked responses in the same model<sup>1</sup>. To select data, click on the button *choose data*. Select an M/EEG SPM-matfile which is the output of the SPM preprocessing. Alternatively, if your data are in another format, you have to convert this file to the SPM-format (see the preprocessing chapter of this manual to see how this can be done). Below the *choose data* button, you can choose under *nr* which of the evoked responses in the SPM-matfile you want to model. For example, if you want to model the second and third evoked response within a SPM-matfile, specify indices 2 and 3. If your two evoked responses are in different files, you have to merge these files first. You can do this with the SPM preprocessing function *merge (spm-eeg-merge)*, s. chapter 13. Under *ms* you can specify the period of peri-stimulus time which you want to model. After your data are loaded, the time-series of all sensors are displayed in the SPM Graphics window.

In DCM, we use a projection of the data to some subspace to reduce the amount of data. Additionally, this data selection also serves as a tool to model only the salient features of the data. Currently, we are using a simple singular value decomposition (SVD) to decompose the data. You can select the number of (first) modes you want to keep. The default is 3, which we experimentally found to be a good value for our own data.

Furthermore, you can choose whether you want to model the mean or drifts of the data at sensor level. If you don't want any such terms, select 0 for detrending. Otherwise, select the number of discrete cosine transform terms you want to use to model the mean (1) or low-frequency drifts ( $> 1$ ).

If you are happy with your data, the projection and the detrending terms, you can click on the  $>$  (forward) button, which will bring you to the next stage *Spatial model specification*. Additionally, when pressing the forward button, the reduced data is displayed in the Graphics window. If you want to try other choices, you can press the  $<$  button (backward) button in the spatial modelling part. This will take you back to the data selection part, where you can change parameters and hit the forward button again.

## 18.4 Spatial model specification

With the present version of DCM, you have two options how to spatially model your data. Either you compute the leadfield, for each area, yourself, or you parameterise the leadfield using an equivalent current dipole model.

For the first option, you need to choose *fixed* in the pull-down menu. Then click on *load lead field* and specify a matfile. This matfile must contain a matrix with one column for each area. Each column must have one entry for each channel. For an example, see [19]. Of course, you must have some means of computing the leadfield for your experiment. Alternatively, you can opt for the alternative and make the leadfield a function of equivalent current dipoles (ECDs). The parameters of the ECDs (location and orientation) can be estimated by DCM. To do this, you need to either select *ECD EEG* or *ECD MEG*. For *ECD EEG*, you also need to select a sensor location file. This can be either a Polehmus file or a matfile with a coordinate matrix. Under *names* specify the names of all areas, one name per row. Under *locations*, specify the locations of these areas in MNI-space, again one location (three coordinates — x y z) per row. You can check the locations of dipoles by clicking on the *plot* button. This will visualize the dipole locations overlaid on an MRI template in MNI space. Note that DCM uses by default uninformative priors

---

<sup>1</sup>Note however, that the theoretical framework allows analysis of any number of evoked responses

on dipole orientations, but rather tight priors on locations [42]. One reason for the tight priors on locations is to ensure that 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 reconstruction 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. When you want to proceed to the next model specification stage, hit the  $>$  (forward) button and proceed to the *connectivity model specification*.

## 18.5 Connectivity model specification

Press *specify connections* to get access to selecting your model's connections between areas. There are 5 elements which you need to go through. The first three are the intrinsic connectivities. In DCM for ERP/ERF 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<sup>2</sup>.

In the present implementation, there is only one input allowed. This input can go to any area, where it only goes to early areas typically. You can select these receiving areas by selecting area indices in the  $C$  vector.

The  $B$  matrix contains all gain modulations of intrinsic connections. These modulations model the difference between the first and second evoked response. In other words, the DCM explains two evoked responses by explaining the first response by using the intrinsic connections only. The 2nd response is modelled by modulating these intrinsic connections by the weights in matrix  $B$ . For instance, if you want to allow modulations of forward connections only, you switch on those connections in  $B$  which are also selected in the intrinsic forward connectivity matrix.

## 18.6 Estimation

When you are done with model specification, you can hit the *estimate* button in the lower left corner. DCM will first try to save your DCM. Select a file to save to. After saving, DCM will 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 commented by iteration number, free energy  $F$ , and the change of  $F$  with respect to the updated variance parameters.

Note that a DCM for evoked responses is more complex than a DCM for fMRI. This means that more model parameters are used and, consequently, the estimation process takes longer. Expect something like 15 - 60 minutes, depending on model/data/computer specification.

## 18.7 Results

After estimation finished, you can assess the results by choosing from the pull-down menu at the bottom (middle).

With *ERPs channel* you can plot, for each mode, the data for both evoked responses, and its fit by the model.

When you select *ERPs sources*, the dynamics of each area are plotted. These corresponds to the (output) states of the dynamic system [19].

The option *coupling (A)* will take you to a summary about the posterior distributions of the intrinsic connectivities. 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.

---

<sup>2</sup>Currently, you can't model self-connections, which may be introduced with a later SPM update.

With the option *coupling(B)* you can access the equivalent posterior means for the gain modulations of the intrinsic connectivities and their probability that they are unequal the prior means.

With *coupling(C)* you see the same 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 (s. above).

The option *Input* shows you the estimated input function. As described by [19], this is a gamma function with an addition of some low-frequency terms.

With *Response*, you can plot the selected data, i.e. the modes you have selected for DCM analysis.

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.

# **Part V**

# **Utilities**



# Chapter 19

## Display Image

### Contents

---

19.1 Image to Display . . . . .	132
---------------------------------	-----

---

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.

## 19.1 Image to Display

Image to display.

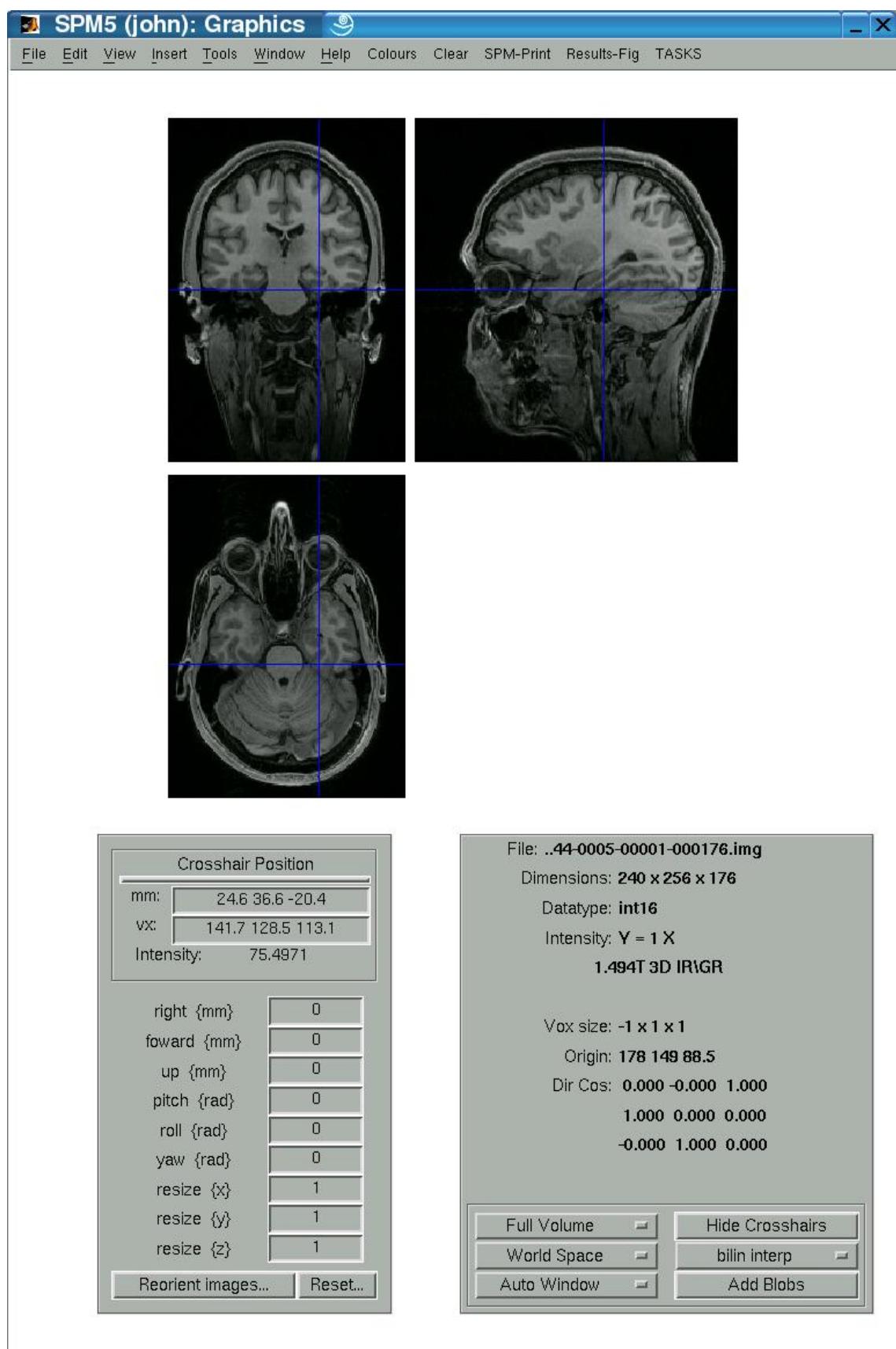


Figure 19.1: The Display routine.



# Chapter 20

## Check Registration

### Contents

---

<b>20.1 Images to Display</b>	.....	135
-------------------------------	-------	-----

---

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.

### 20.1 Images to Display

Images to display.



# Chapter 21

## Image Calculator

### Contents

---

<b>21.1 Input Images</b>	137
<b>21.2 Output Filename</b>	137
<b>21.3 Output Directory</b>	137
<b>21.4 Expression</b>	138
<b>21.5 Options</b>	138
<b>21.5.1 Data Matrix</b>	138
<b>21.5.2 Masking</b>	138
<b>21.5.3 Interpolation</b>	138
<b>21.5.4 Data Type</b>	138

---

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.

### 21.1 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.

### 21.2 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.

### 21.3 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.

## 21.4 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)  
 $f = 'sum(X)'$

## 21.5 Options

Options for image calculator

### 21.5.1 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))].

### 21.5.2 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.

### 21.5.3 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.

Bilinear Interpolation

- OK for PET, or realigned fMRI.

Sinc Interpolation

- Better quality (but slower) interpolation, especially with higher degrees.

### 21.5.4 Data Type

Data-type of output image

# Chapter 22

# DICOM Import

## Contents

---

<b>22.1 DICOM files</b>	139
<b>22.2 Directory structure for converted files</b>	139
<b>22.3 Output directory</b>	139
<b>22.4 Conversion options</b>	140
<b>22.4.1 Output image format</b>	140
<b>22.4.2 Use ICEDims in filename</b>	140

---

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/>

## 22.1 DICOM files

Select the DICOM files to convert.

## 22.2 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

## 22.3 Output directory

Select a directory where files are written. Default is current directory.

## 22.4 Conversion options

### 22.4.1 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.

### 22.4.2 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.

# Chapter 23

## MINC Import

### Contents

---

<b>23.1 MINC files</b>	141
<b>23.2 Options</b>	141
23.2.1 Data Type	141
23.2.2 NIFTI Type	141

---

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, but due to be superceded by a new version relatively soon. 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.

### 23.1 MINC files

Select the MINC files to convert.

### 23.2 Options

Conversion options

#### 23.2.1 Data Type

Data-type of output images. Note that the number of bits used determines the accuracy, and the amount of disk space needed.

#### 23.2.2 NIFTI Type

Output files can be written as .img + .hdr, or the two can be combined into a .nii file.



# Chapter 24

## ECAT Import

### Contents

---

<b>24.1 ECAT files</b> . . . . .	143
<b>24.2 Options</b> . . . . .	143
<b>24.2.1 NIFTI Type</b> . . . . .	143

---

ECAT 7 Conversion. ECAT 7 is the image data format used by the more recent CTI PET scanners.

### 24.1 ECAT files

Select the ECAT files to convert.

### 24.2 Options

Conversion options

#### 24.2.1 NIFTI Type

Output files can be written as .img + .hdr, or the two can be combined into a .nii file.



# Chapter 25

## Deformations

### Contents

---

<b>25.1 Composition</b> . . . . .	<b>145</b>
25.1.1 Imported _sn.mat . . . . .	145
25.1.2 DARTEL flow . . . . .	146
25.1.3 Deformation Field . . . . .	146
25.1.4 Identity . . . . .	146
25.1.5 Inverse . . . . .	146
25.1.6 Composition . . . . .	152
<b>25.2 Save as</b> . . . . .	<b>159</b>
<b>25.3 Apply to</b> . . . . .	<b>159</b>
<b>25.4 Interpolation</b> . . . . .	<b>159</b>

---

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.

Note that ideal deformations can be treated as members of a Lie group. Future versions of SPM may base its warping on such principles.

### 25.1 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 left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first o second) o third)...o last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

#### 25.1.1 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.

**25.1.2 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.

**25.1.3 Deformation Field**

Deformations can be thought of as vector fields. These can be represented by three-volume images.

**25.1.4 Identity**

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.

**25.1.5 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

## 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  $y \circ x:A \rightarrow C$ . Compositions can be combined in an associative way, such that  $z \circ (y \circ x) = (z \circ y) \circ x$ .

In this utility, the left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first o second) o third)...o last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

**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.

**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.

**Deformation Field** Deformations can be thought of as vector fields. These can be represented by three-volume images.

**Identity** 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.

**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  $y' \circ y = y \circ y' = Id$ , where  $Id$  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

**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  $y \circ x:A \rightarrow C$ . Compositions can be combined in an associative way, such that  $z \circ (y \circ x) = (z \circ y) \circ x$ .

In this utility, the left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first o second) o third)...o last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

**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.

**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.

**Deformation Field** Deformations can be thought of as vector fields. These can be represented by three-volume images.

**Identity** 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.

**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  $y' \circ y = y \circ y' = Id$ , where  $Id$  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

**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 left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first o second) o third)...o last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

*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 imagethat 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 imagethat was originally used to estimate the deformation.

*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.

*Deformation Field* Deformations can be thought of as vector fields. These can be represented by three-volume images.

*Identity* 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.

*IMAGE TO BASE INVERSE ON* Specify the image file on which to base the dimensions, orientation etc.

**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 left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first o second) o third)...o last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

*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 imagethat 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 imagethat was originally used to estimate the deformation.

**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.

**DEFORMATION FIELD** Deformations can be thought of as vector fields. These can be represented by three-volume images.

**IDENTITY** 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.

**Image to base inverse on** Specify the image file on which to base the dimensions, orientation etc.

**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  $zoy(yox) = (zoy)o x$ .

In this utility, the left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first o second) o third)...o last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

**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.

**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.

**Deformation Field** Deformations can be thought of as vector fields. These can be represented by three-volume images.

**Identity** 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.

**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

**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 " $\circ$ ". 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  $y \circ x:A \rightarrow C$ . Compositions can be combined in an associative way, such that  $z \circ (y \circ x) = (z \circ y) \circ x$ .

In this utility, the left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first  $\circ$  second)  $\circ$  third)... $\circ$  last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

**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.

**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.

**DEFORMATION FIELD** Deformations can be thought of as vector fields. These can be represented by three-volume images.

**IDENTITY** 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.

**Image to base inverse on** Specify the image file on which to base the dimensions, orientation etc.

**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 left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first o second) o third)...o last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

**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 imagethat 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 imagethat was originally used to estimate the deformation.

**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.

**Deformation Field** Deformations can be thought of as vector fields. These can be represented by three-volume images.

**Identity** 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.

### Image to base inverse on

Specify the image file on which to base the dimensions, orientation etc.

#### 25.1.6 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 left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first o second) o third)...o last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

### 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 imagethat 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 imagethat was originally used to estimate the deformation.

### 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.

### Deformation Field

Deformations can be thought of as vector fields. These can be represented by three-volume images.

### Identity

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.

### 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  $y' \circ y = y \circ y' = Id$ , where  $Id$  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

**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 left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first o second) o third)...o last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

**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 imagethat 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 imagethat was originally used to estimate the deformation.

**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.

**Deformation Field** Deformations can be thought of as vector fields. These can be represented by three-volume images.

**Identity** 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.

**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 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

**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 left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first o second) o third)...o last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

**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 imagethat 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 imagethat was originally used to estimate the deformation.

**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.

**DEFORMATION FIELD** Deformations can be thought of as vector fields. These can be represented by three-volume images.

**IDENTITY** 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.

**Image to base inverse on** Specify the image file on which to base the dimensions, orientation etc.

**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 left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first o second) o third)...o last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

**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 imagethat 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 imagethat was originally used to estimate the deformation.

**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.

**Deformation Field** Deformations can be thought of as vector fields. These can be represented by three-volume images.

**Identity** 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.

**Image to base inverse on** Specify the image file on which to base the dimensions, orientation etc.

**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  $y \circ x:A \rightarrow C$ . Compositions can be combined in an associative way, such that  $z \circ (y \circ x) = (z \circ y) \circ x$ .

In this utility, the left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first o second) o third)...o last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

**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.

**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.

**Deformation Field** Deformations can be thought of as vector fields. These can be represented by three-volume images.

**Identity** 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.

**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

**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  $y \circ x:A \rightarrow C$ . Compositions can be combined in an associative way, such that  $z \circ (y \circ x) = (z \circ y) \circ x$ .

In this utility, the left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first o second) o third)...o last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

**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.

**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.

**Deformation Field** Deformations can be thought of as vector fields. These can be represented by three-volume images.

**Identity** 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.

**Image to base inverse on** Specify the image file on which to base the dimensions, orientation etc.

**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  $y \circ x:A \rightarrow C$ . Compositions can be combined in an associative way, such that  $z \circ (y \circ x) = (z \circ y) \circ x$ .

In this utility, the left-to-right order of the compositions is from top to bottom (note that the rightmost deformation would actually be applied first). i.e. ...((first o second) o third)...o last. The resulting deformation field will have the same domain as the first deformation specified, and will map to voxels in the codomain of the last specified deformation field.

**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.

**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.

**Deformation Field** Deformations can be thought of as vector fields. These can be represented by three-volume images.

**Identity** 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.

## 25.2 Save as

Save the result as a three-volume image. "y\_" will be prepended to the filename. The result will be written to the current directory.

## 25.3 Apply to

Apply the resulting deformation field to some images. The warped images will be written to the current directory, and the filenames prepended by "w". Note that trilinear interpolation is used to resample the data, so the original values in the images will not be preserved.

## 25.4 Interpolation

The method by which the images are sampled when being written in a different space.

Nearest Neighbour: - Fastest, but not normally recommended.

Bilinear Interpolation: - OK for PET, or realigned fMRI.

B-spline Interpolation: - Better quality (but slower) interpolation [62], especially with higher degree splines. Do not use B-splines when there is any region of NaN or Inf in the images.



## **Part VI**

## **Tools**



# Chapter 26

## High-Dimensional Warping

### Contents

---

<b>26.1 Subjects</b> . . . . .	<b>163</b>
26.1.1 Subject . . . . .	163
<b>26.2 Bias Correction Options</b> . . . . .	<b>164</b>
26.2.1 Iterations . . . . .	164
26.2.2 Bias FWHM . . . . .	164
26.2.3 Bias regularisation . . . . .	164
26.2.4 Levenberg-Marquardt regularisation . . . . .	164
<b>26.3 Warping Options</b> . . . . .	<b>164</b>
26.3.1 Iterations . . . . .	164
26.3.2 Warping regularisation . . . . .	164

---

This toolbox is a Bayesian method for three dimensional registration of brain images [4]. A finite element approach is used to obtain a maximum a posteriori (MAP) estimate of the deformation field at every voxel of a template volume. The priors used by the MAP estimate penalize unlikely deformations and enforce a continuous one-to-one mapping. The deformations are assumed to have some form of symmetry, in that priors describing the probability distribution of the deformations should be identical to those for the inverses (i.e., warping brain A to brain B should not be different probabilistically from warping B to A). A gradient descent algorithm is used to estimate the optimum deformations.

Deformation fields are written with the same name as the moved image, but with "y\_" prefixed on to the filename. Jacobian determinant images are also written (prefixed by "jy\_").

### 26.1 Subjects

Specify pairs of images to match together.

#### 26.1.1 Subject

Two images of the same subject, which are to be registered together. Prior to nonlinear high-dimensional warping, the images should be rigidly registered with each other.

##### Reference Image

This is the reference image, which remains stationary.

### Moved Image

This is the moved image, which is warped to match the reference.

## 26.2 Bias Correction Options

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.

Before registering the images, an approximate bias correction is estimated for the moved image. This is based on minimising the difference between the images in a symmetric way. Prior to registering the images, they should be rigidly aligned together. The bias correction is estimated once for these aligned images.

### 26.2.1 Iterations

Number of iterations for the bias correction

### 26.2.2 Bias FWHM

FWHM of Gaussian smoothness of bias. If your intensity nonuniformity 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 nonuniformity 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 nonuniformities.

### 26.2.3 Bias regularisation

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 nonuniformity 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 nonuniformity.

### 26.2.4 Levenberg-Marquardt regularisation

Levenberg-Marquardt regularisation keeps the bias correction part stable. Higher values mean more stability, but slower convergence.

## 26.3 Warping Options

There are a couple of user-customisable warping options.

### 26.3.1 Iterations

Number of iterations for the warping.

### 26.3.2 Warping regularisation

There is a tradeoff between the smoothness of the estimated warps, and the difference between the registered images. Higher values mean smoother warps, at the expense of a lower mean squared difference between the images.

# Chapter 27

## DARTEL Tools

### Contents

---

<b>27.1 Initial Import</b>	166
27.1.1 Parameter Files	166
27.1.2 Output Directory	166
27.1.3 Bounding box	166
27.1.4 Voxel size	167
27.1.5 Image option	167
27.1.6 Grey Matter	167
27.1.7 White Matter	167
27.1.8 CSF	167
<b>27.2 Run DARTEL (create Templates)</b>	167
27.2.1 Images	167
27.2.2 Settings	167
<b>27.3 Run DARTEL (existing Templates)</b>	169
27.3.1 Images	170
27.3.2 Settings	170
<b>27.4 Create Warped</b>	171
27.4.1 Flow fields	171
27.4.2 Images	171
27.4.3 Modulation	171
27.4.4 Time Steps	171
27.4.5 Interpolation	171
<b>27.5 Jacobian determinants</b>	172
27.5.1 Flow fields	172
27.5.2 Time Steps	172
<b>27.6 Create Inverse Warped</b>	172
27.6.1 Flow fields	172
27.6.2 Images	172
27.6.3 Time Steps	172
27.6.4 Interpolation	172
<b>27.7 Kernel Utilities</b>	172
27.7.1 Generate Residuals	173
27.7.2 Kernel from Resids	173
27.7.3 Kernel from Flows	174

---

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.

## 27.1 Initial Import

Images first need to be imported into a form that DARTEL can work with. This involves taking the results of the segmentation (\*.seg\_sn.mat) [9], 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.

### 27.1.1 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.

### 27.1.2 Output Directory

Select the directory where the resliced files should be written.

### 27.1.3 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.

### 27.1.4 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.

### 27.1.5 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.

### 27.1.6 Grey Matter

Produce a resliced version of this tissue class?

### 27.1.7 White Matter

Produce a resliced version of this tissue class?

### 27.1.8 CSF

Produce a resliced version of this tissue class?

## 27.2 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. An example is shown in figure 27.1.

### 27.2.1 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.

### 27.2.2 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.

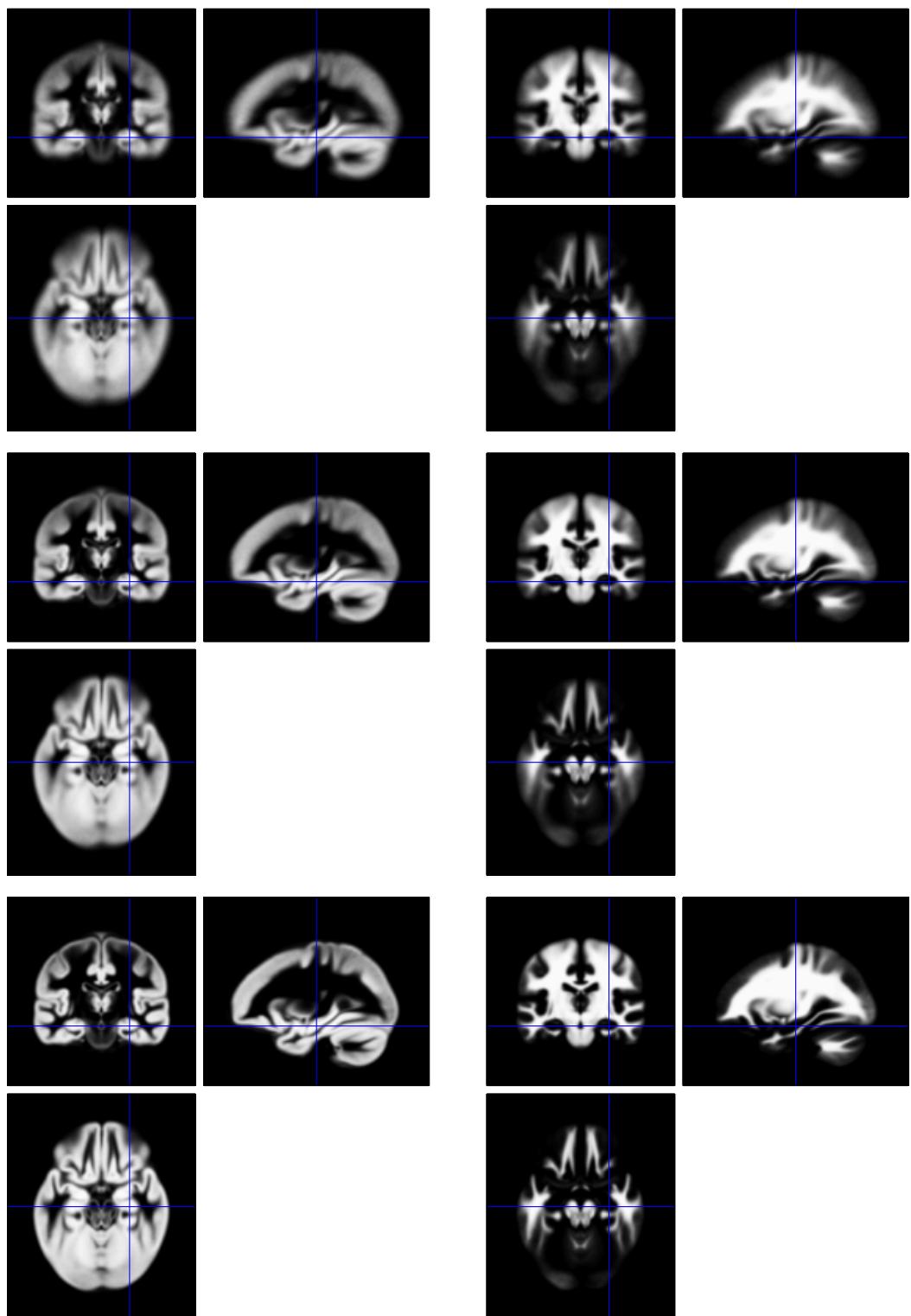


Figure 27.1: This figure shows the intensity averages of grey (left) and white (right) matter images after different numbers of iterations. The top row shows the average after initial rigid-body alignment. The middle row shows the images after three iterations, and the bottom row shows them after 18 iterations.

### 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  $(-\lambda * \text{Laplacian} + id1)^2 + id2$ .

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.

## 27.3 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.

### 27.3.1 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.

### 27.3.2 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  $(-\lambda * \text{Laplacian} + id1)^2 + id2$ .

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 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.

## 27.4 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.

### 27.4.1 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).

### 27.4.2 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.

### 27.4.3 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. The square-root modulation is for special applications, so can be ignored in most cases.

### 27.4.4 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).

### 27.4.5 Interpolation

The method by which the images are sampled when being written in a different space.

Nearest Neighbour: - Fastest, but not normally recommended.

Bilinear Interpolation: - OK for PET, or realigned fMRI.

B-spline Interpolation: - Better quality (but slower) interpolation [62], especially with higher degree splines. Do not use B-splines when there is any region of NaN or Inf in the images.

## 27.5 Jacobian determinants

Create Jacobian determinant fields from flowfields.

### 27.5.1 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).

### 27.5.2 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).

## 27.6 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.

### 27.6.1 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).

### 27.6.2 Images

Select the image(s) to be inverse normalised. These should be in alignment with the template image generated by the warping procedure.

### 27.6.3 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).

### 27.6.4 Interpolation

The method by which the images are sampled when being written in a different space.

Nearest Neighbour: - Fastest, but not normally recommended.

Bilinear Interpolation: - OK for PET, or realigned fMRI.

B-spline Interpolation: - Better quality (but slower) interpolation [62], especially with higher degree splines. Do not use B-splines when there is any region of NaN or Inf in the images.

## 27.7 Kernel Utilities

DARTEL can be used for generating Fisher kernels for various kernel pattern-recognition procedures. The idea is to use both the flow fields and residuals to generate two separate Fisher kernels (see the work of Jaakkola and Haussler for more information). Residual images need first to be generated in order to compute the latter kernel, prior to the dot-products being computed.

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.

### 27.7.1 Generate Residuals

Generate residual images in a form suitable for computing a Fisher kernel. 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 the kernel from the residuals, with a kernel derived from the flow-fields. Such a combined kernel should then encode more relevant information than the individual kernels alone.

#### 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).

#### Flow fields

Select the flow fields for each subject.

#### Template

Residual differences are computed between the warped images and template, and these are scaled by the square root of the Jacobian determinants (such that the sum of squares is the same as would be computed from the difference between the warped template and individual images).

#### 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).

#### Smoothing

The residuals can be smoothed with a Gaussian to reduce dimensionality. More smoothing is recommended if there are fewer training images.

### 27.7.2 Kernel from Resids

Generate a kernel matrix from residuals. 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).

**27.7.3 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 28

## FieldMap Toolbox

### 28.1 Introduction

This chapter describes how to use the FieldMap toolbox<sup>1</sup> for creating unwrapped field maps that can be used to do geometric distortion correction of EPI images [40, 39, 37, 38, 2]. The toolbox is designed to be interactive so that the user can see the effect of applying different field maps and unwarping parameters to EPI images. However, once a set of parameters has been established for a specific scanning protocol, the routines used by the toolbox can also be scripted. FieldMap Version 2.0 for SPM5 also supports the new SPM5 User-Interface allowing jobs to be created and scripted. The toolbox creates a voxel-displacement map that can be used with Realign & Unwarp for doing a combined static and dynamic distortion correction.

#### 28.1.1 Latest News

Version 2.0 of FieldMap is now available for SPM5. To ensure full compatibility between the FieldMap toolbox and Realign & Unwarp in SPM5, please also update `spm_get_image_def.m` and `spm_config_realign_and_unwarp.m` in the main SPM5 distribution. The main changes are listed below:

1. The convention used to describe the direction of the k-space traversal is now 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. The change in convention means that if previously the polarity of phase-encode blips in the FieldMap gui was usually set to be negative, it should now be positive and vice versa. This also applies to the default value for `spm.def.K_SPACE_TRAVERSAL_BLIP_DIR` in the `spm_defaults.m` file which should be set to -1 if it was +1 previously and vice versa.
2. In the previous version of FieldMap, there was a bug in the application of the Jacobian modulation when using an EPI-based field map. This has now been fixed. The toolbox now allows the user to select a specified defaults file. This means it is possible to have more than one default file containing different sets of parameters relating to different field map and /or EPI sequences. This is particularly useful for sites where more than one sequence is routinely used.
3. The toolbox can be run using the new SPM5 User Interface. This will allow FieldMap jobs to be saved serving as a log about how the data was processed. It will also facilitate scripting and batching jobs.

---

<sup>1</sup> FieldMap Version 2.0 can now be downloaded as part SPM5: <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>

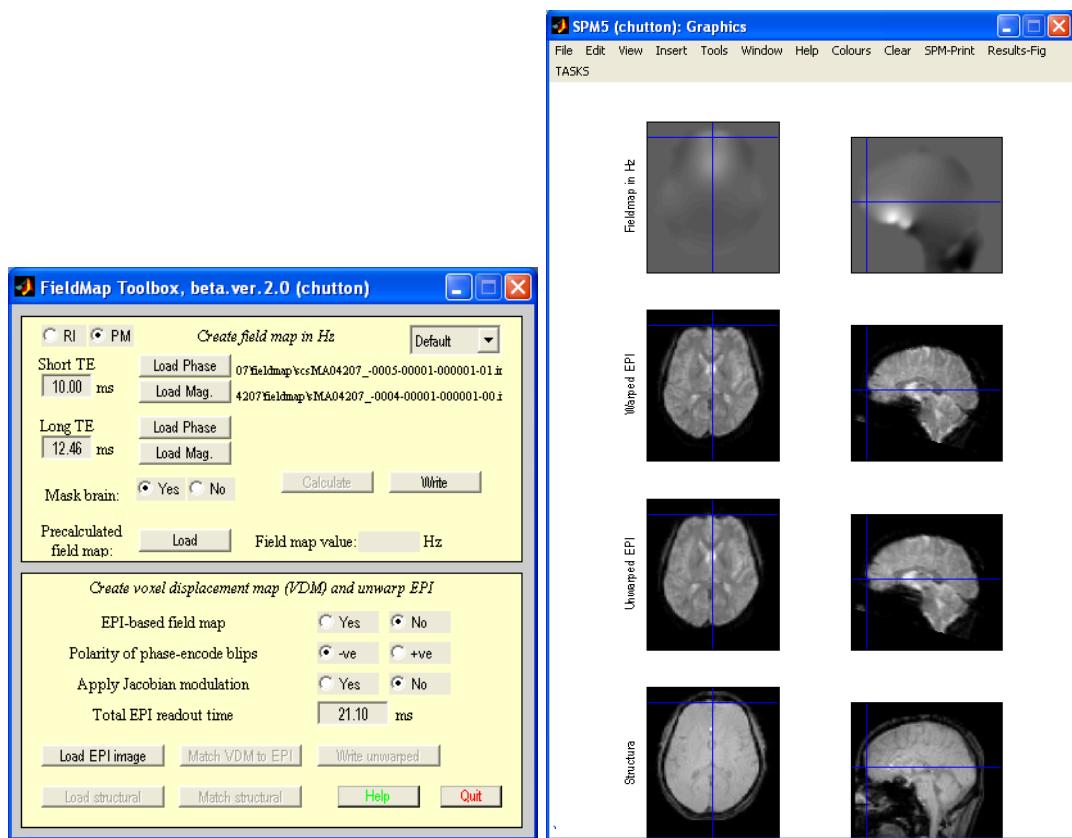


Figure 28.1: FieldMap GUI and Results.

4. The toolbox now reads and writes files in NIfTI format (as does SPM5). The toolbox now save a vdm5\_- file which will also be expected by Realign & unwarp. If doing Unwarp with multiple sessions, but a single vdm5\_- file, it will be necessary to select this file for each session.
5. All versions of FieldMap and Unwarp are only designed to work with images collected with the phase-encode direction in y.

## 28.2 Creating Field Maps Using the FieldMap GUI

The FieldMap Toolbox GUI is shown on the left Figure 28.1. 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:

### 28.2.1 Create field map in Hz

#### Load defaults file (FieldMap Version 2.0 for SPM5 only)

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 `spm_defaults.m` or create a new file called `spm_defaults_NAME.m` (as described in Section 28.3).

### 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. FieldMap version 1.1 and 2.0 will ask user 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 Analyze format images for SPM2 and Nifti format images for SPM5. 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 28.7 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

<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 28.3.

<sup>3</sup> MRIcro is freely available from <http://www.cla.sc.edu/psyc/faculty/rorden/mricro.html>.

3. Phase unwrapping
4. Smoothing and dilation of the processed fieldmap

The processed field map (in Hz) is displayed in the graphics window (top row, right Figure 28.1) 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 (or vdm5\_NAME-OF-FIRST-INPUT-IMAGE.img for version 2.0) 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 SPM2 or SPM5 (see Section 28.6).

### **Write**

Write out the processed field map (in Hz) as an Analyze format image in SPM2 or a Nifti format image in SPM5. 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 and in Analyze format for SPM2 or Nifti format for SPM5. 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 28.1) 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.

## **28.2.2 Create voxel displacement map (VDM) and unwarped 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 28.2.2.

### **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.

### **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 Analyze format for SPM2 and Nifti format for SPM5). This image is automatically unwarped using the VDM calculated with the current parameters. The warped and the unwarped image are displayed in the graphics window underneath the field map (middle rows, right, Figure 28.1).

**Match VDM to EPI**

Select this option to match the field map magnitude data to the EPI image before it is used to un warp 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 unwarped**

Write unwarped EPI Analyze image with the filename uNAME\_OF\_EPI.img.

**Load structural**

Load a structural image for comparison with unwarped EPI. This is displayed in the graphics window below the other images (bottom row, right fig 1).

**MatchStructural**

Coregister the structural image to the unwarped EPI and write the resulting transformation matrix to the .mat file of the selected structural image.

**Help**

Call spm\_help to display FieldMap.man.

**Quit**

Quit the toolbox and closes all windows associated with it.

## 28.3 Using the spm\_defaults file

Input parameters and the mode in which the toolbox works can be customised in the defaults file called spm\_defaults.m. FieldMap version 2.0 for SPM5, allows different defaults files to be selected. This means that it is possible to have more than one set of default parameters, e.g. to accommodate different scanners or sequences. To be recognised by the FieldMap toolbox, these files must be named spm\_defaults\_NAME.m and have the same format as the existing spm\_defaults.m. These defaults files can be loaded using the FieldMap GUI (as described in section 28.2.1) or selected when filling in the job fields using the new SPM5 User-Interface (as described in section 6).

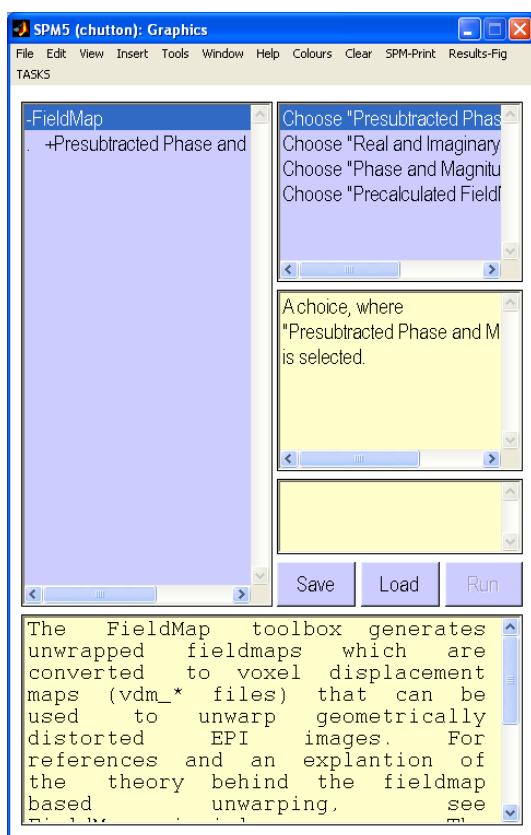


Figure 28.2: FieldMap using the SPM5 User Interface.

## 28.4 Using the SPM5 User Interface

FieldMap version 2.0 for SPM5 can be run using the new SPM5 UI (Figure 28.2). To do this, from the top menu on the SPM5 Graphics window, select TASKS, then Tools then FieldMap. Select the type of FieldMap job to run from 1) Presubtracted Phase and Magnitude Data (this means a single phase and magnitude pair), 2) Real and Imaginary Data, 3) Phase and Magnitude Data (this means a double phase and magnitude pair) and 4) Precalculated FieldMap. Double click on the job type in the left panel, highlight Data in the left panel then select New Subject in the right panel, then double click on Subject in the left panel to fill in the different job fields. All fields marked with an X must be filled in. The other fields can be optionally filled or contain default values which can be changed if required. To fill in a field, highlight it in the left panel and then click on the Specify in the top right hand corner to select a file, a menu option or enter a value etc. Once all of the fields have been specified the job can be saved and run. It can also be loaded at a later date and edited for another set of data etc.

## 28.5 Using the FieldMap in Batch scripts

FieldMap\_preprocess.m which calls FieldMap\_create.m give 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 = '/home/chutton/study1/subj1/fieldmap')  
*epi\_dir* - name of directory containing epi images. (e.g. epi\_dir = '/home/chutton/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 unwarped version of the EPI saved with the prescript "u".

## 28.6 Using the VDM file with Unwarp

In SPM, select the Realign + Unwarp option. When requested, select the vdm5\_ or vdm5\_- file for the subject and/or session. If you acquired more than one session and a field map for each session, select the vdm5\_\* file for each corresponding session. If you acquired more than one session but only one fieldmap, select the vdm5\_\* file for the first session and then when asked to select the vdm5\_\* file for the other sessions, select the first one again. For more information about Unwarp see <http://www.fil.ion.ucl.ac.uk/spm/toolbox/unwarp>.

## 28.7 Appendices

### 28.7.1 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  $\geq 0$  (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  $1/(2\pi \times \text{difference in echo time})$  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.

### 28.7.2 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 unwarped, resulting in a VDM. The readout time is specified by spm\_def.TOTAL\_EPI\_READOUT\_TIME (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 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 +/-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\_BLIP\_DIR = +/- 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 unwarped EPI image. This modulates the intensity of the unwarped 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 unwarped EPI by  $1 + \text{the 1-d derivative of the VDM}$  in the phase direction. An intensity adjustment of this nature may improve the coregistration results between an unwarped 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.

### 28.7.3 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`.



## **Part VII**

# **Data sets and examples**



# Chapter 29

## Auditory fMRI data

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 (64x64x64 3mm x 3mm x 3mm 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`. 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`. These images are stored in Analyse format and are available from the SPM site <http://www.fil.ion.ucl.ac.uk/spm/data/>. 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).

To analyse the data, first create a new directory DIR

eg. `c:\home\wpenny\fmri_analysis\auditory`, in which to place the results of your analysis. Then create 3 subdirectories (i) `jobs`, (ii) `classical` and (iii) `bayesian`. As the analysis proceeds these directories will be filled with job-specification files, design matrices and models estimated using classical or Bayesian methods.

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 'command' 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.

### 29.1 Spatial pre-processing

#### 29.1.1 Realignment

Under the spatial pre-processing section of the SPM base window select 'Realign' from the 'Realign' pulldown menu. This will call up a realignment job specification in the graphics window. Then

- Select 'New Realign:Estimate and Reslice'
- Open the newly created 'Realign:Estimate and Reslice' option.
- 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. '`fM000*.img`'.
- Save the job file as eg. `DIR/jobs/realign.mat`.
- Press the RUN button in the graphics window.

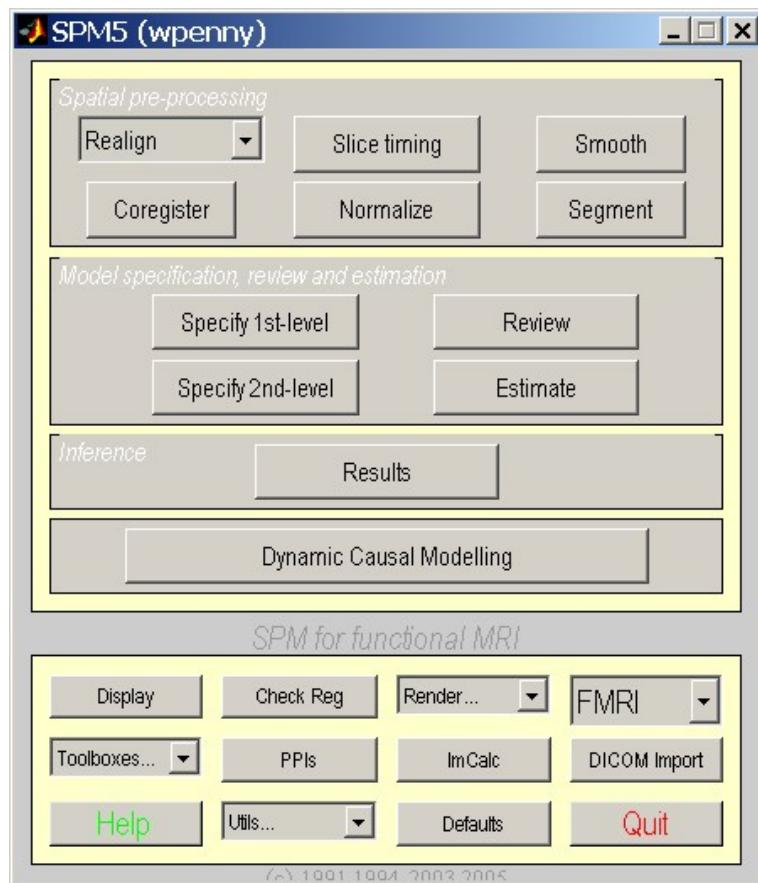


Figure 29.1: The SPM base window comprises three sections i) spatial pre-processing, (ii) model specification, review and estimation and (iii) inference.

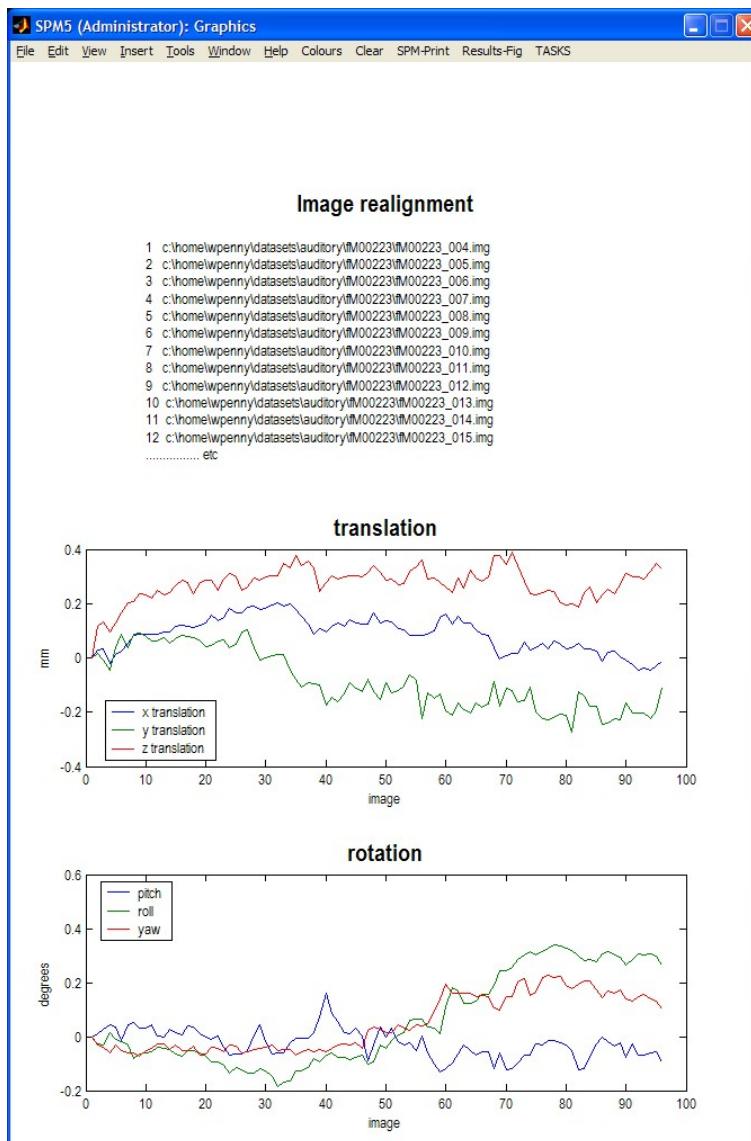


Figure 29.2: Realignment of auditory data.

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 29.2. These data are also saved to a file eg. `rp_fM00223_004.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. `meanfM00223_004.img` which will be used in the next step of spatial processing - coregistration.

### 29.1.2 Coregistration

Press the ‘Coreg’ button. This will call up the specification of a coregistration job in the graphics window.

- Select New ”Coreg;Estimate”
- Double-click on the newly created Coreg;Estimate

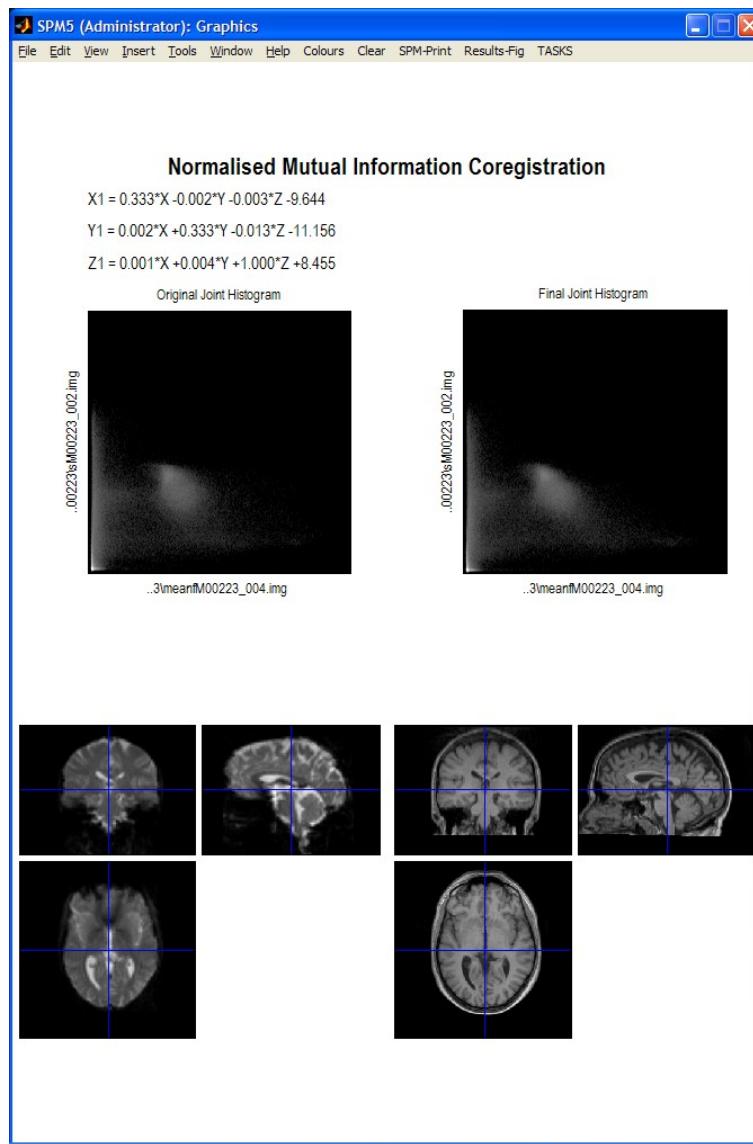


Figure 29.3: *Mutual Information Coregistration of Auditory data.*

- Highlight ‘Reference Image’ and then select the mean fMRI scan from realignment eg. `meanfM00223_004.img`
- Highlight ‘Source Image’ and then select the structural image eg. `sM00223_002.img`.
- Press the Save button and save the job as `coreg.job`
- Then press RUN

SPM will then implement a coregistration between the structural and functional data that maximises the mutual information. The image in figure 29.3 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 base window and then the select the Reference and Source Images specified above ie `meanfM00223_004.img` and `sM00223_002.img`. SPM will then produce an image like that shown in Figure 29.4 in the graphics window. You can then use your mouse to navigate these images to confirm that there is an anatomical correspondence.

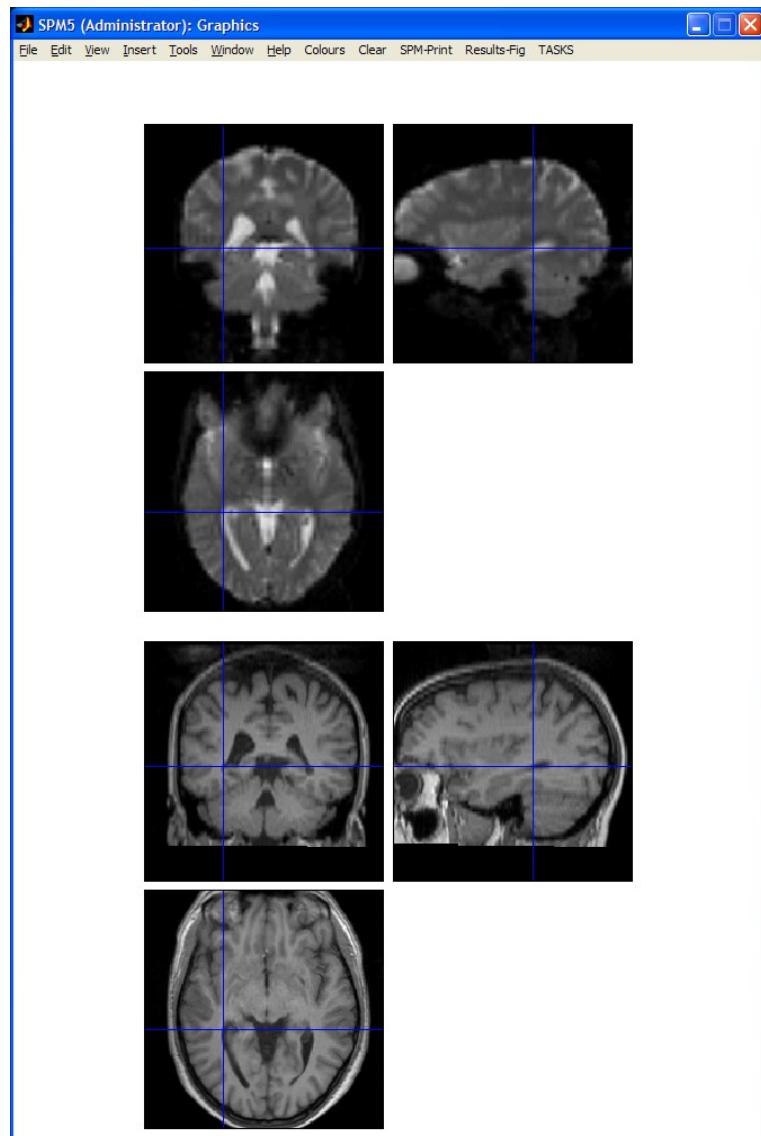


Figure 29.4: *Checking registration of functional and ‘registered’ structural data.*

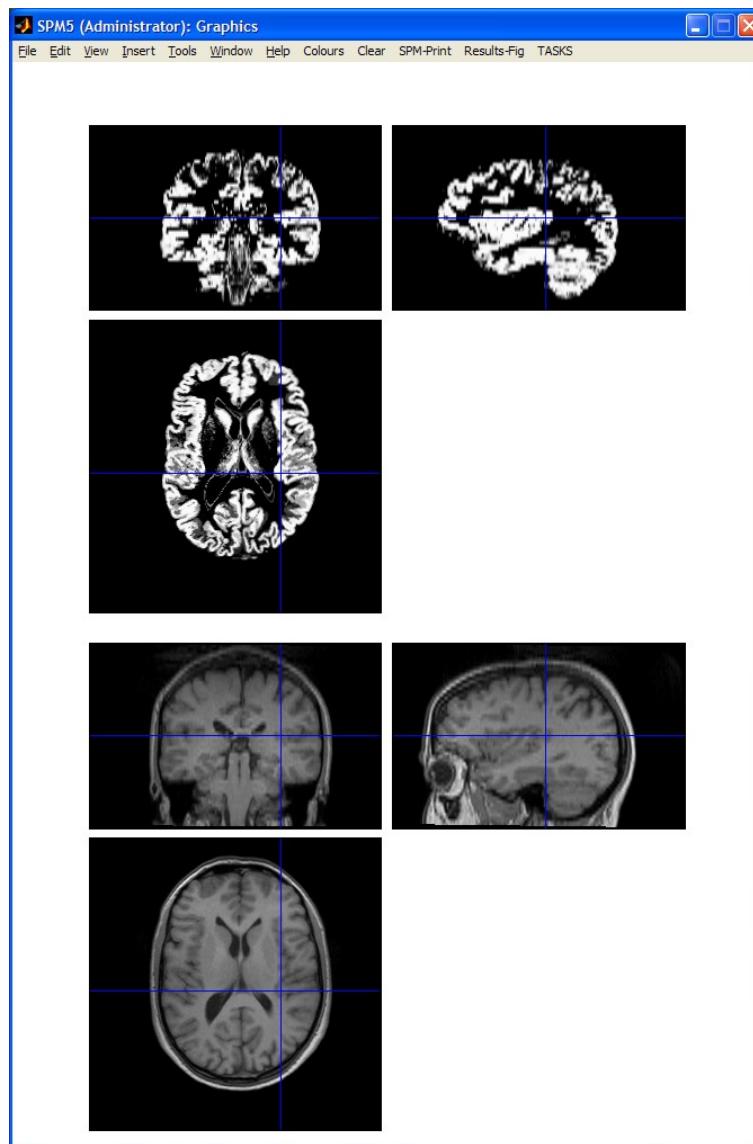


Figure 29.5: *Gray matter image and ‘registered’ structural image.*

### 29.1.3 Segmentation

Press the ‘Segment’ button. This will call up the specification of a segmentation job in the graphics window. Highlight the Data field and then select the subjects registered anatomical image eg. `sM00223_002.img`. 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.

Faster, though perhaps less optimal results can be obtained by eg. reducing the number of Gaussians per class from [2 2 2 4] to eg. [1 1 1 4], increasing the sampling distance from eg. 3 to 4mm. These options can be edited under the ‘Custom’ sub-menu and saved before the job is run. The results obtained in figure 29.5 were obtained using the default values.

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 (press segment and select . Figure 29.5 shows the gray matter image, `c1sM0023_002.img` along with the original structural.

SPM will also write a spatial normalisation eg. `sM00223_0020_seg_sn.mat` and inverse spatial normalisation parameters `sM00223_0020_seg_inv_sn.mat` to files in the original structural directory. These can be used to normalise the functional data.

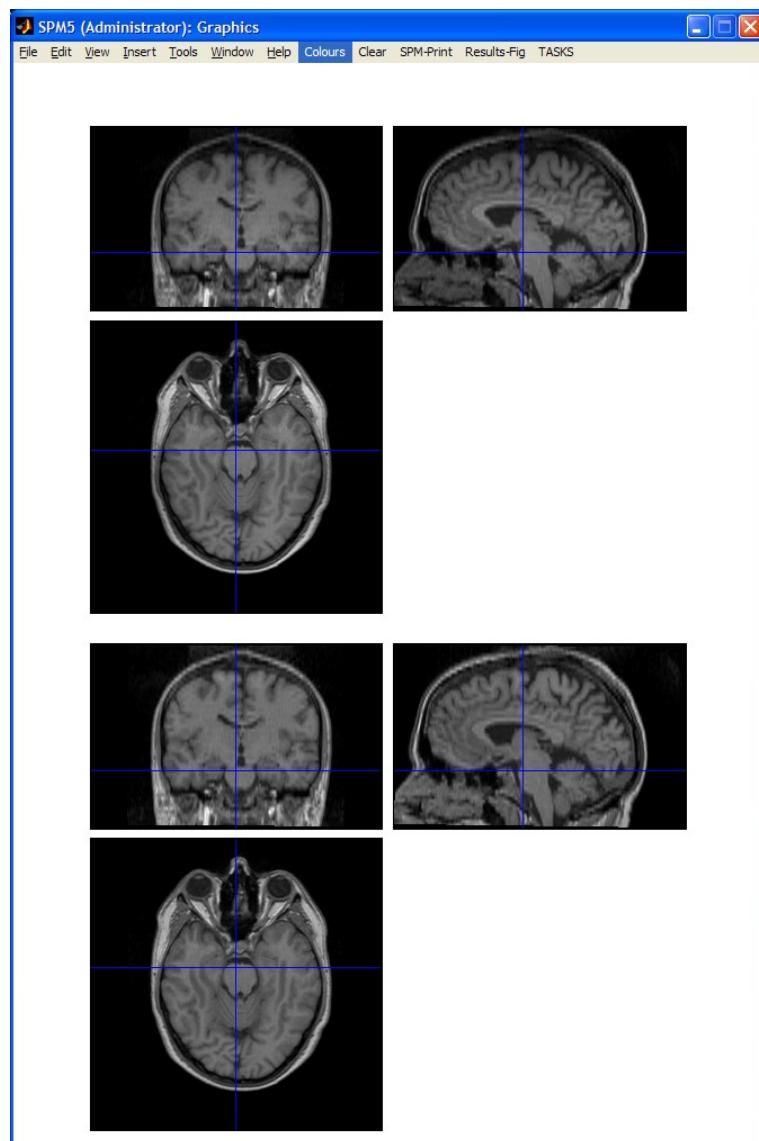


Figure 29.6: Structural image (top) and bias-corrected structural image (bottom). Notice that the original structural is darker at the top than at the bottom. This non-uniformity has been removed in the bias-corrected image.

### 29.1.4 Normalize

Press the ‘Normalize’ button. This will call up the specification of a normalise job in the graphics window.

- Select a New ”Normalise:Write” field. This allows previously estimated warps to be applied to a series of images.
- Highlight ‘Data’, select New ”Subject”
- Open ‘Subject’, highlight ‘Parameter File’ and select the `sM00223_0020_seg_sn.mat` file that you created in the previous section
- Highlight images to write and select all of the realigned functional images ‘`rFM000*.img`’. Note: This can be done efficiently by changing the filter in the SPM file selector to `^r.*`. SPM will then only list those files beginning with the letter *r* ie. those that have been realigned. You can then right click over the listed files, choose ‘Select all’ and press ‘Done’.
- Open ‘Writing Options’, and change ‘Voxel sizes’ from [2,2,2] to [3,3,3].<sup>1</sup>
- Press ‘Save’, save the job as `normalise.mat` and then press ‘Run’.

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

- Press ‘Normalise’, select New ”Normalise:Write”
- Open ‘Normalise: Write’, highlight ‘Data’, select New ”Subject”
- Open ‘Subject’
- Highlight ‘Parameter File’, select the `sM00223_0020_seg_sn.mat` file that you created in the previous section, press ‘Done’.
- Highlight ‘Images to Write’, select the bias-corrected structural eg. `msM00223_002.img`, 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 `norm_struct.mat` and press ‘Run’.

### 29.1.5 Smoothing

Press the ‘Smooth’ button<sup>3</sup>. This will call up the specification of a smooth job in the graphics window.

- Open ‘Smooth’, select ‘Images to Smooth’ and then select the spatially normalised files created in the last section eg. `wrfM000*.img`.
- 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 ‘Run’.

---

<sup>1</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>2</sup>Beginners may wish to skip this step, and instead just superimpose functional activations on an ‘average structural image’.

<sup>3</sup>The smoothing step is unnecessary if you are only interested in Bayesian analysis of your functional data.

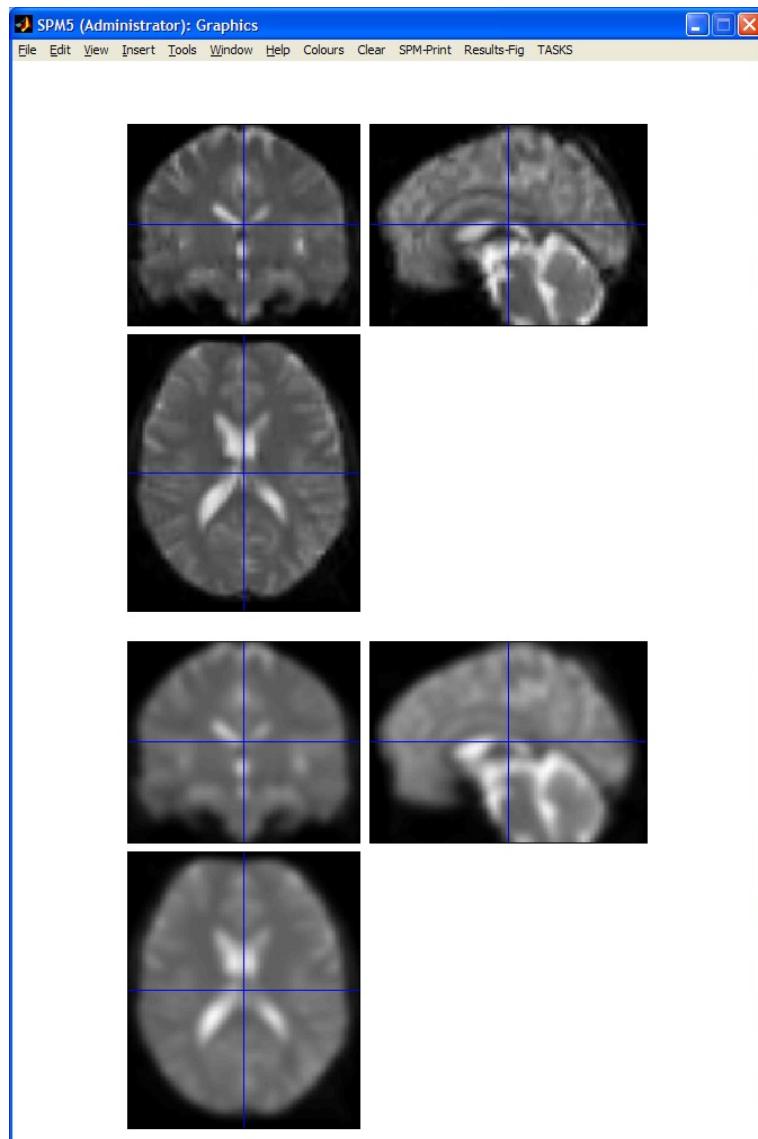


Figure 29.7: Functional image (top) and 6mm-smoothed functional image (bottom). These images were obtained using SPM's 'CheckReg' facility.

## 29.2 Model specification, review and estimation

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.

Press the ‘Specify 1st-level’ button. This will call up the specification of an fMRI specification job in the graphics window. Then

- Open the fMRI model specification option
- Open the ‘Timing parameters’ option
- Highlight ‘Units for design’ and select ‘Scans’
- Highlight ‘Interscan interval’ and enter 7
- 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 `swrfM00223_016.img - *_099.img`. These can be selected easily using the `^s.*` filter, and select all (provided you have moved the scans 4 to 15 into a different directory). Then press ‘Done’.
- Highlight ‘Condition’ and select ‘New condition’
- Open the newly created ‘Condition’ option. Highlight ‘Name’ and enter ‘active’. 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 ‘RUN’

SPM will then write an `SPM.mat` file to the `DIR/classical` directory. It will also plot the design matrix, as shown in Figure 29.8.

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 29.8. If you select ‘Explore’ then ‘Session 1’ then ‘active’, SPM will produce the plots shown in Figure 29.9.

If you select the second item on the ‘Design’ tab, ‘Design Orthogonality’, SPM will produce the plot shown in Figure 29.10. 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 29.10 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.

### 29.2.1 Estimate

Press the ‘Estimate’ button. This will call up the specification of an fMRI estimation job in the graphics window. Then

- Open the ‘fMRI model estimation’ option
- Highlight the ‘Select SPM.mat’ option and then choose the `SPM.mat` file saved in the classical subdirectory
- Save the job as `estimate.job` and press Run

SPM will write a number of files into the selected directory including an `SPM.mat` file.

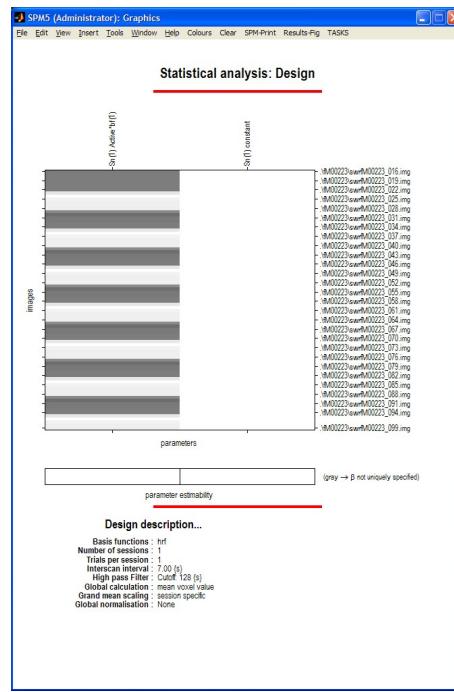


Figure 29.8: *Design matrix*. The filenames on the right-hand side of the design matrix indicate the scan associated with each row.

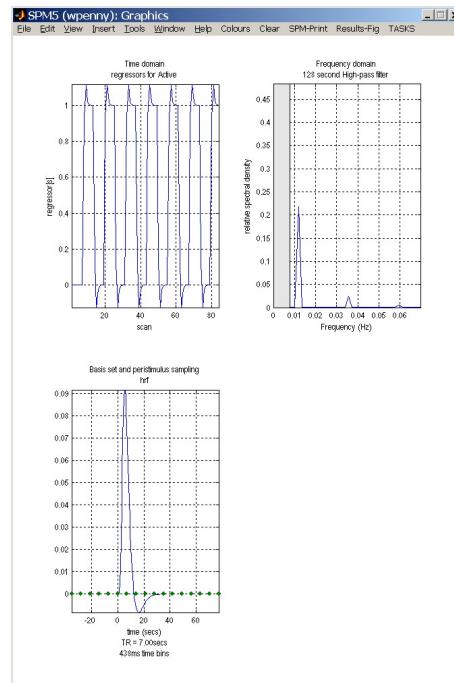


Figure 29.9: *Exploring the design matrix in Figure 29.8*. This shows the time series of the 'active' regressor (top left), a frequency domain plot of the active regressor (top right) and the basis function used to convert assumed neuronal activity into hemodynamic activity. In this model we used the default option - the canonical basis function. The frequency domain plot shows that the frequency content of the 'active' 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).

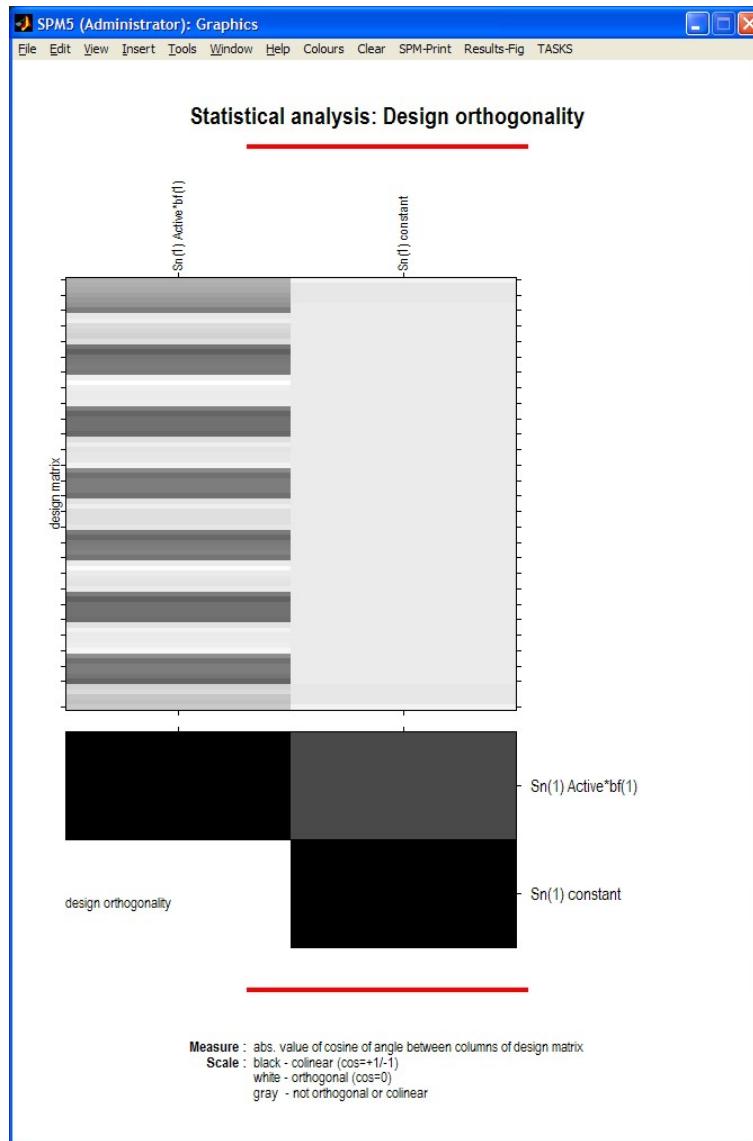


Figure 29.10: *Design Orthogonality*. The description above the first column in the design matrix  $Sn(1)Active*bf(1)$  means that this column refers to the first session of data (in this analysis there is only 1 session), the name of this condition/trial is ‘Active’ and the trial information has been convolved with the first basis function (the canonical hemodynamic response). The constant regressor for session 1 is referred to as  $Sn(1)Constant$ . The orthogonality matrix at the bottom indicates a degree of collinearity between regressors.

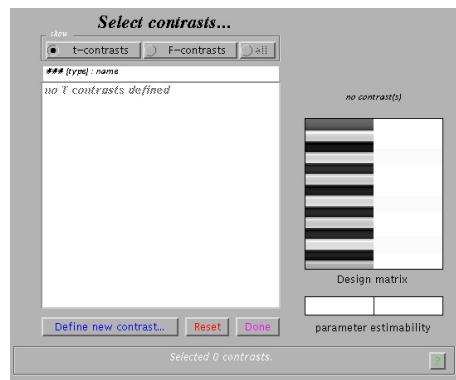


Figure 29.11: The contrast manager

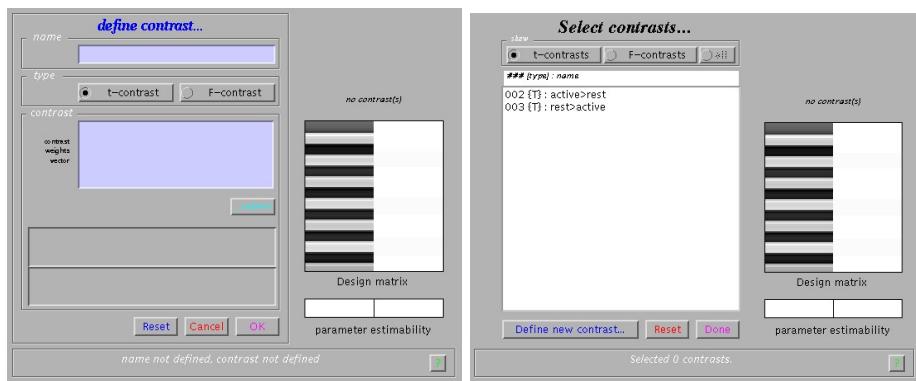


Figure 29.12: Left: A contrast is entered by specifying the numeric values in the lower window and the name in the upper window. Right: After contrasts have been specified they can be selected.

## 29.3 Inference

After estimation:

- Press ‘Results’
- Select the **SPM.mat** file created in the last section

This will invoke the contrast manager.

### 29.3.1 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 active condition (i.e., a one-sided t-test) can be specified (in this example) as ‘1’ (active > rest) and ‘-1’ (rest > active). SPM will accept correct 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., ‘active > rest’
- Press ‘Done’

### 29.3.2 Masking

You will then be prompted with

- *Mask with other contrast ? [Yes/No]*
- Specify No.

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.

### 29.3.3 Thresholds

You will then be prompted with

- *Title for comparison ?*
- Enter eg. 'active > rest'
- *Corrected height threshold ? [Yes/No]*
- Enter Yes.
- *p value adjustment to control: [FWE/FDR/none]*
- Select FWE
- *p value(family-wise error)*
- 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 1 in 20 SPMs contains a false positive 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.

The final option here is False Discovery Rate (FDR). If you set this at 0.1, this means that of all the discoveries you make (ie. above threshold voxels that appear in the SPM) 10% of them are likely to be false.

You will then be prompted with

- *Extent Threshold {voxels} [0]*
- Accept the default value, 0

Entering a value  $v$  here will produce SPMs with clusters containing at least  $v$  voxels. SPM will then produce the SPM shown in Figure 29.13.

- Select 'Define new contrast'

### 29.3.4 Files

A number of files are written to the working directory at this time. Images containing weighted parameter estimates are saved as `con-0002.hdr/img`, `con-0003.hdr/img`, etc. in the working directory. Images of T-statistics are saved as `spmT-0002.hdr/img`, `spmT-0003.hdr/img` etc., also in the working directory.

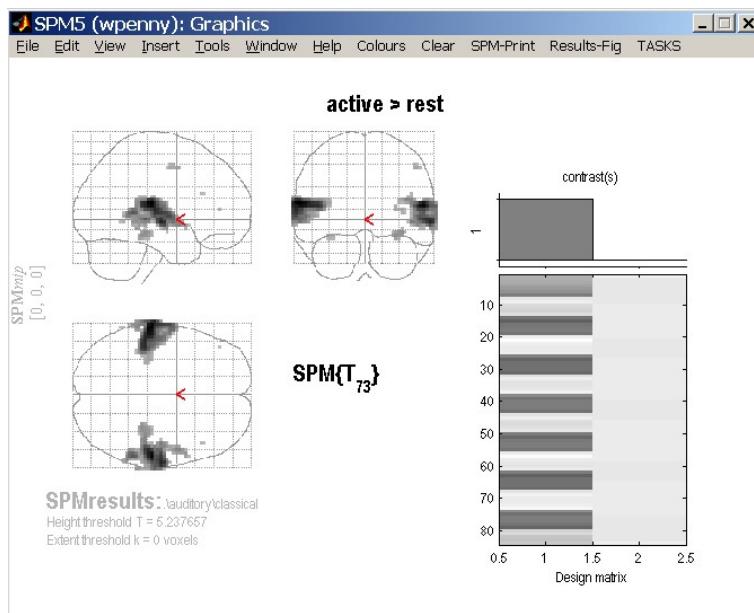


Figure 29.13: *SPM showing bilateral activation of auditory cortex.*

### 29.3.5 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: R-clicking in the MIP will activate a pulldown menu, L-clicking on the red cursor will allow it to be dragged to a new position.

### 29.3.6 Design matrix

SPM also displays the design matrix with the selected contrast. The design matrix is also surfable: R-clicking will show parameter names, L-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.

### 29.3.7 Statistical tables

To get a summary of local maxima, press the 'volume' 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 29.15

The columns in volume table show, from right to left:

- x, y, z (mm): coordinates in MNI space for each maximum
- voxel-level: the chance (p) of finding (under the null hypothesis) a voxel 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 / 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

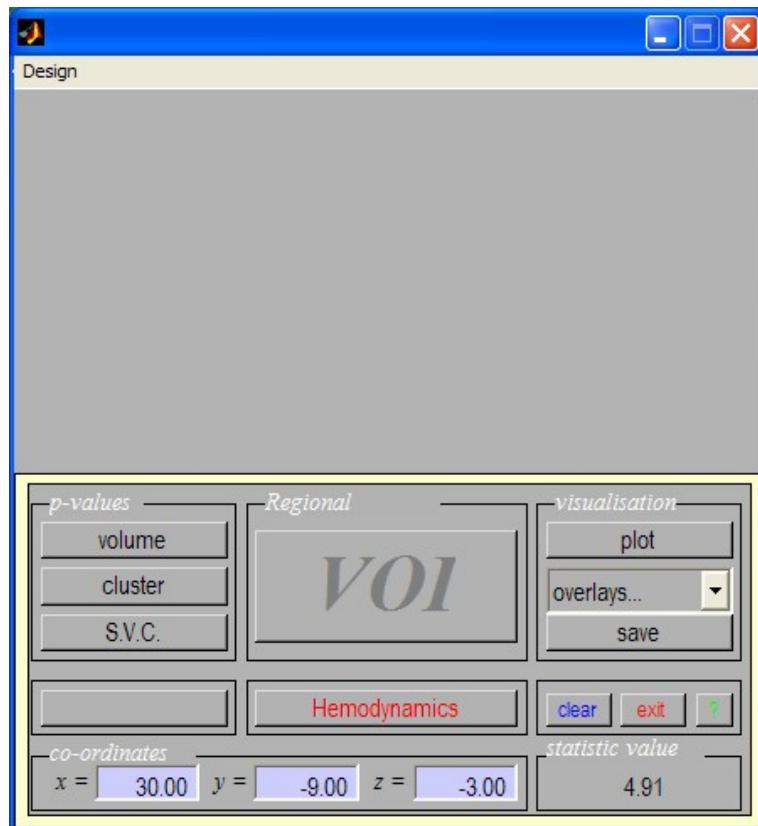


Figure 29.14: *SPM's interactive window during results assessment. The 'p-values' section is used to produce tables of statistical information. The visualisation section is used to plot responses at a voxel or to visualise activations overlaid on anatomical images. The 'Regional' section, i.e. the VOI button, is used to extract data for subsequent analyses such as assessment of PsychoPhysiological Interactions (PPIs) or Dynamic Causal Models (DCMs).*

Statistics: <i>p</i> -values adjusted for search volume									
set-level		cluster-level		voxel-level				x,y,z [mm]	
<i>p</i>	<i>c</i>	<i>p</i> connected	<i>k</i> <sub>E</sub>	<i>p</i> uncorrected	<i>p</i> <sub>FWE-corr</sub>	<i>p</i> <sub>FDR-corr</sub>	<i>T</i>	$ Z $	<i>p</i> uncorrected
0.000	9	0.000	514	0.000	0.000	0.000	14.19	Inf	0.000
					0.000	0.000	11.86	Inf	0.000
					0.000	0.000	9.54	7.66	0.000
					0.000	0.000	13.62	Inf	0.000
					0.000	0.000	12.24	Inf	0.000
					0.000	0.000	9.82	7.80	0.000
					0.000	0.000	7.33	6.32	0.000
					0.001	0.000	6.32	5.63	0.000
					0.001	0.000	6.22	5.55	0.000
					0.002	0.000	6.07	5.44	0.000
					0.002	0.000	6.02	5.41	0.000
					0.006	0.000	5.76	5.22	0.000
					0.022	0.000	5.45	4.97	0.000
					0.047	0.000	5.25	4.82	0.000
0.015	1	0.166							-45 -42 9

table shows 3 local maxima more than 8.0mm apart

Height threshold:  $T = 5.24$ ,  $p = 0.000$  (0.050)  
 Extent threshold:  $k = 0$  voxels,  $p = 1.000$  (0.050)  
 Expected voxels per cluster,  $\ll > = 0.553$   
 Expected number of clusters,  $\ll > = 0.09$   
 Expected false discovery rate,  $\ll > = 0.00$

Degrees of freedom = [1.0, 73.0]  
 FWHM = 8.9 8.9 7.9 mm; 3.0 3.0 2.6 (voxels);  
 Volume: 1787508; 66204 voxels; 2573.5 resels  
 Voxel size: 3.0 3.0 3.0 mm; (resel = 22.95 voxels)

Figure 29.15: Volume table for ‘active > rest’ effect. This table of values was created by pressing the ‘Results-Fig’ tab at the top of the graphics window and then pressing the ‘Volume’ button. This displays the table of results in a separate window.

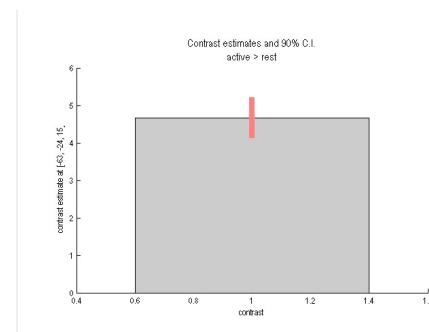
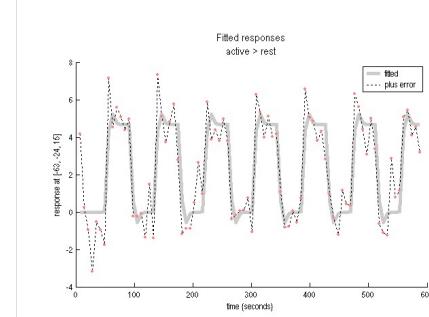
- The table is surfable: clicking a row of cluster coordinates will move the pointer in the MIP to that cluster, clicking other numbers will display the exact value in the Matlab window (e.g. 0.000 = 6.1971e-07).
- To inspect a specific cluster (e.g., in this example data set, the R auditory cortex), either move the cursor in the MIP (by L-clicking and dragging the cursor, or R-clicking the MIP background which will activate a pulldown menu).
- 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, elect the relevant cluster in the MIP and then press the ‘cluster’ button in the p-values section of the interactive window. This will show coordinates and voxel-level statistics for local maxima (>4mm apart) in the selected cluster. This table is also surfable.

### 29.3.8 Plotting responses at a voxel

A voxel can be chosen with co-ordinates 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:

- Contrast estimates and 90% CI: SPM will prompt for a specific contrast (e.g., active>rest). The plot will show effect size and 90% confidence intervals. See eg. Figure 29.16
- 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 29.17

Figure 29.16: *Estimated effect size.*Figure 29.17: *Fitted responses.*

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

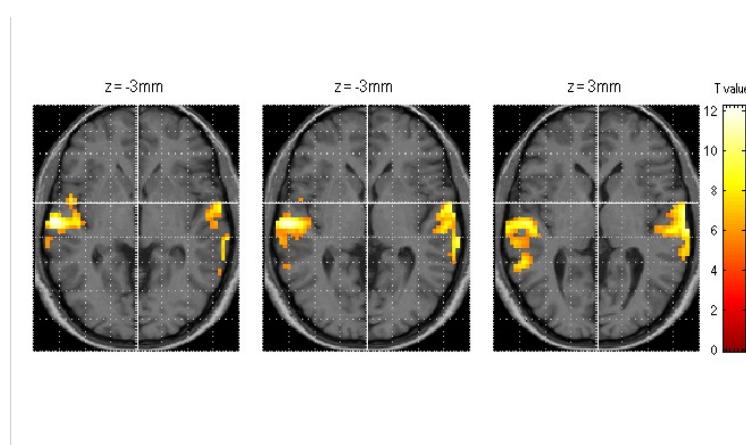
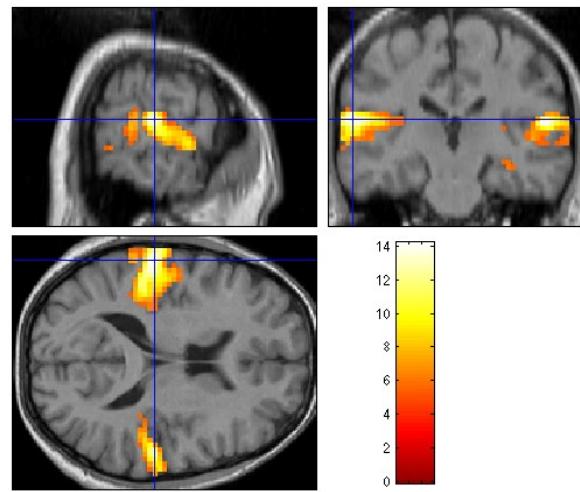
1. Fitted response and PSTH (peri-stimulus time histogram): plots mean regressor(s) (ie. averaged over session) and mean signal +/- SE for each peri-stimulus time bin.
2. Fitted response and 90% CI: plots mean regressor(s) along with a 90% confidence interval.
3. Fitted response and adjusted data: plots 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.

### 29.3.9 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 three options

Figure 29.18: *Slices*.Figure 29.19: *Sections*.

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-template.man`) or an individual T1/mean EPI image for single-subject analyses.
2. Sections: overlay on three intersecting (sagittal, coronal, transaxial) slices. These renderings are surfable: clicking the images will move the crosshair.
3. Render: overlay on a volume rendered brain, with options for using a smoothed brain, and old (left) and new (right) style rendering.

Renderings can be saved as `filename.img` and `filename.hdr` in the working directory by using the *write filtered* option. In Figures 29.18, 29.19 and 29.20 the ‘active > rest’ activation has been superimposed on the spatially normalised, bias-corrected anatomical image `wmsM00223_002.img` created earlier.

For the ‘Render’ option we first created a rendering for this subject. This was implemented by

- Selecting ‘Xtract Surface’ from the ‘Render’ pulldown menu

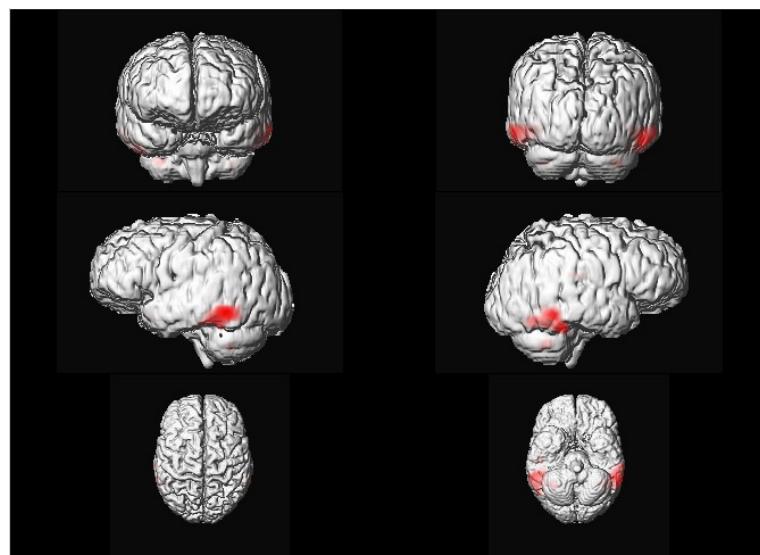


Figure 29.20: *Render*.

- Selecting the gray and white matter images `c1sM00223_002.img` and `c2sM00223_002.img` created earlier.
- Saving the results using the default options (Rendering and Surface)

SPM plots the rendered anatomical image in the graphics window and saves it as `render_c1sM00223_002.img`. The surface image is saved as `surf_c1sM00223_002.img`.

### 29.3.10 Miscellaneous

Other options (in the results controls panel):

- clear: clears lower subpanel of Graphics window
- exit: exits the results section
- ? : launches spm-results-ui help

## 29.4 Bayesian analysis

### 29.4.1 Specification

Press the ‘Specify 1st-level’ button. This will call up an fMRI specification job in the graphics window. Then

- Open the fMRI model specification option
- Load the ‘specify.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 `~w.*` filter and ‘select all’ option available from a right mouse click in the SPM file selector (top right window)<sup>4</sup>. The Bayesian analysis

---

<sup>4</sup>Remember not to select the first 12 scans, scans 4 to 15, as these may contain T1 effects. This can be done during selection or by first moving the files to a different directory.

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/classical directory).
- Save the job as `specify_bayesian.mat` and press ‘Run’

#### 29.4.2 Estimation

Press the ‘Estimate’ button. This will call up the specification of an fMRI estimation job in the graphics window. Then

- Open the ‘fMRI model estimation’ option
- Highlight the ‘Select SPM.mat’ option and then choose the SPM.mat file saved in the DIR/bayesian directory
- Highlight ‘Method’ and select the ‘Choose Bayesian 1st-level’ option.
- Open the newly created ‘Bayesian 1st-level’ option, highlight ‘AR model order’ and select 0. This data set has a TR=7s, so is unlikely to have temporally autocorrelated errors.
- Save the job as `estimate_bayesian.job` and press ‘Run’.

SPM will write a number of files into the output directory including

- An `SPM.mat` file.
- Images of estimated regression coefficients `Cbeta_0001.img` and `Cbeta_0002.img`. 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_0001.img` and `SDbeta_0002.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.

#### 29.4.3 Inference

After estimation:

- Press ‘Results’
- Select the `SPM.mat` file created in the last section
- Select ‘Define new contrast’
- Enter the name ‘active > rest’
- Enter the value ‘1’, press ‘Submit’, ‘OK’, ‘Done’
- *Mask with other contrast ? [Yes/No]*
- Specify No
- Title for comparison, accept the default
- *Effect size threshold for PPM*

- Enter the value 2
- *Posterior probability threshold for PPM*
- Enter the value 0.99
- *Extent threshold [0]*
- Accept the default value
- *Plot effect size [Yes/No]*
- Select the default ‘Yes’

SPM will then plot a map of effect sizes at voxels where it is 99% sure that the effect size is greater than 2% of the global mean. This is a large activation. 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 29.21

It is also possible to look for regions where responses in the active condition are different to those at rest. Active responses could be greater or smaller.

- Press ‘Results’
- Select the **SPM.mat** file created in the last section
- Select ‘Define new contrast’ and highlight the ‘F’ radio button
- Enter the name ‘active != rest’
- Enter the value ‘1’, press ‘Submit’, ‘OK’, ‘Done’
- *Mask with other contrast ? [Yes/No]*
- Specify No
- Title for comparison, accept the default
- *Posterior probability threshold for PPM*
- Accept the default value<sup>5</sup>
- *Extent threshold [0]*
- Accept the default value,0
- *Plot effect size [Yes/No]*
- Select the default ‘Yes’.

SPM will then plot a map of  $\chi^2$  statistic values at above threshold voxels. 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 29.22

When you revisit the contrast manager this contrast will be referred to as a ‘P’ contrast, rather than an ‘F’ contrast. This indicates that Bayes rule is used to make the inference. To indicate that we are testing a two-sided effect it is advisable to make this clear when naming the contrast (as we have done with the label ‘active != rest’).

---

<sup>5</sup>The default PPM threshold is set to  $1 - 1/S$  where S is the number of voxels in the volume being analysed. The rationale for this is that inference is based on an approximate posterior distribution,  $Q$ , which factorises across voxels. The approximate posterior is chosen to best match the true posterior in the sense of KL-divergence. Given the factorisation in  $Q$ , the expected number of false positives in the PPM is 1.

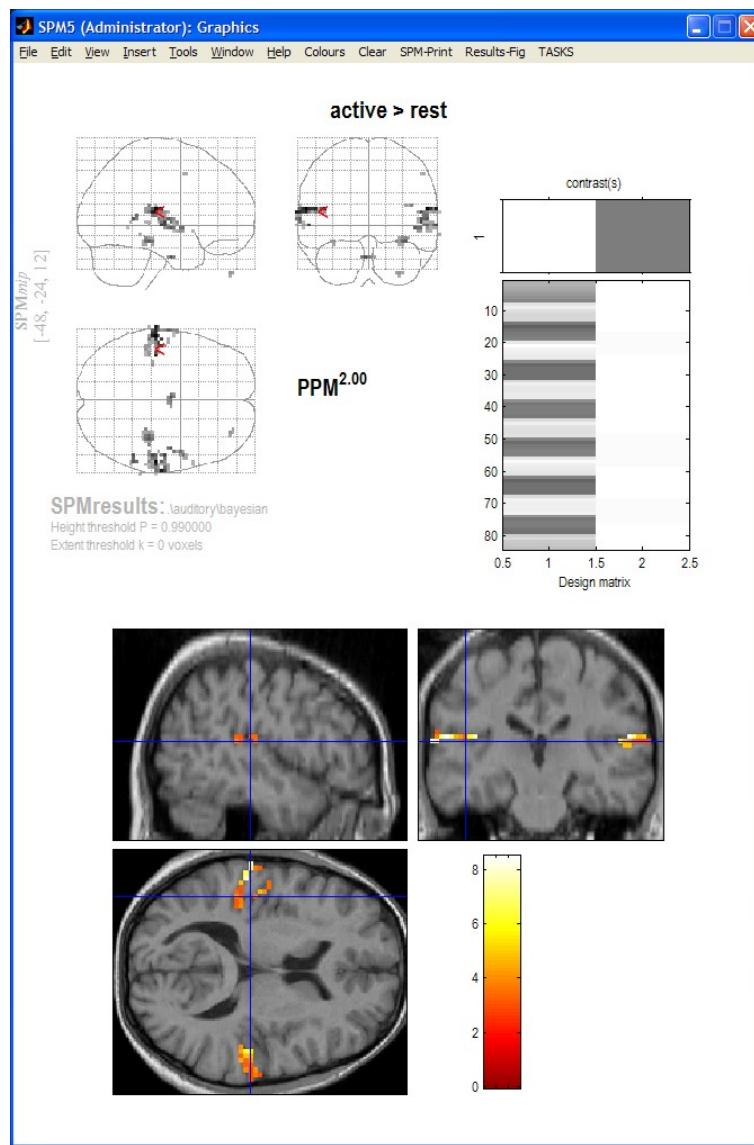


Figure 29.21: **Bayesian analysis:** MIP and overlay of effect sizes at voxels where SPM is 99% sure that the effect size is greater than 2% of the global mean.

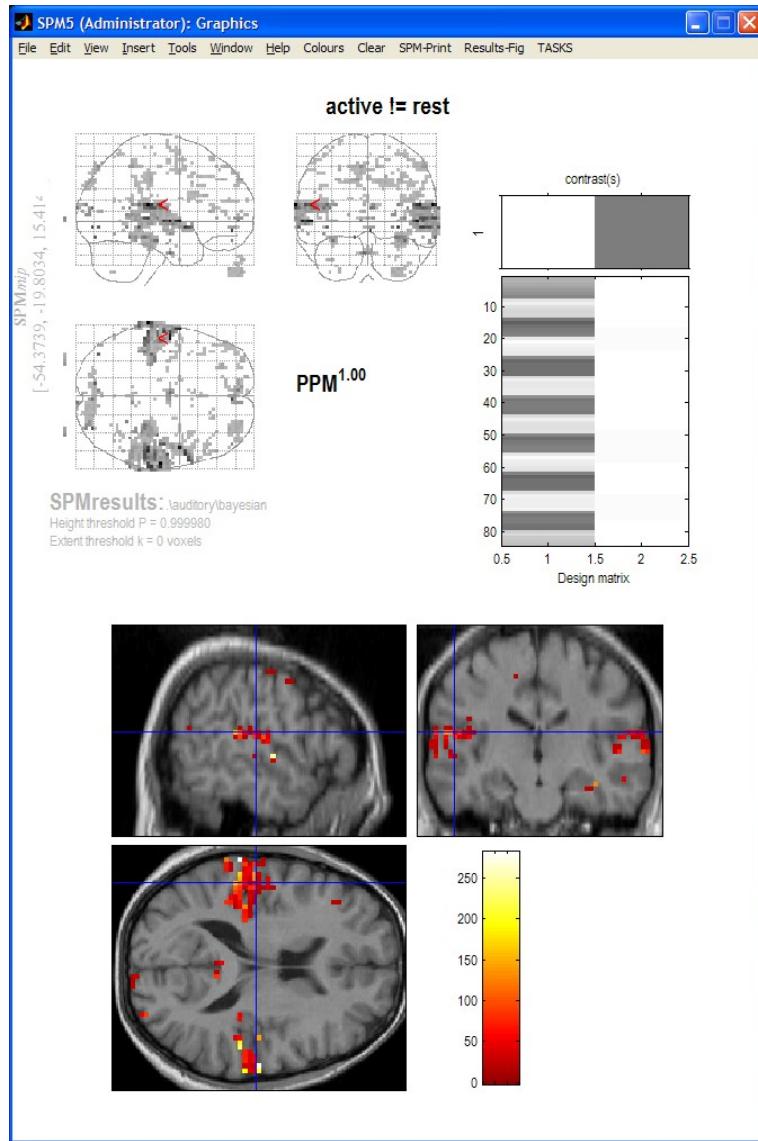


Figure 29.22: **Two-sided Bayesian analysis:** MIP and overlay of  $\chi^2$  statistic values at above threshold voxels. This shows regions where activity is different between active and rest conditions, whether positive or negative.

# Chapter 30

## Face data

As a third and more sophisticated example, consider the data from a repetition priming experiment performed using event-related fMRI. Briefly, this is a 2x2 factorial study with factors ‘fame’ and ‘repetition’ where famous and non-famous faces were presented twice against a checkerboard baseline (for more details, see [35]). 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 30.1 and 30.2.

Images were acquired using continuous Echo-Planar Imaging (EPI) with TE=40ms, TR=2s and 24 descending slices ( $64 \times 64 \times 3 \text{ mm}^3$ ), 3mm thick with a 1.5mm gap. The data archive is available from [http://www.fil.ion.ucl.ac.uk/spm/data/face\\_rep\\_SPM5.html](http://www.fil.ion.ucl.ac.uk/spm/data/face_rep_SPM5.html). This contains 351 Analyse format functional images `sM03953_0005_*.``img` of dimension  $64 \times 64 \times 24$  with  $3 \text{ mm} \times 3 \text{ mm} \times 4.5 \text{ mm}$  voxels. A structural image is also provided `sM03953_0007.``img` also in Analyse format.

To analyse the data, first create a new directory DIR  
eg. `c:\home\wpenny\fmri_analysis\face-rep\all`, 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 ‘command’ 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.

### 30.1 Spatial pre-processing

#### 30.1.1 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 if necessary’ from the ‘Effects’ tab at the top of the graphics window.

#### 30.1.2 Realignment

Under the spatial pre-processing section of the SPM base window select ‘Realign’ from the ‘Realign’ pulldown menu. This will call up a realignment job specification in the graphics window. Then

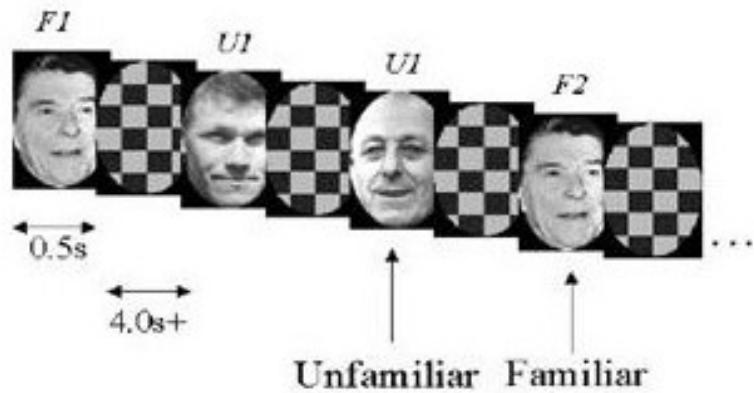


Figure 30.1: *Face repetition paradigm*. There were 2 presentations of 26 Famous and 26 Nonfamous Greyscale photographs, for 0.5s each, randomly intermixed. The minimal Stimulus Onset Asynchrony (SOA)=4.5s, with probability 2/3 (ie 1/3 null events). The subject made one of two right finger key presses denoting whether or not the subject thought the face was famous.

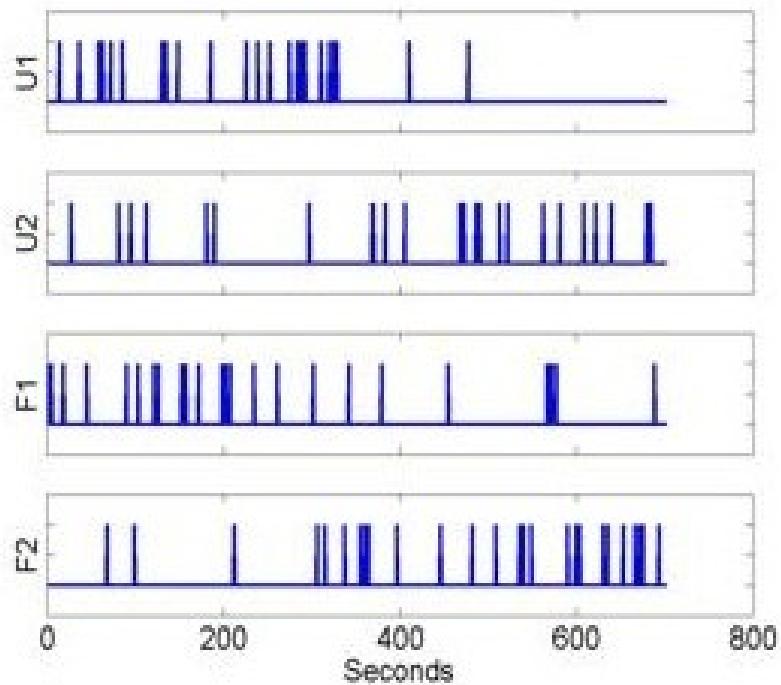


Figure 30.2: *Time series of events*.

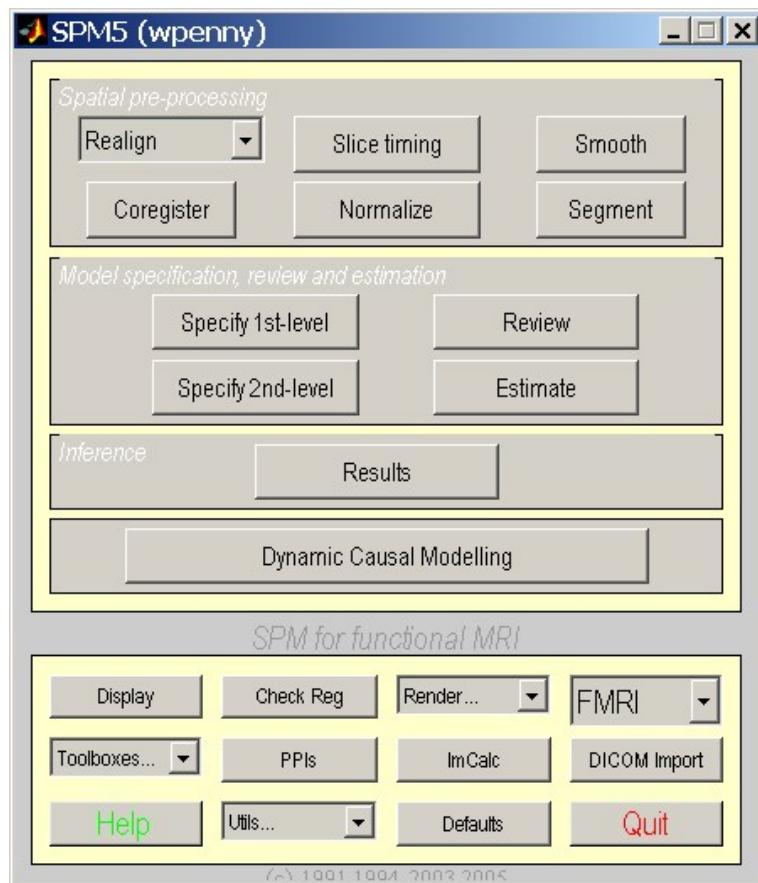
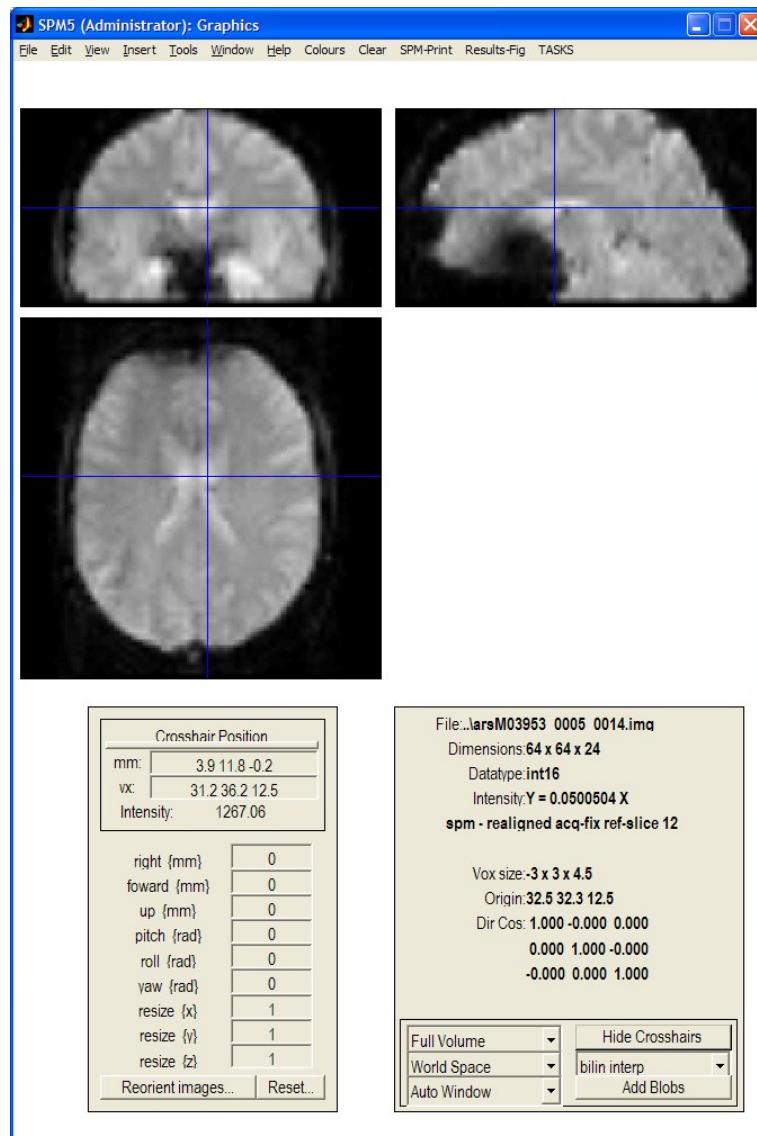


Figure 30.3: The SPM base window comprises three sections i) spatial pre-processing, (ii) model specification, review and estimation and (iii) inference.

Figure 30.4: *Signal dropout in EPI images.*

- Select ‘New Realign:Estimate and Reslice’
- Open the newly created ‘Realign:Estimate and Reslice’ option.
- 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`.
- Save the job file as eg. `DIR/jobs/realign.mat`.
- Press the RUN button in the graphics window.

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 30.5. 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. To prepare for this copy the file into the `DIR\jobs\` directory and rename it `movepars.txt`. This allows movements effects to be discounted when looking for brain activations.

SPM will also create a mean image eg. `meansM03953_0005_0006.img` which will be used in the next step of spatial processing - coregistration.

### 30.1.3 Slice timing correction

Press the ‘Slice timing’ button. This will call up the specification of a slice timing job in the graphics 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].

- Open the ‘Slice Timing’ option
- 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 ‘Run’

SPM will write slice-time corrected files with the prefix ‘a’ in the functional data directory.

### 30.1.4 Coregistration

Press the ‘Coreg’ button. This will call up the specification of a coregistration job in the graphics window.

- Select New ”Coreg:Estimate”
- Double-click on the newly created Coreg:Estimate
- Highlight ‘Reference Image’ and then select the mean functional image `meansM03953_0005_0006.img`

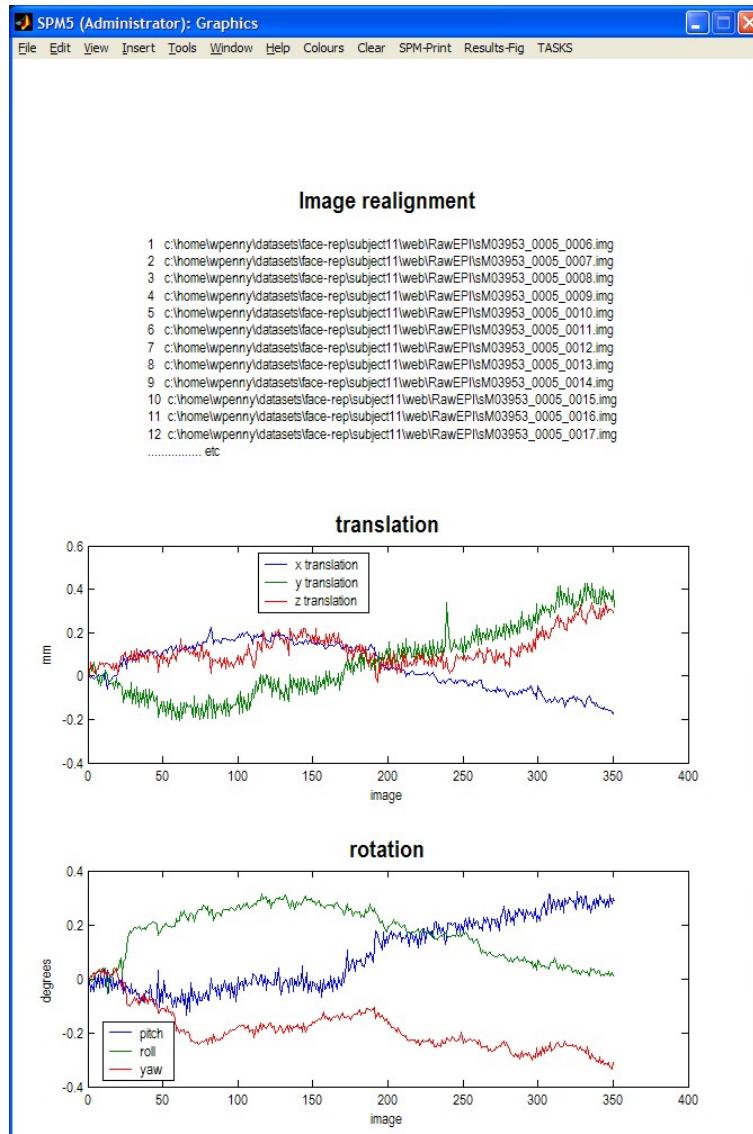


Figure 30.5: *Realignment of face data. Movement less than the size of a voxel, which for this data set is 3mm, is not considered problematic.*

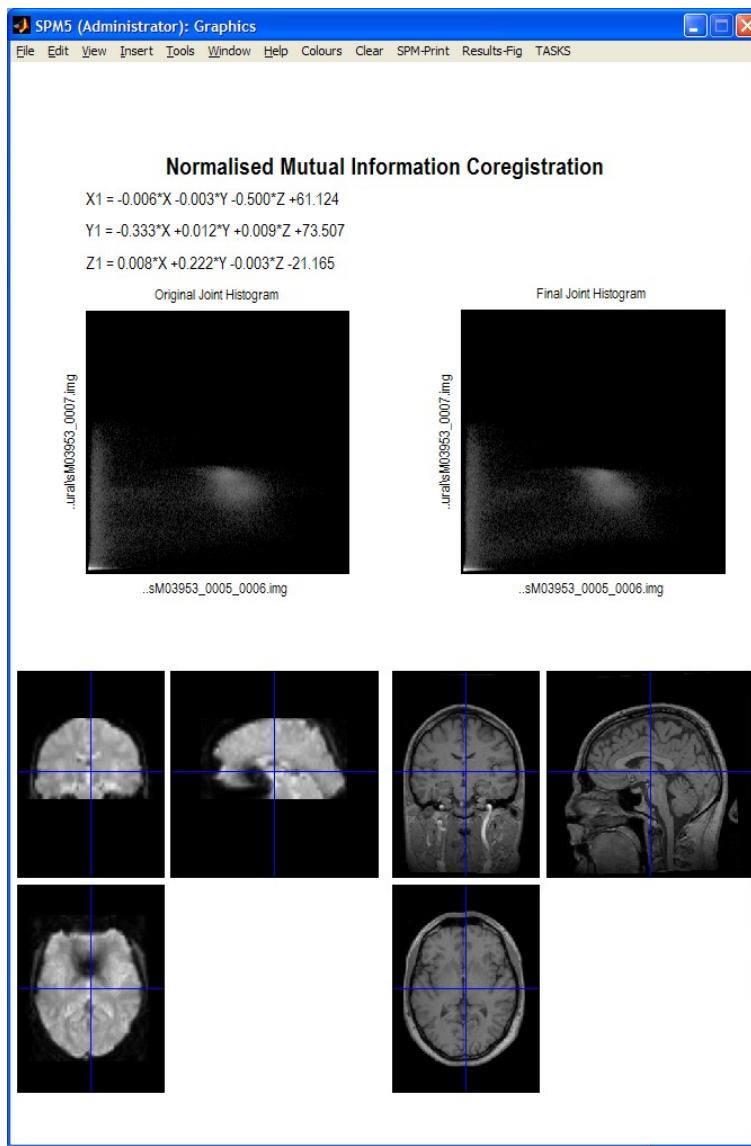


Figure 30.6: *Mutual Information Coregistration of Face data.*

- 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 RUN

SPM will then implement a coregistration between the structural and functional data that maximises the mutual information. The image in figure 30.6 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.img`.

### 30.1.5 Segmentation

Press the ‘Segment’ button. This will call up the specification of a segmentation job in the graphics window. Highlight the Data field and then select the subjects coregistered anatomical image eg. `sM03953_0007.img`. 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. SPM will create, by default, gray and white matter images and bias-field corrected structural image. These can be

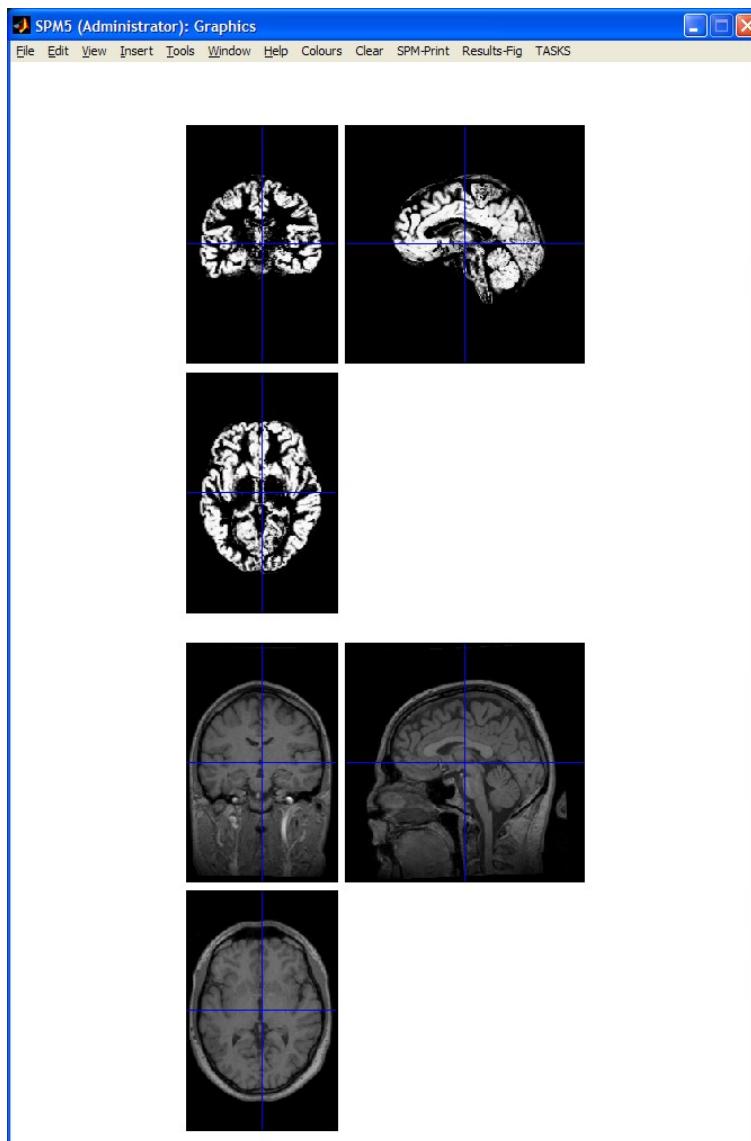


Figure 30.7: Gray matter (top) produced by segmentation of structural image (below).

viewed using the CheckReg facility as described in the previous section. Figure 30.7 shows the gray matter image, `c1sM03953_0007.img`, along with the original structural.<sup>1</sup>

SPM will also write a spatial normalisation eg. `sM03953_0007_seg_sn.mat` file in the original structural directory. This will be used in the next section to normalise the functional data.

### 30.1.6 Normalize

Press the ‘Normalize’ button. This will call up the specification of a normalise job in the graphics window.

- Select a New ”Normalise:Write” field. This allows previously estimated warps to be applied to a series of images.
- Highlight ‘Data’, select New ”Subject”

---

<sup>1</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.

- Open ‘Subject’, highlight ‘Parameter File’ and select the `sM03953_0007_seg_sn.mat` 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>2</sup>
- Press ‘Save’, save the job as `normalise.mat` and then press ‘Run’.

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>3</sup> you will also need to apply the spatial normalisation parameters to their (bias-corrected) anatomical image. To do this

- Press ‘Normalise’, select New ”Normalise:Write”
- Open ‘Normalise: Write’, highlight ‘Data’, select New ”Subject”
- Open ‘Subject’
- Highlight ‘Parameter File’, select the `sM03953_0007_seg_sn.mat` file that you created in the previous section, press ‘Done’.
- Highlight ‘Images to Write’, select the bias-corrected structural eg. `msM03953_0007.img`, 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’.

### 30.1.7 Smoothing

Press the ‘Smooth’ button<sup>4</sup>. This will call up the specification of a smooth job in the graphics window.

- Open ‘Smooth’, 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’.

This will smooth the data by (the default) 8mm in each direction, the default smoothing kernel width.

---

<sup>2</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>3</sup>Beginners may wish to skip this step, and instead just superimpose functional activations on an ‘canonical structural image’.

<sup>4</sup>The smoothing step is unnecessary if you are only interested in Bayesian analysis of your functional data.

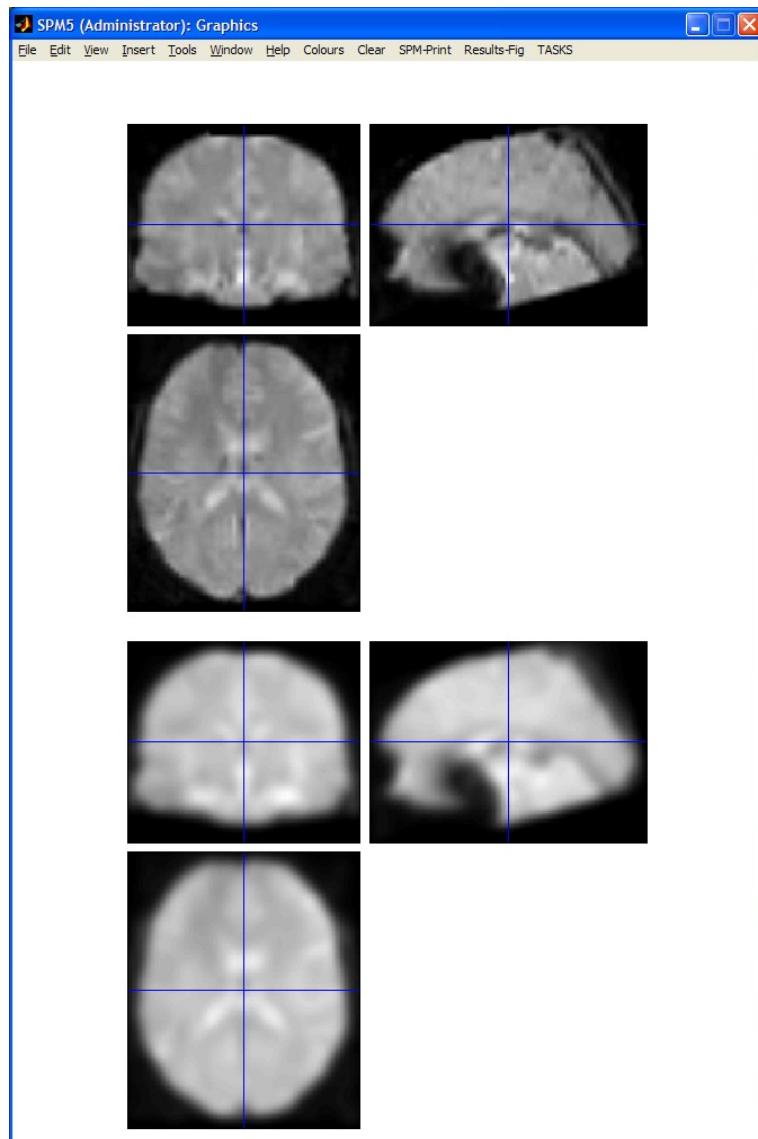


Figure 30.8: Functional image (top) and 8mm-smoothed functional image (bottom). These images were plotted using SPM's 'CheckReg' facility.

## 30.2 Modelling categorical responses

Before setting up the design matrix we must first load the Stimulus Onsets Times (SOTs) and movement parameters 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>5</sup>

- At the matlab command prompt type ‘load sots’
- Then type ‘load movepars.txt’

Now press the ‘Specify 1st-level’ button. This will call up the specification of a fMRI specification job in the graphics window. Then

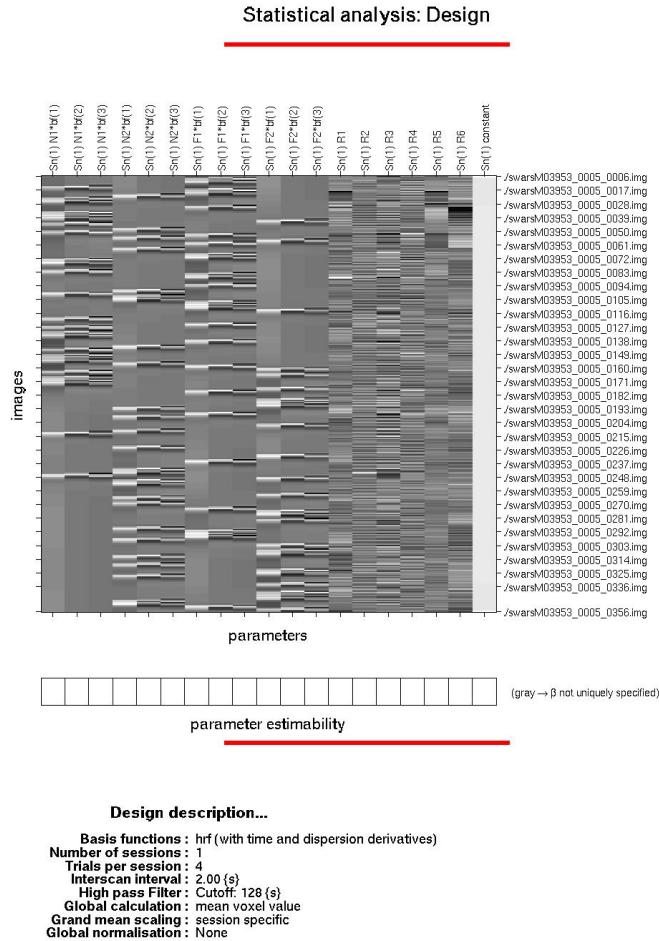
- Open the fMRI model specification option
- Open the ‘Timing parameters’ option
- Highlight ‘Units for design’ and select ‘Scans’
- Highlight ‘Interscan interval’ and enter 2
- Highlight ‘Microtime resolution’ and enter 24
- Highlight ‘Microtime onset’ and enter 12. These last two options make the creating 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’. Then open the newly created ‘Subject/Session’ option.
- 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>6</sup>
- Open the newly created ‘Condition’ option. Highlight ‘Name’ and enter ‘N1’. Highlight ‘Onsets’ and enter ‘`sot{1}`’. Highlight ‘Durations’ and enter 0.
- Highlight ‘Condition’ 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 ‘Condition’ 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 ‘Condition’ 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 `movepars.txt` file.<sup>7</sup>

---

<sup>5</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>6</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>7</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 30.9: *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>8</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 ‘RUN’

SPM will then write an `SPM.mat` file to the DIR/categorical directory. It will also plot the design matrix, as shown in Figure 30.9.

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

<sup>8</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.

window clicking on which produces a pulldown menu. If you select the first item ‘Design Matrix’ SPM will produce the image shown in Figure 30.9. If you select ‘Explore’ then ‘Session 1’ then ‘N1’, SPM will produce the plots shown in Figure 30.10.

### 30.2.1 Estimate

Press the ‘Estimate’ button. This will call up the specification of an fMRI estimation job in the graphics window. Then

- Open the ‘fMRI model estimation’ option
- 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

SPM will write a number of files into the selected directory including an `SPM.mat` file.

### 30.2.2 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 30.11.

- 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’.
- *Mask with other contrast ? [Yes/No]*
- Specify No.
- *Title for comparison ?*
- Enter ‘Canonical HRF: Faces > Baseline’
- *p value adjustment to control: [FWE/FDR/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 30.12.

### 30.2.3 Statistical tables

To get a summary of local maxima, press the ‘Volume’ 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 30.12. The columns in volume table show, from right to left:

- x, y, z (mm): coordinates in MNI space for each maximum
- voxel-level: the chance (p) of finding (under the null hypothesis) a voxel with this or a greater height (T- or Z-statistic), corrected (FWE or FDR)/ uncorrected for search volume.

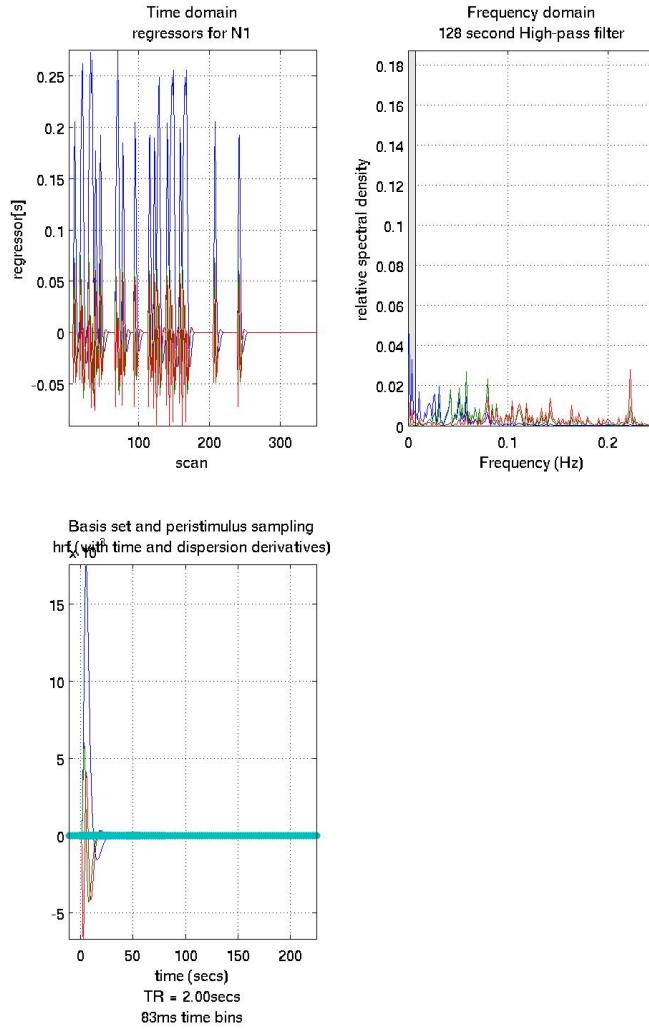


Figure 30.10: Exploring the design matrix in Figure 30.9. This shows the time series of the ‘active’ regressor (top left), the three basis functions used to convert assumed neuronal activity into hemodynamic activity (bottom left), and a frequency domain plot of the three regressors for the basis functions 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).

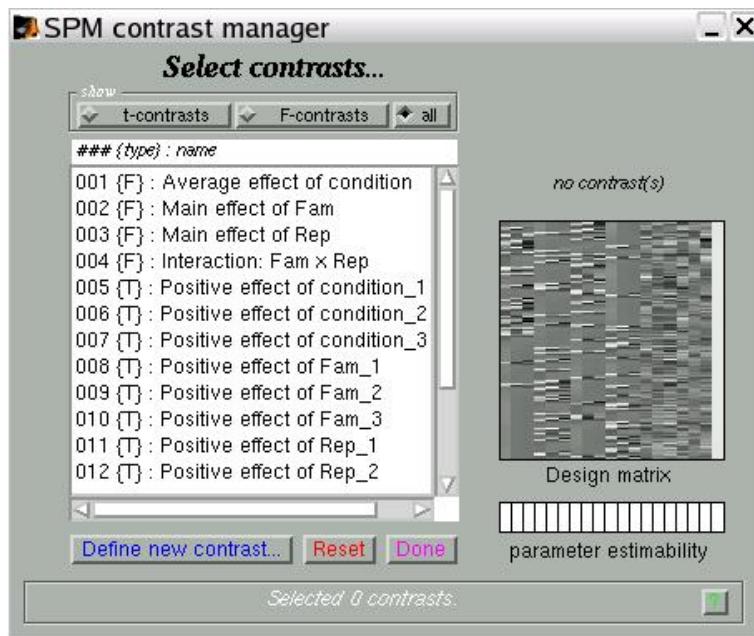


Figure 30.11: *Contrast Manager containing default contrasts for categorical design.*

- cluster-level: the chance ( $p$ ) of finding a cluster with this many(ke) or a greater number of voxels, corrected / uncorrected for search volume
- 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 -72 -12). You can view this activation on the subject’s normalised, attenuation-corrected structural ('wmsM03953\_0007.img), which gives best anatomical precision, or on the normalised mean functional (‘wmeansM03953\_0005\_0006.img), 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).

#### 30.2.4 F-contrasts

To assess the main effect of repeating faces, as characterised by both the hrf *and* its derivatives, . 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 30.13. Press ‘Done’.
- *Mask with other contrast ? [Yes/No]*
- Specify Yes.

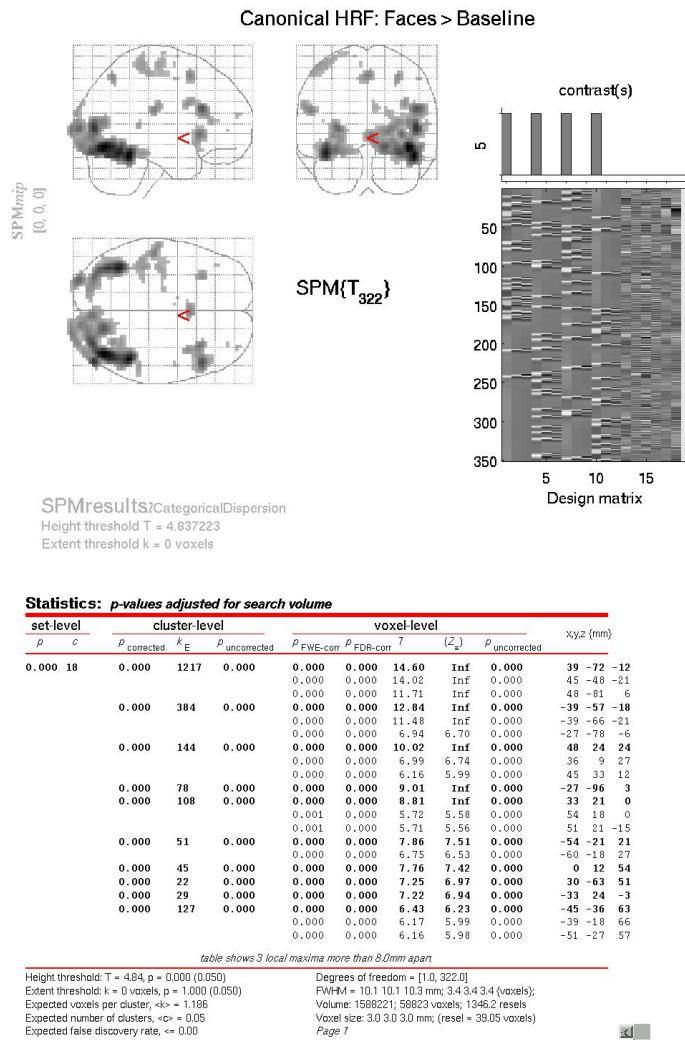


Figure 30.12: MIP and Volume table for Canonical HRF: Faces &gt; Baseline.

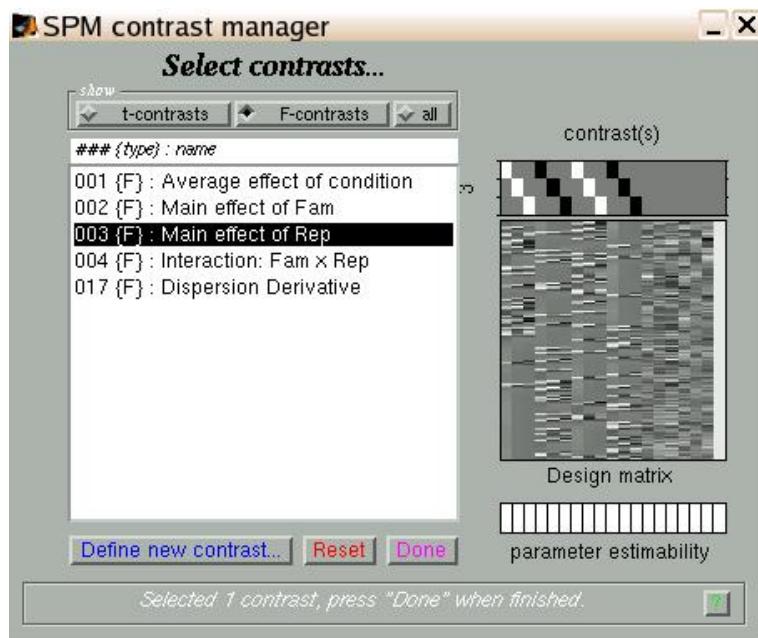


Figure 30.13: Contrast manager showing selection of the first contrast 'Main effect of Rep (repetition; F1 and N1 vs F2 and N2)'

- Select contrast 5 - Positive effect of condition 1 (the T-contrast of activation versus baseline, collapsed across conditions, that we evaluated above)
- *uncorrected mask p-value ?*
- Change to 0.001
- *nature of mask?*
- Select 'inclusive'
- *Title for comparison ?*
- Keep 'Main effect of Rep (masked with ...)'
- *p value adjustment to control: [FWE/FDR/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 30.14.

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). Only two small blobs will appear - one in right ventral temporal cortex (45 -60 -9).

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 Input window, and

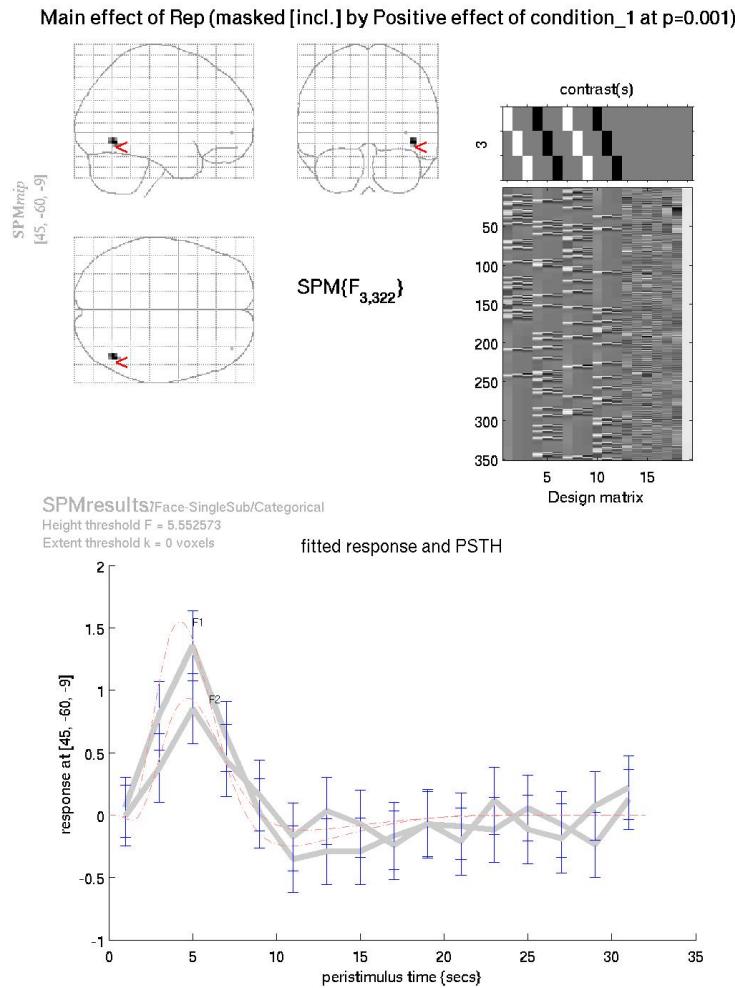


Figure 30.14: MIP for Main effect of Rep, masked inclusively with Canonical HRF: Faces > Baseline at  $p < .001$  uncorrected. Shown below are the best-fitting responses and peri-stimulus histograms (PSTH) for F1 and F2.

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 30.14.

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 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], 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.

### 30.2.5 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 'mask with other contrasts?' (no), 'title for comparison' (accept default), '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.img`)

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 30.15). (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.

## 30.3 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 sots`. 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 graphics window. Then

- Open the fMRI model specification option
- Press 'Load' and select the `categorical_spec.mat` job file you created earlier

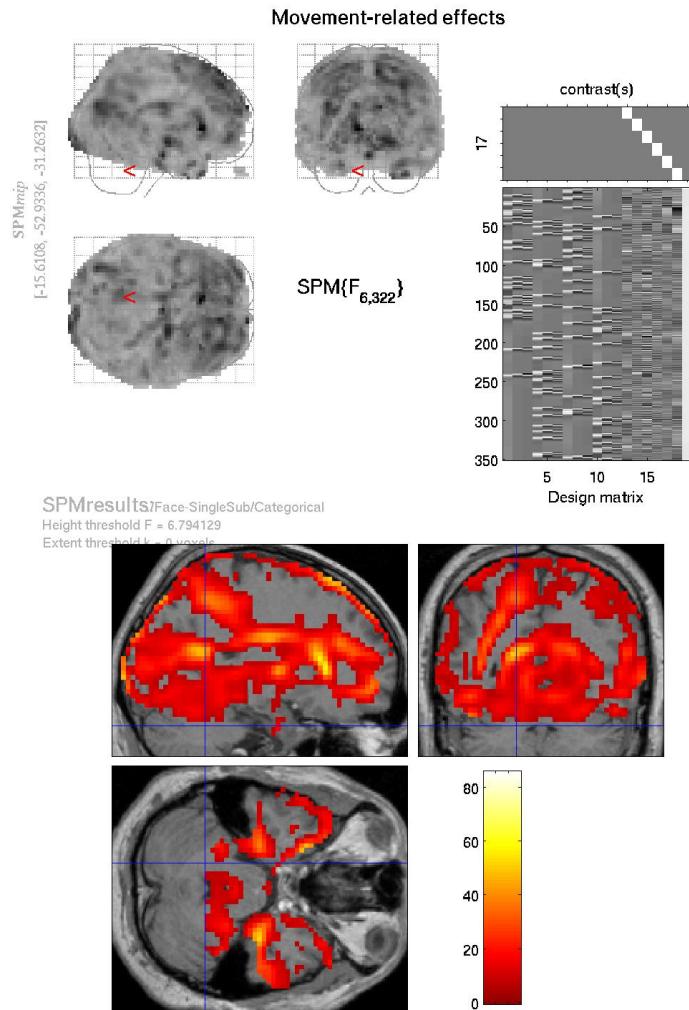


Figure 30.15: *Movement-related activations.* These spurious ‘activations’ are due to residual movement of the head during scanning. These effects occur at tissue boundaries and boundaries between brain and non-brain, as this is where contrast differences are greatest. Including these regressors in the design matrix means these effects cannot be falsely attributed to neuronal activity.

- Open ‘Conditions’ and then open the second ‘Condition’
- Highlight ‘Parametric Modulations’, select ‘New Parameter’ and open the newly created ‘Parameter’ option.
- 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’ and open the newly created ‘Parameter’ option.
- 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 ‘Run’

This should produce the design matrix shown in Figure 30.16.

### 30.3.1 Estimate

Press the ‘Estimate’ button. This will call up the specification of an fMRI estimation job in the graphics window. Then

- Open the ‘fMRI model estimation’ option
- 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 Run

SPM will write a number of files into the selected directory including an `SPM.mat` file.

### 30.3.2 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.
- *Mask with other contrast ? [Yes/No]*
- Specify Yes.
- Select the ‘Positive Effect of Condition 1’ T contrast
- Change to an 0.05 uncorrected mask p-value
- Nature of Mask ? inclusive
- *Title for comparison ?*

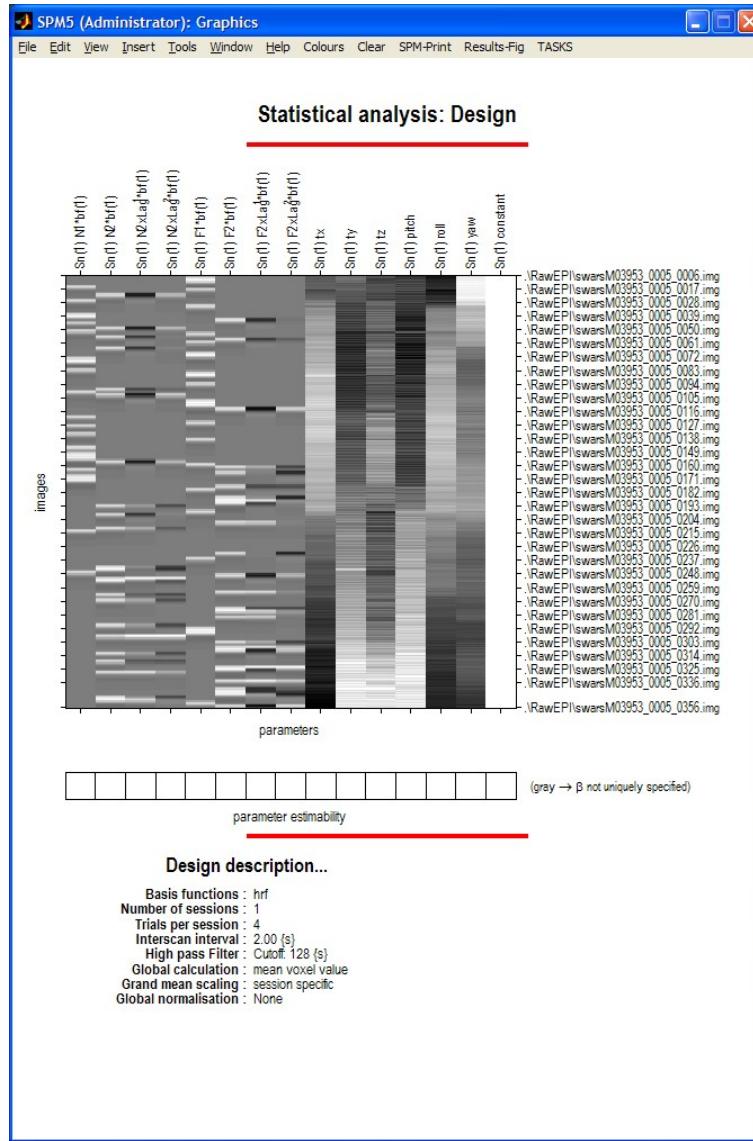


Figure 30.16: *Design matrix for testing repetition effects parametrically. Regressor 2 indicates the second occurrence of a nonfamous face. Regressor 3 modulates this linearly as a function of lag (ie. how many faces have been shown since that face was first presented), and regressor 4 modulates this quadratically as a function of lag. Regressors 6,7 and 8 play the same roles, but for famous faces.*

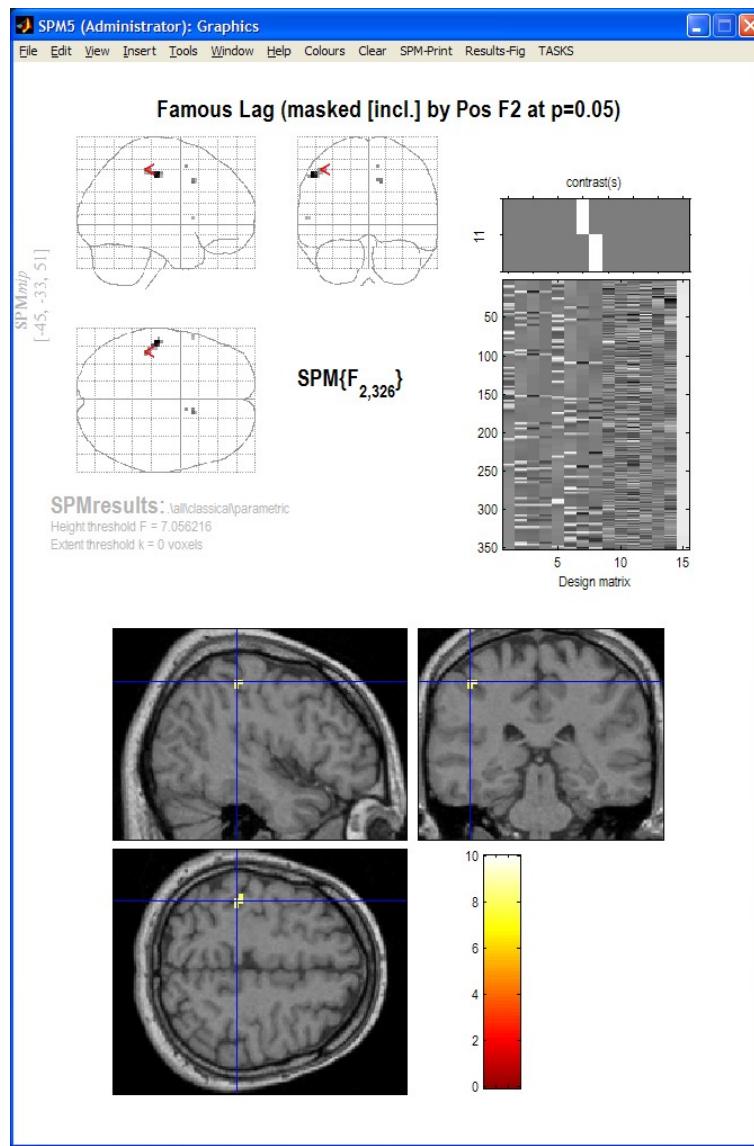


Figure 30.17: MIP and overlay of parametric lag effect in parietal cortex.

- Accept what is offered
- *p value adjustment to control: [FWE/FDR/none]*
- Select None
- *Threshold {F or p value}*
- Accept the default value, 0.001
- *Extent threshold {voxels} [0]*
- Accept the default value, 0

Figure 30.17 shows the MIP and an overlay of this parametric effect using overlays, sections and selecting the `wmsM03953_0007.img` image. The effect is plotted in the time domain in figure 30.18. This was obtained by

- Right clicking on the MIP and selecting ‘global maxima’

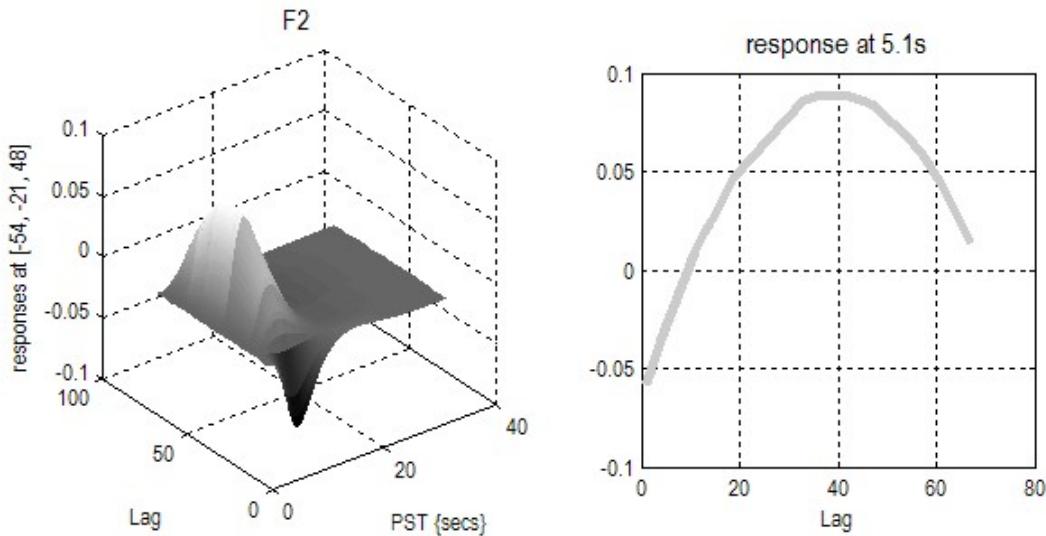


Figure 30.18: *Response as a function of lag.*

- 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).

## 30.4 Bayesian analysis

### 30.4.1 Specification

Press the ‘Specify 1st-level’ button. This will call up an fMRI specification job in the graphics window. Then

- Open the fMRI model specification option
- 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).
- Save the job as `specify_bayesian.mat` and press ‘Run’

### 30.4.2 Estimation

Press the ‘Estimate’ button. This will call up the specification of an fMRI estimation job in the graphics window. Then

- Open the ‘fMRI model estimation’ option
- 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 Run

SPM will write a number of files into the output directory including

- An `SPM.mat` file.
- Images `Cbeta_k.img` 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.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.
- Images `Sess1_AR_p.img` where  $p$  indexes the  $p$ th AR coefficient. See eg. Figure 30.19.
- Images `con_i.img` and `con_sd_i.img` which are the mean and standard deviation of the  $i$ th pre-defined contrast.

### 30.4.3 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>9</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).

---

<sup>9</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.

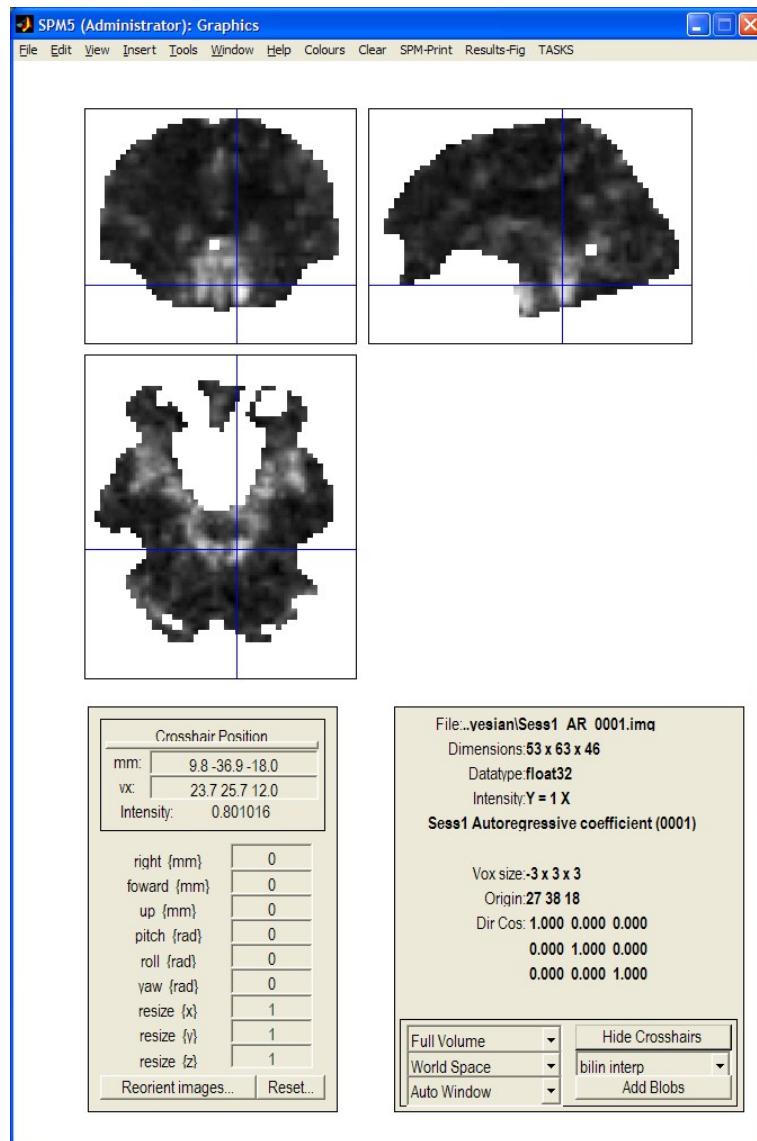


Figure 30.19: *Bayesian analysis: Estimated AR(1) coefficient image indicating heterogeneity near the circle of Willis*

- 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’.
- *Mask with other contrast ? [Yes/No]*
- Specify No
- *Title for comparison*
- Enter ‘AVERAGE Canonical HRF: Faces > Baseline’
- *Effect size threshold for PPM*
- Enter the value
- *Posterior probability threshold for PPM*
- Enter the value 0.95
- *Extent threshold [0]*
- Accept the default value
- *Plot effect size [Yes/No]*
- Select the default ‘Yes’

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 30.20

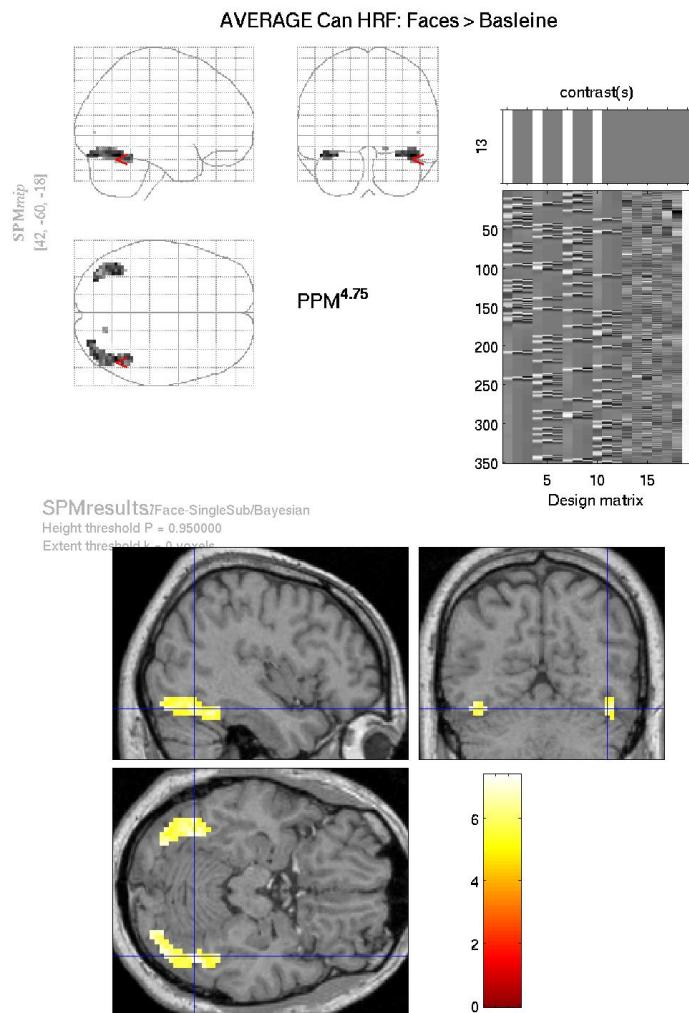


Figure 30.20: Bayesian analysis: MIP and overlay of effect sizes at voxels where PPM is 95% sure that the effect size is greater than 1% of the global mean. The cursor is at the location  $x = 42, y = -60, z = -18$  mm

# Chapter 31

## Face group data

### 31.1 Introduction

These examples illustrate multisubject ‘random effects’ analyses or ‘second-level’ models of fMRI data [54]<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 x 2 second timebins. This uses 12 observations (contrast images) per subject. Data are analysed using a ‘One-way ANOVA’ with 12 levels.

### 31.2 Data

The data come from the ‘implicit’ condition of the Henson et al. study [35]. 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 earlier Chapter 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 (SOAmin) was 4.5s, with all faces randomly intermixed together with a further 52 null events (ie 2/3 probability of a face every SOAmin).

Original images were continuous EPI (TE=40ms,TR=2s) 24 descending slices (64x64 3x3mm<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/archive/face-contrasts/rfx-multiple.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 ... 0.5] contrast for each basis function, or "kron(eye(Nf),[0.5 0.5])" more generally) was examined.

The 12 (subjects) con\*.imgs from the 1st-level model using the canonical HRF (M1c) are in the zipped file

[http://www.fil.ion.ucl.ac.uk/spm/data/archive/face-contrasts/cons\\_can.zip](http://www.fil.ion.ucl.ac.uk/spm/data/archive/face-contrasts/cons_can.zip).

The 12 (subjects) x 3 (basis functions) con\*.imgs from the 1st-level model using the informed basis (M1i) set are in the zipped file

[http://www.fil.ion.ucl.ac.uk/spm/data/archive/face-contrasts/cons\\_informed.zip](http://www.fil.ion.ucl.ac.uk/spm/data/archive/face-contrasts/cons_informed.zip).

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/data/archive/face-contrasts/cons\\_fir.zip](http://www.fil.ion.ucl.ac.uk/spm/data/archive/face-contrasts/cons_fir.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:\home\wpenny\fmri\_analysis\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.

### 31.3 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.
- Double click on the ‘+Factorial design specification’ text.
- Double click on the ‘+One-sample t-test’ text, then highlight ‘Scans’.
- Select ‘Specify Files’ and use the SPM file selector to choose contrast images 3 to 14.
- Highlight Directory, Specify 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 in the graphics window.

SPM will then show you the design matrix shown in Figure 31.1. 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’, double click on ‘+fMRI model estimation’, select the SPM.mat file just created, and press ‘RUN’. 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\*.img’s you have specified.

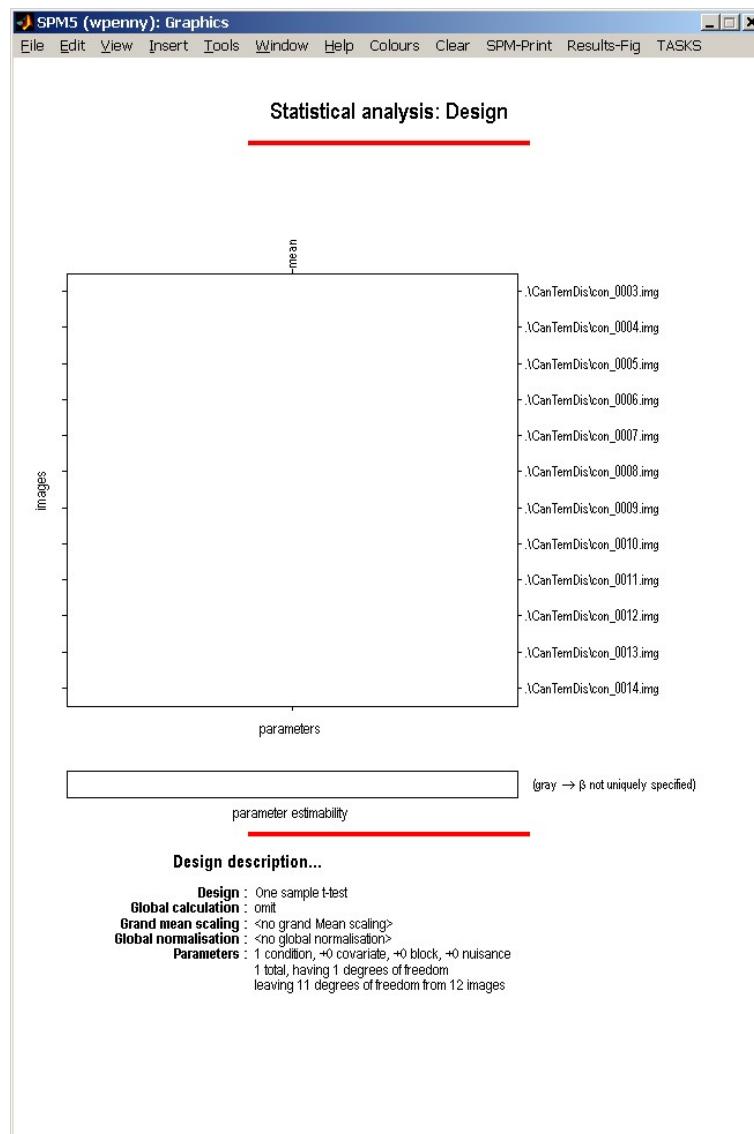


Figure 31.1: *Design matrix for canonical responses. This corresponds to a one-sample t-test.*

- Now press the 'Results' button.
- Select the SPM.mat file.
- 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]
- Title for comparison: accept [Faces vs Baseline: Canonical HRF]
- 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 (e.g, +30 -63 -27, Z=6.04), 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 31.2.

### 31.4 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.

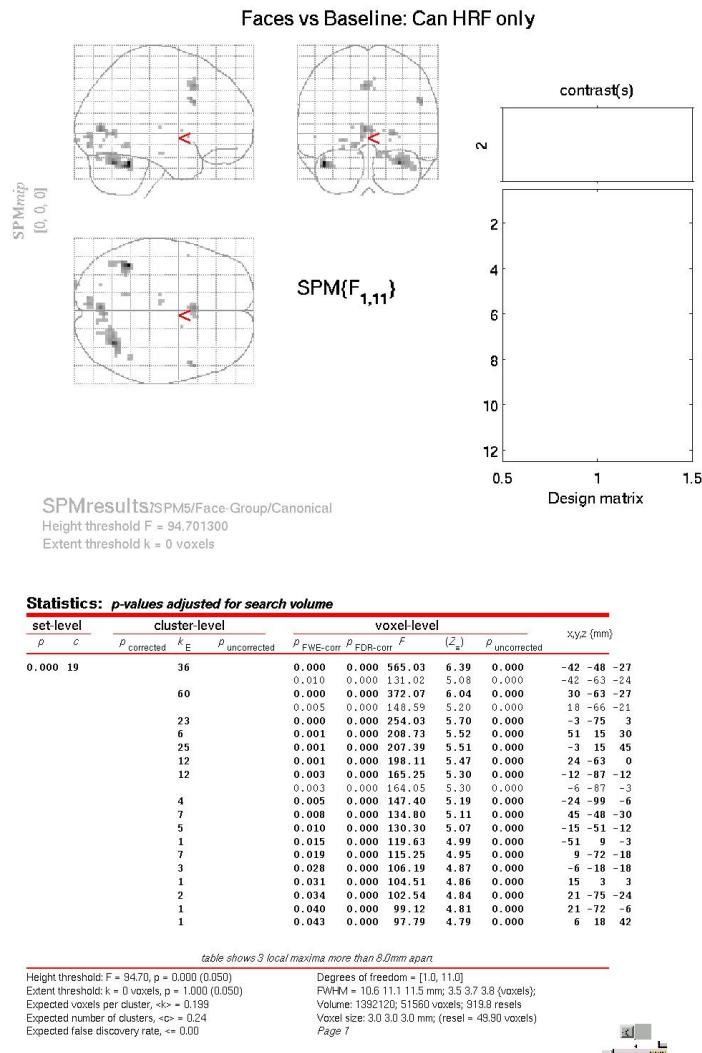


Figure 31.2: Main population effect of faces vs baseline, as characterised using the Canonical HRF.

- Double click on the ‘+Factorial design specification’ text.
- Highlight ‘Design’ and then choose ‘Full Factorial’
- Double click ‘+Full Factorial’, and under ‘Factors’ create a single ‘New Factor’
- Open this Factor and type in ‘Basis’ for Name and enter 3 under ‘Levels’.
- 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 graphics window.

SPM will then show you the design matrix shown in Figure 31.3. This design is encoded in the ‘SPM.mat’ file that is written to the output directory. Then press ‘Estimate’, double click on ‘+fMRI model estimation’, select the SPM.mat file just created, and press ‘RUN’. SPM will now estimate the parameters of the model (and hyperparameters governing the nonsphericity).

### 31.4.1 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 31.4. 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.

### 31.4.2 Informed Results

- Now press the ‘Results’ button.
- Select the SPM.mat file.

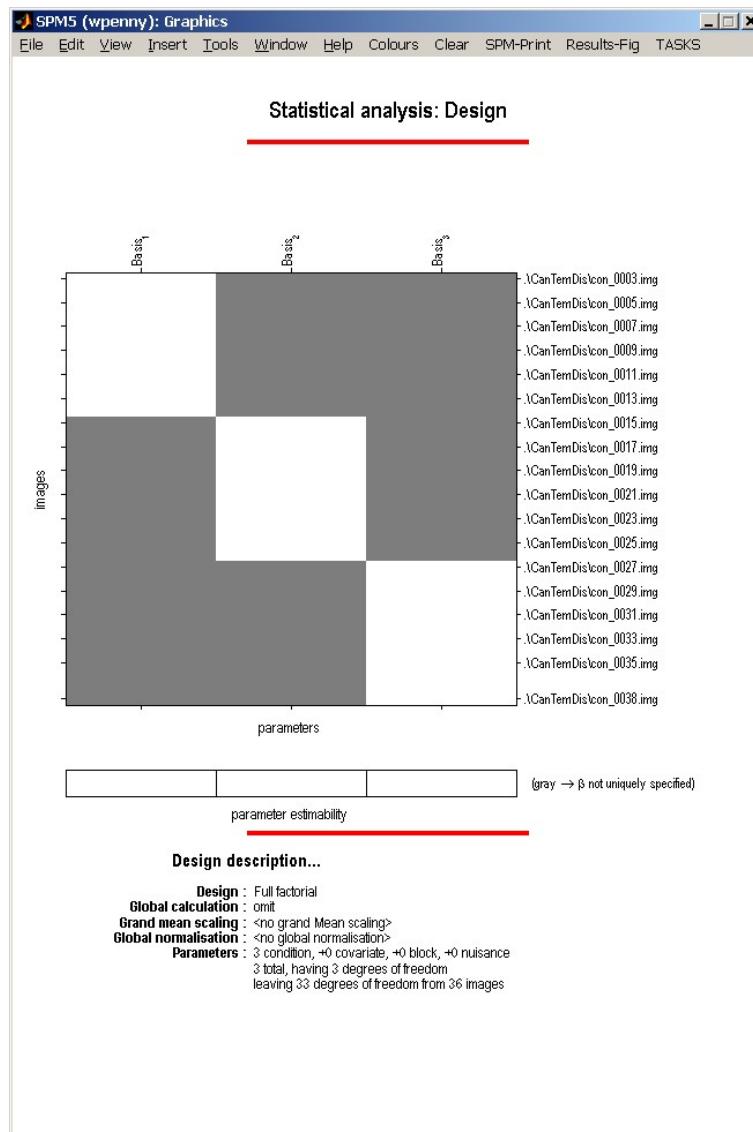


Figure 31.3: *Design matrix for informed basis set. This corresponds to a one-way ANOVA with three levels (but no constant term, since we want to test whether the basis functions are different from zero, not whether they are different from each other).*

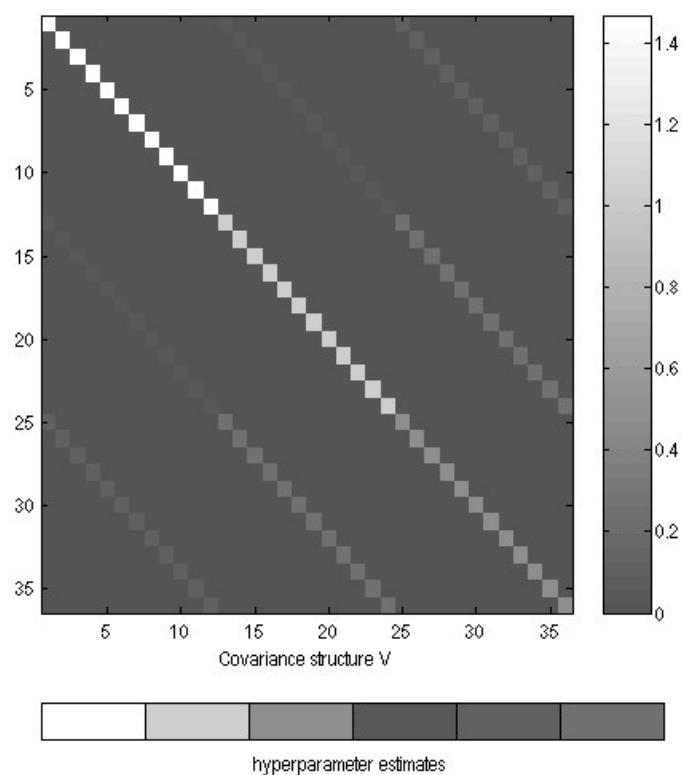


Figure 31.4: Estimated covariance matrix for informed basis set. The 6 differently valued hyperparameters are shown in different shades of gray.

- 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 [1 0 0; 0 1 0; 0 0 1].<sup>2</sup>.
- Press the '..submit' button. Press OK.
- Now press the 'Done' button.
- Mask with other contrast(s) [No]
- Title for comparison: accept [Faces vs Baseline: Informed]
- 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 31.5.

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 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 31.5) produces more significant results than the corresponding F-contrast in the model with the canonical HRF shown in Figure 31.2. 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 31.6. 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).

### 31.4.3 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 31.7.

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=7.51) 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) [30].

Finally, evaluate the F-contrasts [0 1 0] and [0 0 1]. These are shown in Figures 31.8 and 31.9. 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 Eg3.1 below for more discussion; see also to Henson et al (2000) [34].

---

<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.

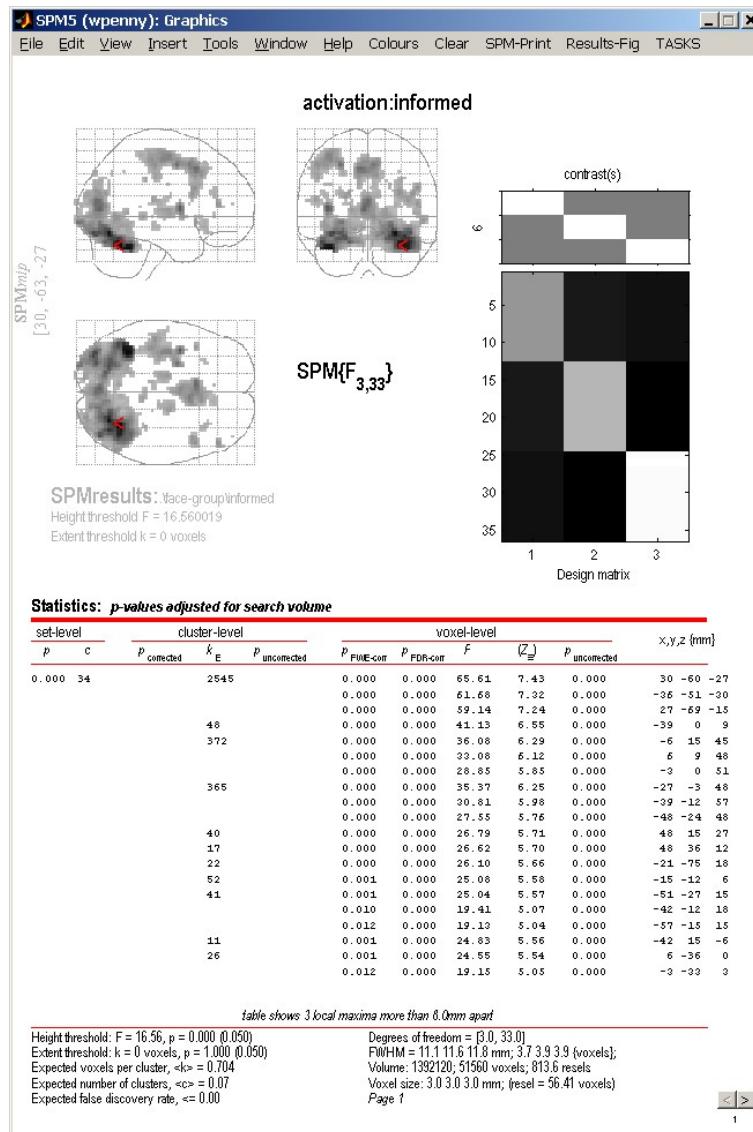


Figure 31.5: Main population effect of faces, as characterised with the informed basis set.

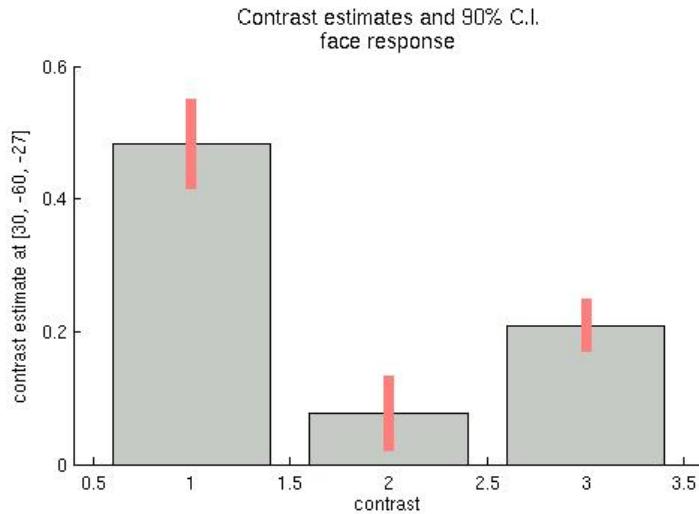


Figure 31.6: Plotting the three basis functions for the global maximum showing reliable effects of the canonical HRF and its time and dispersion derivatives.

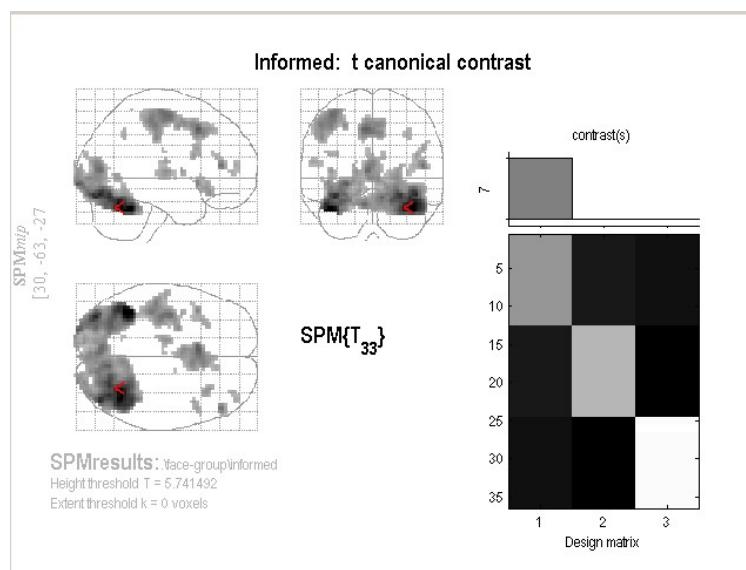


Figure 31.7: Main population effect of faces, as characterised with the canonical HRF using a  $[1 \ 0 \ 0]$  t-contrast on the informed basis coefficients.

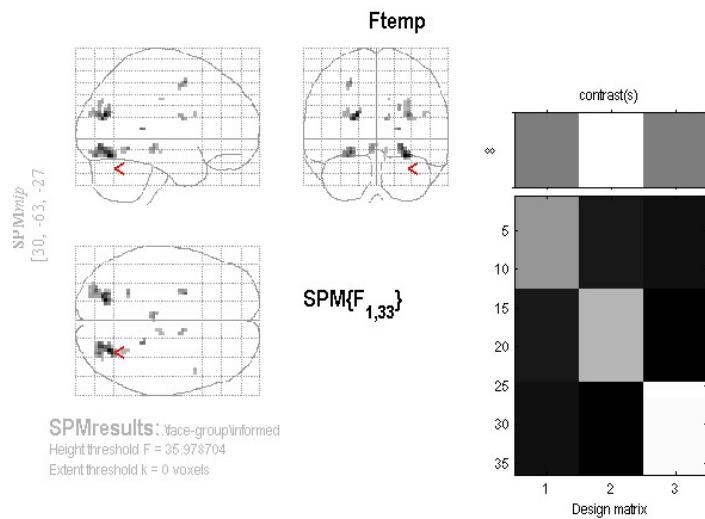


Figure 31.8: Significantly non-zero temporal derivative coefficients. These voxels show responses earlier or later than canonical responses.

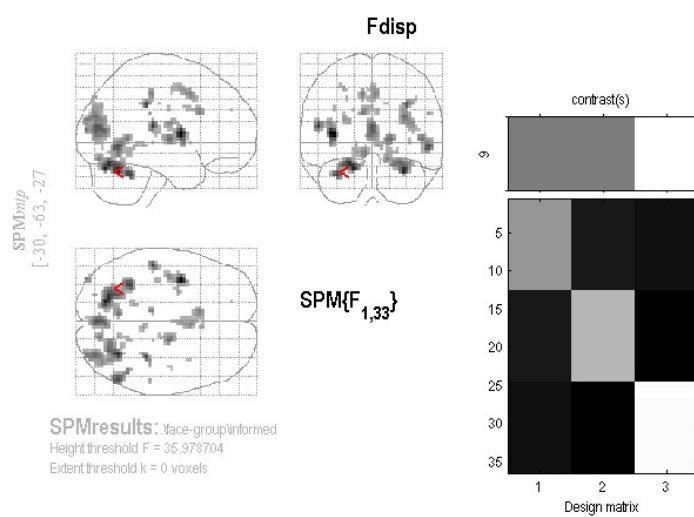


Figure 31.9: Significantly non-zero dispersion derivative coefficients. These voxels show responses narrower or wider than canonical responses.

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).

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 [33]). 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).

## 31.5 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.
- Double click on the '+Factorial design specification'<sup>3</sup> text.
- Highlight 'Design' and then choose 'Full Factorial'
- Double click '+Full Factorial', and under 'Factors' create a single 'New Factor'
- Open this Factor and 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 `^\\w*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 `^\\w*bin02.*`.
- Similarly for Levels 3 to 12.

---

<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.

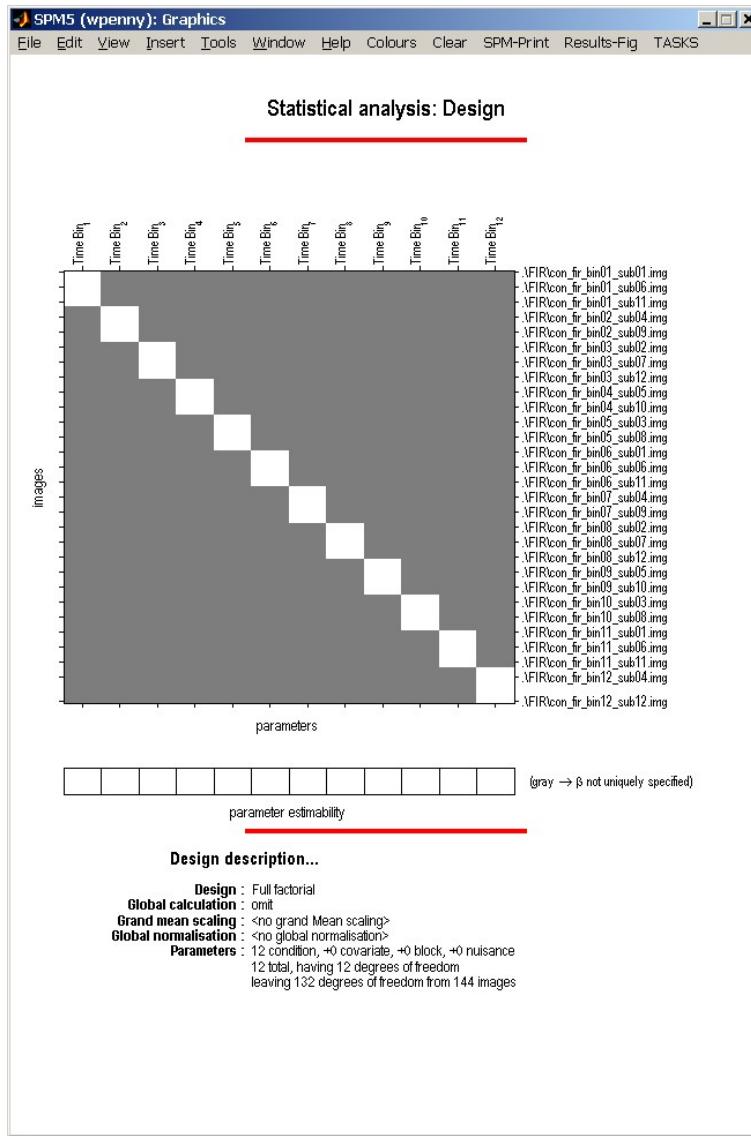


Figure 31.10: *Design matrix for FIR basis set. This corresponds to a one-way ANOVA with 12 levels.*

- 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 graphics window.

SPM will then show you the design matrix shown in Figure 31.10. This design is encoded in the ‘SPM.mat’ file that is written to the output directory. Then press ‘Estimate’, double click on ‘+fMRI model estimation’, select the SPM.mat file just created, and press ‘RUN’. SPM will now estimate the parameters of the model.

### 31.5.1 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).

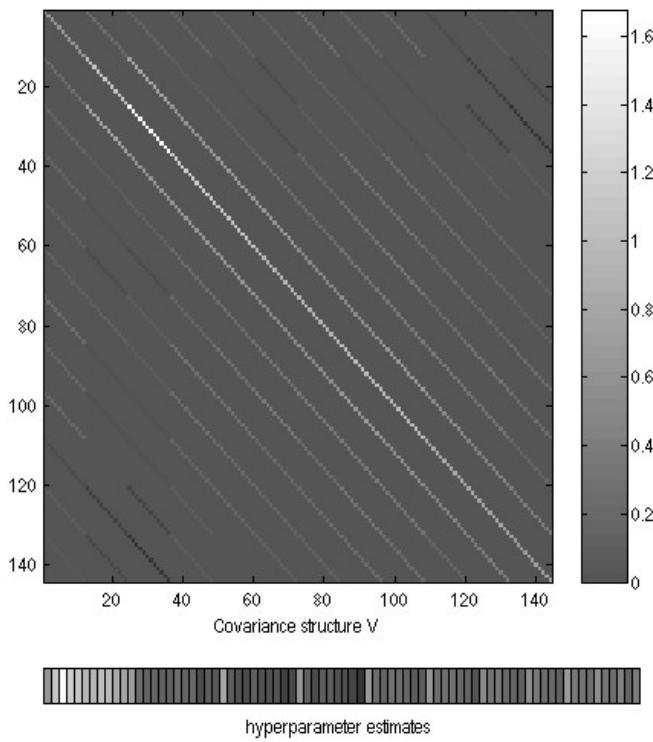


Figure 31.11: *Estimated covariance matrix for FIR basis set. The differently valued hyperparameters are shown in different shades of gray. Notice that the most variable responses occur in the third time bin (scans 25 to 36) corresponding to responses 4-6 seconds post stimulus, ie. at the peak of the hemodynamic response, as expected.*

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 31.11. 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).

### 31.5.2 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]
- Title for comparison: accept [Faces vs Baseline: FIR]
- p value adjustment to control [FWE]
- Family-wise p-value [0.05]
- Extent threshold voxels [0]

Note how the design matrix, shown in Figure 31.12 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 31.11).

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 31.12. 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 down-sample the basis functions every 2s, which is the bin-width of the FIR. If you type '`corcoef(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.

This produces the MIP in Figure 31.13. 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)

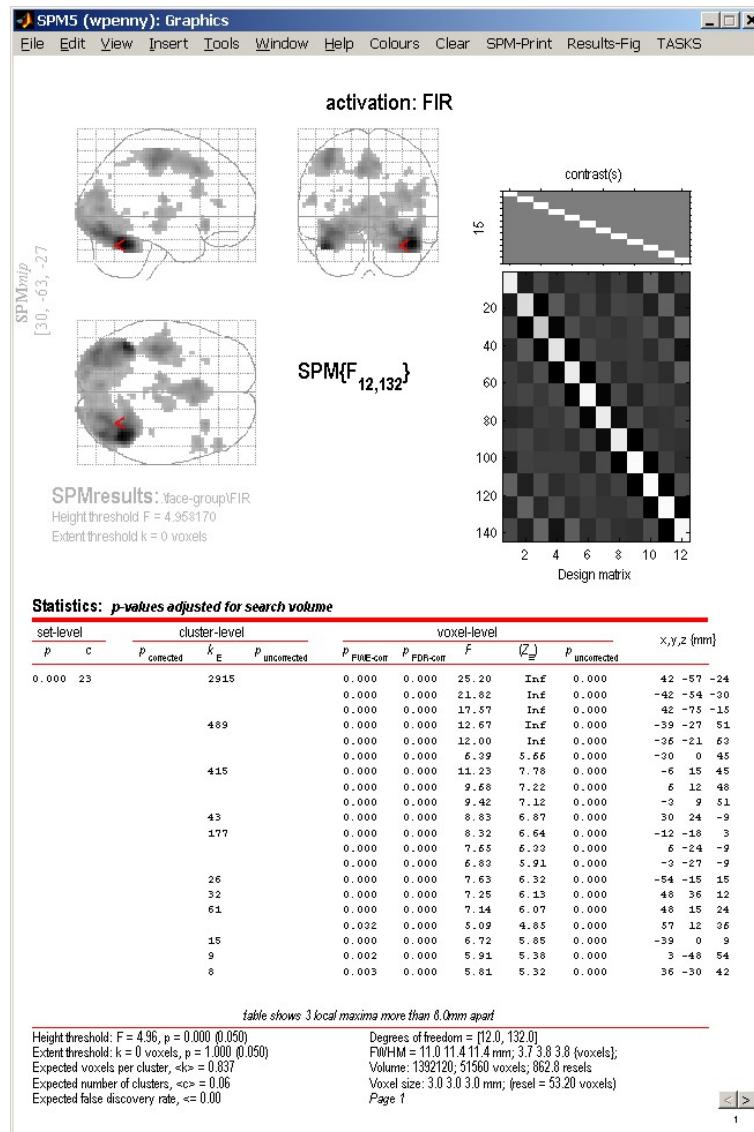


Figure 31.12: Main population effect of faces, as characterised with the FIR basis set.

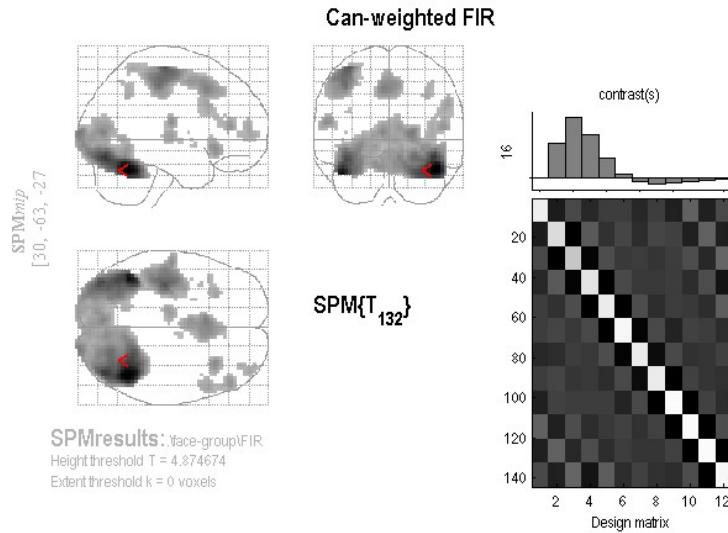


Figure 31.13: *Main population effect of faces, as characterised with a canonically weighted contrast of FIR bases.*

voxels [30]. 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.

[30]. You can see, in Figure 31.14 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).
- Enter [‘nullall’] as the contrast weights (defined in Matlab workspace as above), and ‘Null space of informed basis set’ 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.

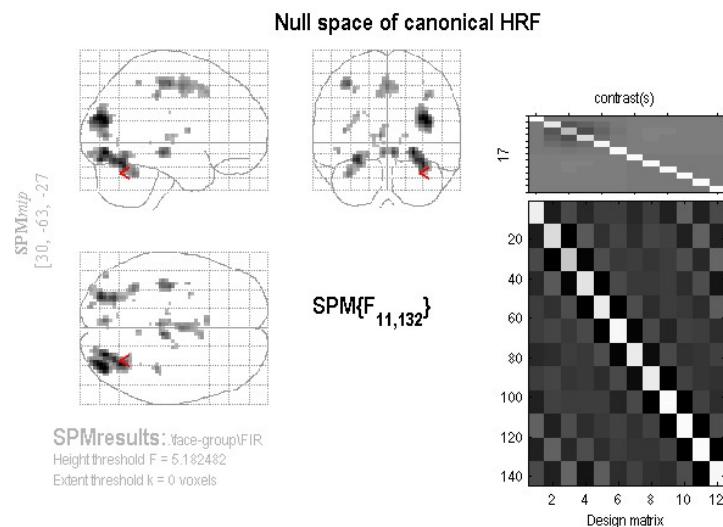


Figure 31.14: Regions expressing variability across subjects not captured by canonical HRF.

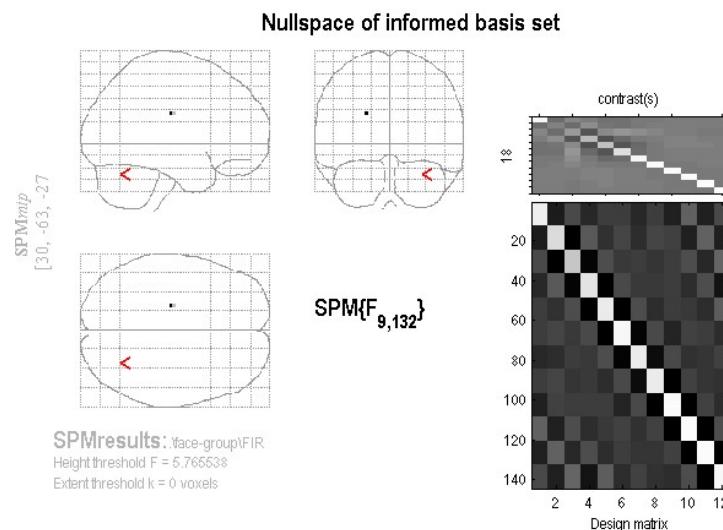


Figure 31.15: Regions expressing variability across subjects not captured by informed basis set.

You will see, in Figure 31.15 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 [34] for further details.

# Chapter 32

## Verbal Fluency PET data

### 32.1 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 32.1. The files are named

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 32.1: *Conditions for PET data: (A) word shadowing and (B) word generation.*

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

To analyse the data, first create a new directory DIR  
eg. c:\home\wpenny\fmri\_analysis\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.

### 32.2 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 ‘PET’ button.
- Double click on the ‘+Factorial design specification’ text.
- Choose the ‘Flexible Factorial’ design and open it up.
- 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.

- 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>1</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 in the graphics window.

SPM will then show you the design matrix shown in Figure 32.1. 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]
- Title for comparison [activation]
- 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.

### 32.3 Multiple subjects

The data set can be analysed in several ways which are discussed in [41].

---

<sup>1</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.

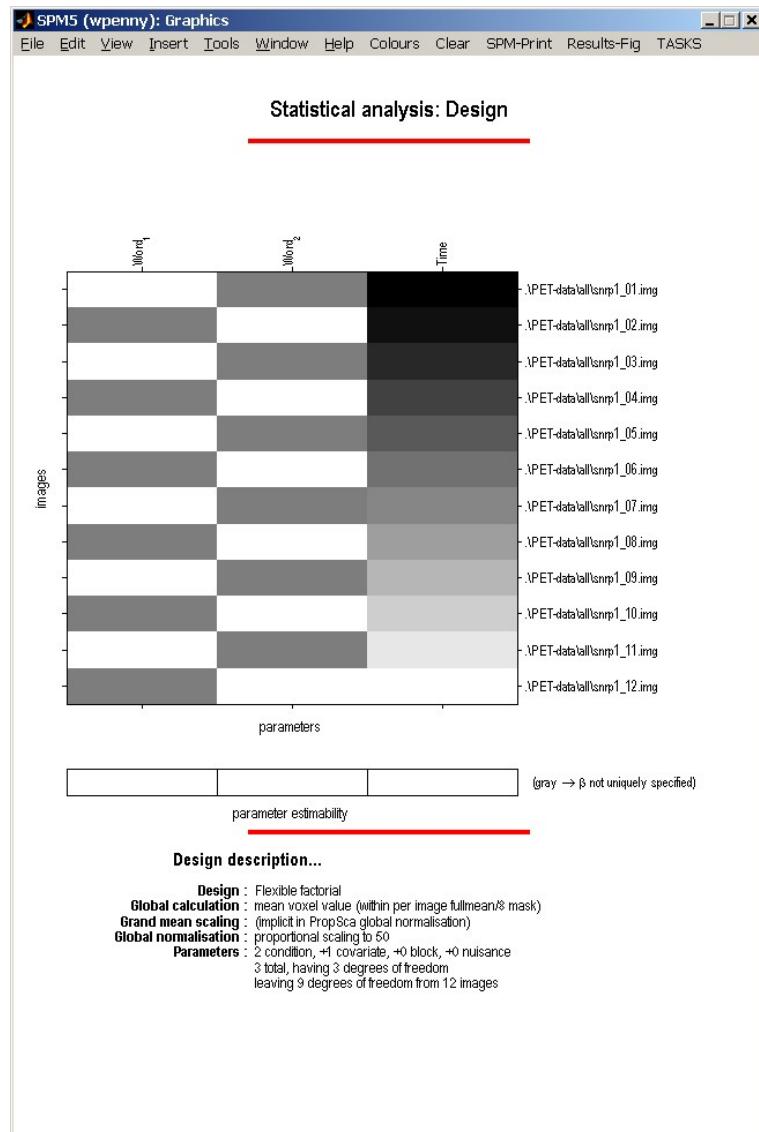


Figure 32.1: *Design matrix for single-subject data. The first two columns model responses to word shadowing and word generation. The third column models time-varying responses.*

### 32.3.1 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 ‘PET’ button.
- Double click on the ‘+Factorial design specification’ text.
- Under ‘Design’, choose the ‘Flexible Factorial’ design and open it up.
- 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].
- 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 32.1).
- 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 in the graphics window.

SPM will then show you the design matrix shown in Figure 32.2. This design is encoded in the ‘SPM.mat’ file that is written to the output directory.

### 32.3.2 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, 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 ‘PET’ button.
- Double click on the ‘+Factorial design specification’ text.
- Under ‘Design’, choose the ‘Flexible Factorial’ design and open it up.

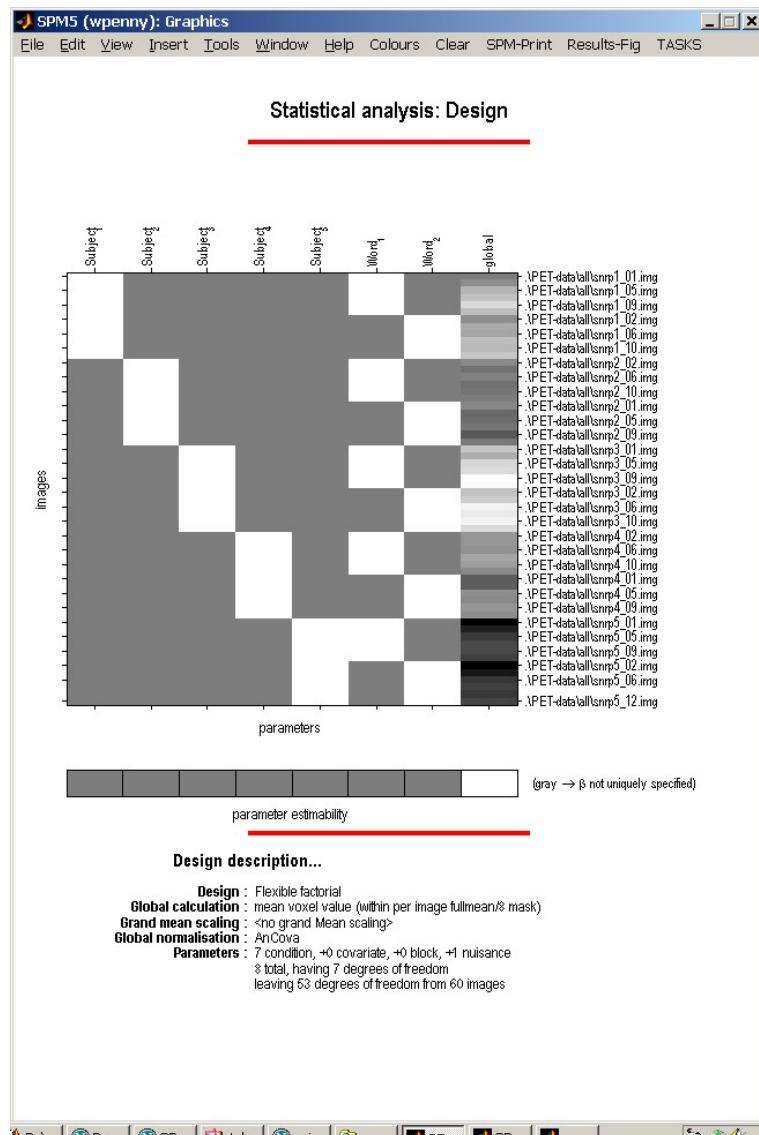


Figure 32.2: *Subjects and Conditions design for multiple-subject data. The first five columns model effect and the next two columns model condition effects. The last column models global effects (ANCOVA).*

- Highlight ‘Factors’ and create a new Factor.
- 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 in the graphics window.

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.

### 32.3.3 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 ‘PET’ button.
- Double click on the ‘+Factorial design specification’ text.
- Under ‘Design’, choose the ‘Flexible Factorial’ design and open it up.
- 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.

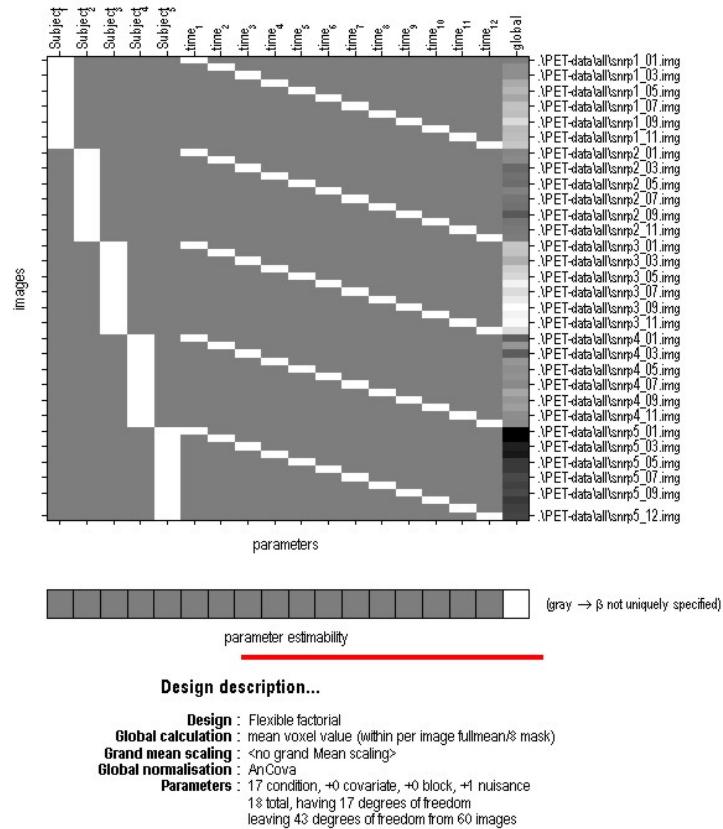


Figure 32.3: *Subjects and Time design for multiple-subject data. The first five columns model subjects effects and the next 12 model time effects. The last column models global effects (ANCOVA).*

- 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].
- 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 32.1).
- 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 in the graphics window.

SPM will then show you the design matrix shown in Figure 32.4. 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.

### 32.3.4 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]
- Title for comparison [activation]
- p value adjustment to control [FWE]
- Family-wise p-value [0.05]
- Extent threshold voxels [0]

This should produce the contrast in Figure 32.5. 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)

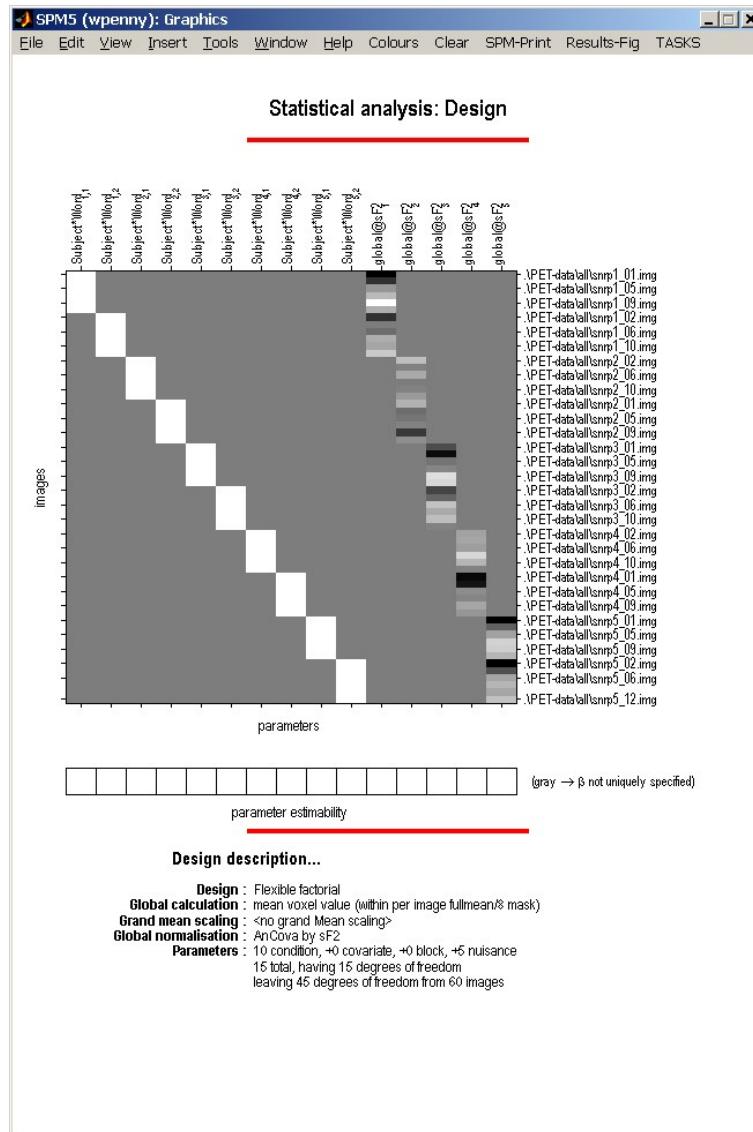


Figure 32.4: *Subject by Condition design for multiple-subject data. The first ten columns model interactions between ‘Subject’ and ‘Word’. The last five columns model out global effects for each subject. Inclusion of these last five regressors implements a so-called ‘ANCOVA-by-subject’ normalisation.*

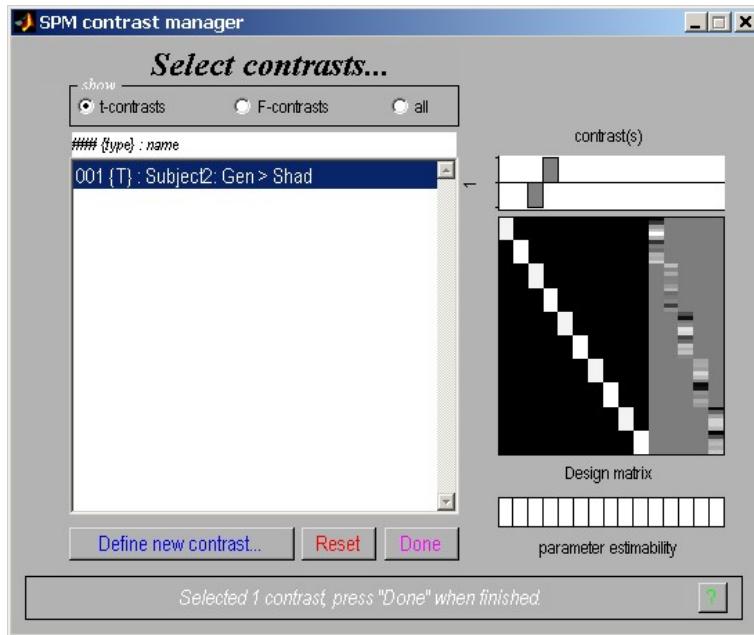


Figure 32.5: Activation contrast for subject 2. Note that the block of the design matrix encoding the experimental conditions is now coloured differently. This is because we have allowed the variance of responses over subjects to be different between word shadowing and generation conditions. This ‘nonsphericity’ 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.

- 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]
- 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.

### 32.3.5 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.

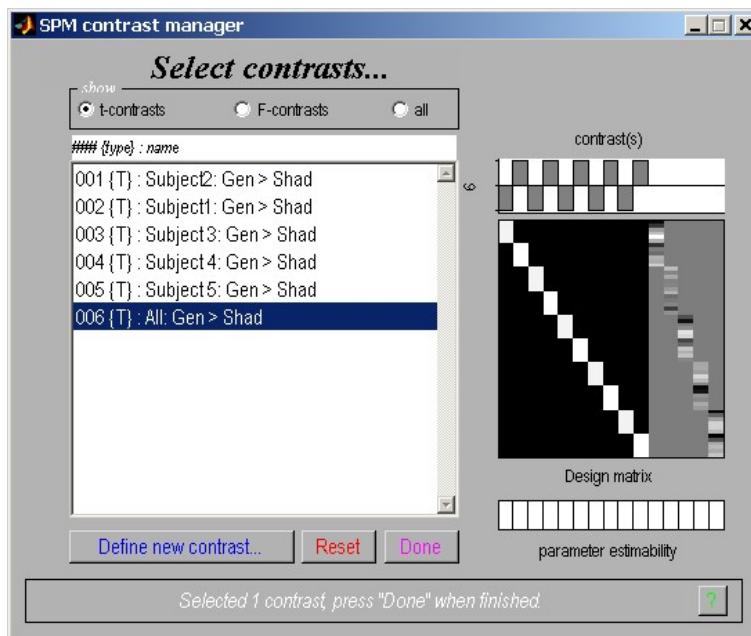


Figure 32.6: Activation contrast for all subjects.

### 32.3.6 MIPs and results tables

The above contrast specifications should configure the contrast manager to appear as in Figure 32.6 and will configure SPM's graphics window to look like Figure 32.7. 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's interactive window shown in Figure 32.8. 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 32.9. As in the previous example, this will list all clusters above the chosen level of significance as well as separate ( $\geq 8\text{mm}$  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
- voxel-level: the chance (p) of finding (under the null hypothesis) a voxel 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.

It's 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 L prefrontal cortex), 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).

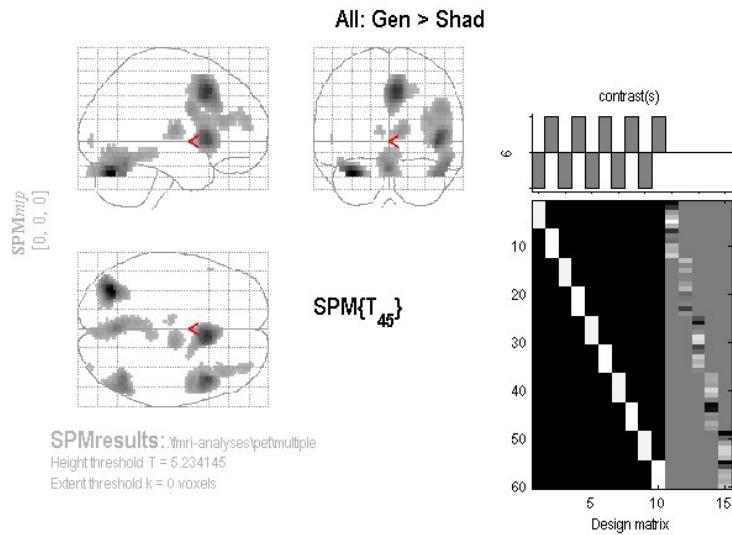


Figure 32.7: *SPMs* graphics window displays (Left) a maximum intensity projection (MIP) on a glass brain in three orthogonal planes. The MIP is surfable: R-clicking in the MIP will activate a pulldown menu, L-clicking on the red cursor will allow it to be dragged to a new position, (Right) the design matrix (showing the selected contrast). The design matrix is also surfable: R-clicking will show parameter names, L-clicking will show design matrix values for each scan.



Figure 32.8: *SPM*'s interactive window.

Statistics: p-values adjusted for search volume												
set-level	c	cluster-level			voxel-level					x,y,z {mm}		
		p <sub>corrected</sub>	K <sub>E</sub>	p <sub>uncorrected</sub>	p <sub>FWE-corr</sub>	p <sub>FDR-corr</sub>	T	Z	p <sub>uncorrected</sub>	x	y	z
0.000	10	0.000	227	0.000	0.000	0.000	13.24	Inf	0.000	-34 -70 -28		
		0.000	625	0.000	0.000	0.000	10.80	7.55	0.000	6 16 44		
					0.000	0.000	8.12	6.34	0.000	2 24 36		
					0.008	0.000	5.86	5.03	0.000	20 0 44		
		0.000	843	0.000	0.000	0.000	10.02	7.23	0.000	44 16 0		
					0.000	0.000	6.96	5.71	0.000	48 4 28		
					0.001	0.000	6.51	5.43	0.000	38 32 16		
		0.000	439	0.000	0.000	0.000	8.37	6.47	0.000	0 -66 -24		
					0.000	0.000	7.14	5.81	0.000	-4 -60 -16		
					0.001	0.000	6.56	5.47	0.000	4 -78 -24		
		0.000	259	0.000	0.000	0.000	8.35	6.45	0.000	52 -58 -20		
					0.000	0.000	8.32	6.44	0.000	48 -60 -28		
					0.000	0.000	7.29	5.90	0.000	54 -66 -16		
		0.000	103	0.000	0.000	0.000	6.92	5.68	0.000	10 -10 8		
					0.011	0.000	5.78	4.97	0.000	2 -12 8		
0.009	4	0.177		0.009	0.000	5.84	5.01	0.000	-52 20 4			
0.001	14	0.019		0.010	0.000	5.80	4.99	0.000	-8 -16 8			
0.009	4	0.177		0.026	0.000	5.46	4.76	0.000	32 -90 0			
0.017	2	0.336		0.034	0.000	5.37	4.69	0.000	-6 -20 4			

table shows 3 local maxima more than 6.0mm apart

Height threshold: T = 5.23, p = 0.000 (0.050)  
Extent threshold: k = 0 voxels, p = 1.000 (0.050)  
Expected voxels per cluster, <k> = 2.338  
Expected number of clusters, <c> = 0.05  
Expected false discovery rate, <> = 0.00

Degrees of freedom = [10, 45, 0]  
FWHM = 9.6 10.5 15.3 mm; 4.8 5.2 3.8 {voxels};  
Volume: 880432; 55027 voxels; 502.7 resels  
Voxel size: 2.0 2.0 4.0 mm; {resel} = 96.07 voxels

Figure 32.9: SPM results table. This appears below the MIP, shown in Figure 32.7, 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 32.10. The table is also surfable. Both in the 'volume' and 'cluster' options, p-values are corrected for the entire search volume.

### 32.3.7 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. See also Matthew Brett's tutorial at [http://www.mrc-cbu.cam.ac.uk/Imaging/vol\\_corr.html](http://www.mrc-cbu.cam.ac.uk/Imaging/vol_corr.html). Press the S.V.C. button in SPMs 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 32.11.

### 32.3.8 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 V.O.I. (Volume Of Interest) in the interactive window
- Select ('don't adjust')
- Specify 'Broca' for name of region and 0 for the VOI radius.

SPM displays a graph of the first eigenvariate of the data in or centered around the chosen voxel, as shown in Figure 32.12. 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			voxel-level					
$p_{\text{corrected}}$	$k_E$	$p_{\text{uncorrected}}$	$p_{\text{FWE-corr}}$	$p_{\text{FDR-corr}}$	$T$	$(Z_{\text{v}})$	$p_{\text{uncorrected}}$	x,y,z [mm]
0.000	843	0.000	0.000	0.000	10.02	7.23	0.000	44 16 0
			0.000	0.000	6.96	5.71	0.000	48 4 28
			0.000	0.000	6.83	5.63	0.000	50 0 28
			0.001	0.000	6.51	5.43	0.000	38 32 16
			0.002	0.000	6.35	5.34	0.000	34 36 20
			0.003	0.000	6.20	5.24	0.000	36 28 20
			0.003	0.000	6.19	5.23	0.000	38 12 16
			0.003	0.000	6.16	5.22	0.000	40 14 20
			0.004	0.000	6.11	5.19	0.000	38 28 20
			0.004	0.000	6.08	5.17	0.000	34 52 20
			0.005	0.000	6.03	5.13	0.000	36 54 12
			0.010	0.000	5.81	4.99	0.000	34 44 24
			0.012	0.000	5.74	4.94	0.000	36 20 16
			0.022	0.000	5.52	4.80	0.000	30 12 12
			0.022	0.000	5.52	4.80	0.000	32 28 16

table shows 32 local maxima more than 4.0mm apart

Height threshold:  $T = 5.23$ ,  $p = 0.000$  (0.050)  
Extent threshold:  $k = 0$  voxels,  $p = 1.000$  (0.050)  
Expected voxels per cluster,  $\langle k \rangle = 2.338$   
Expected number of clusters,  $\langle n \rangle = 0.05$   
Expected false discovery rate,  $\langle FDR \rangle = 0.00$

Degrees of freedom = [1,0,45,0]  
FWHM = 9.6 10.5 15.3 mm; 4.8 5.2 3.8 (voxels);  
Volume: 880432; 55027 voxels; 502.7 resels  
Voxel size: 2.0 2.0 4.0 mm; (resel = 96.07 voxels)

Figure 32.10: 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			voxel-level					
$p_{\text{corrected}}$	$k_E$	$p_{\text{uncorrected}}$	$p_{\text{FWE-corr}}$	$p_{\text{FDR-corr}}$	$T$	$(Z_{\text{v}})$	$p_{\text{uncorrected}}$	x,y,z [mm]
0.000	701	0.000	0.000	0.000	10.02	7.23	0.000	44 16 0
			0.000	0.000	6.73	5.57	0.000	50 2 24
			0.000	0.000	6.57	5.47	0.000	44 6 28
			0.000	0.000	6.51	5.43	0.000	38 32 16
			0.000	0.000	6.35	5.34	0.000	34 36 20
			0.000	0.000	6.20	5.24	0.000	36 28 20
			0.000	0.000	6.19	5.23	0.000	38 12 16
			0.000	0.000	6.16	5.22	0.000	40 14 20
			0.002	0.000	5.74	4.94	0.000	36 20 16
			0.003	0.000	5.52	4.80	0.000	30 12 12
			0.003	0.000	5.52	4.80	0.000	32 28 16

table shows 16 local maxima more than 4.0mm apart

Height threshold:  $T = 5.23$ ,  $p = 0.000$  (0.008)  
Extent threshold:  $k = 0$  voxels,  $p = 1.000$  (0.008)  
Expected voxels per cluster,  $\langle k \rangle = 2.338$   
Expected number of clusters,  $\langle n \rangle = 0.01$   
Expected false discovery rate,  $\langle FDR \rangle = 0.00$

Degrees of freedom = [1,0,45,0]  
FWHM = 9.6 10.5 15.3 mm; 4.8 5.2 3.8 (voxels);  
Volume: 66128; 4133 voxels; 73.6 resels  
Voxel size: 2.0 2.0 4.0 mm; (resel = 96.07 voxels)

Figure 32.11: 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 32.10.

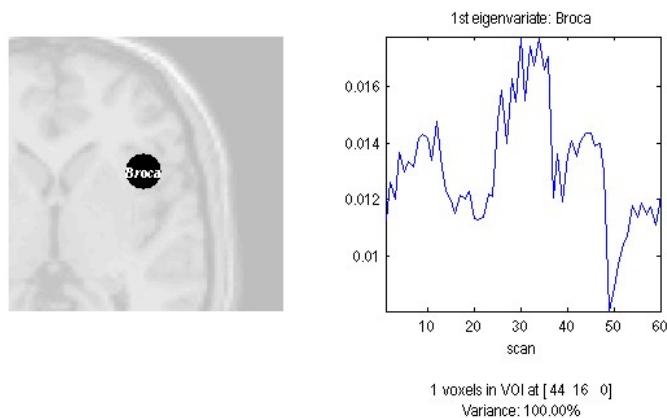


Figure 32.12: *Data extracted from a Volume of Interest (VOI).*

A file called `VOI_regionname.mat` is created in the working directory containing Y and VOI details (in the data structure `xY`).

### 32.3.9 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 32.13.
- Uncorrected mask p-value [0.05]
- Nature of mask [inclusive]
- Title for comparison [accept the default]
- 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 32.14.

### 32.3.10 Conjunctions

To perform a conjunction approach across subjects:

- Press the ‘Results’ button.
- Select the SPM.mat file.

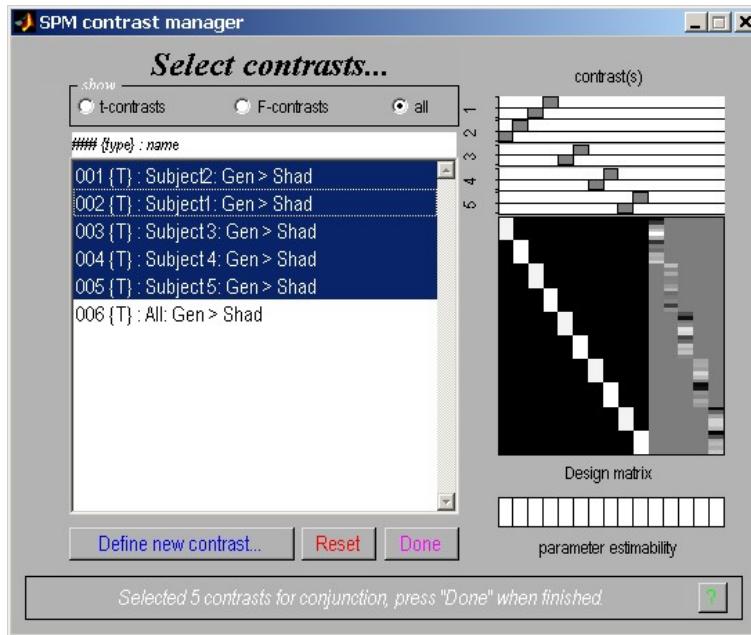


Figure 32.13: *SPM can produce maps based on multiple contrasts by holding down [control] whilst selecting contrasts. This can be used during masking and when making a conjunction inference.*

- Then hold down the [control] button whilst selecting all the individual contrasts. The contrast manager should then appear as in Figure 32.13 (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 32.15. 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.

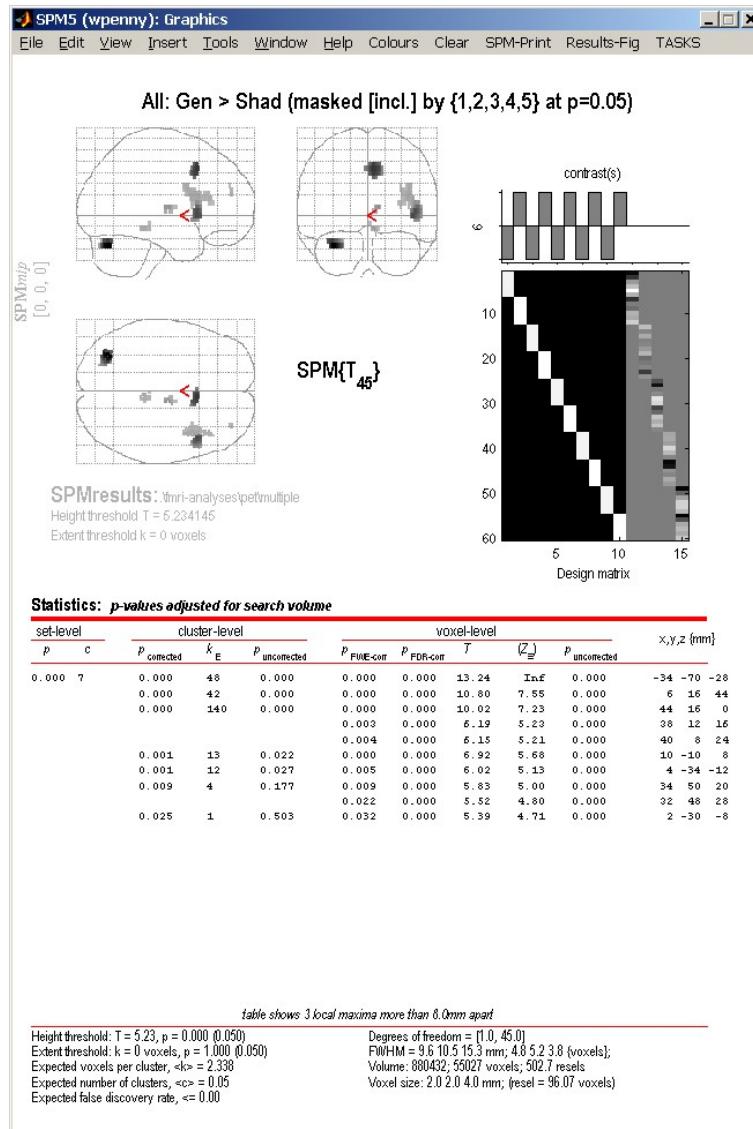


Figure 32.14: The SPM shows results from the inclusive masking approach. It shows all voxels which are (a) significant at  $p < 0.05$  corrected across all subjects and (b) significant at  $p < 0.05$  uncorrected for each subject individually.

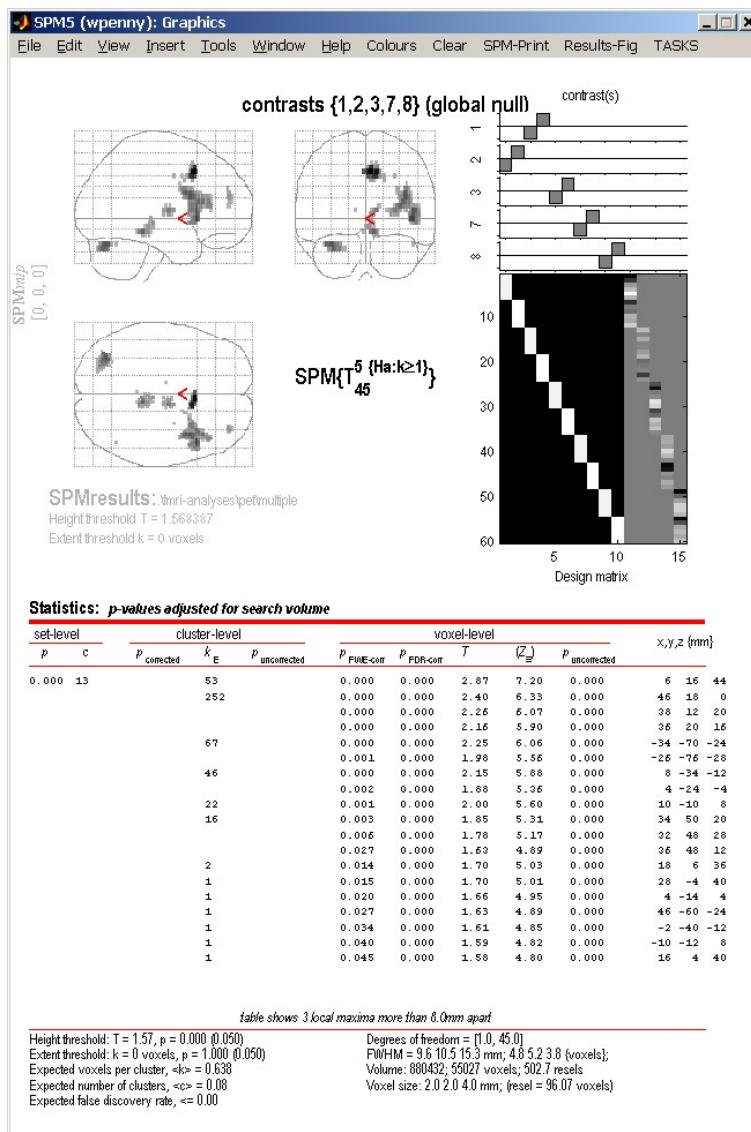


Figure 32.15: Conjunction SPM.

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 [28] and [51].

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 33

## Dynamic Causal Modeling for fMRI

### 33.1 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 [24]. 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 33.1 for a summary of the differences between DCM implementations for fMRI and Event-Related Potentials (ERPs).

As in state-space models, two distinct levels constitute a DCM (see Figure 33.2). 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. , 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):

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ .. \\ \dot{z}_k \end{bmatrix} = \dot{z} = \frac{dz}{dt} = F(z, u, \theta^n) \quad (33.1)$$

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

$$\dot{z} = Az + \sum_{j=1}^m u_j B_j z + Cu \quad (33.2)$$

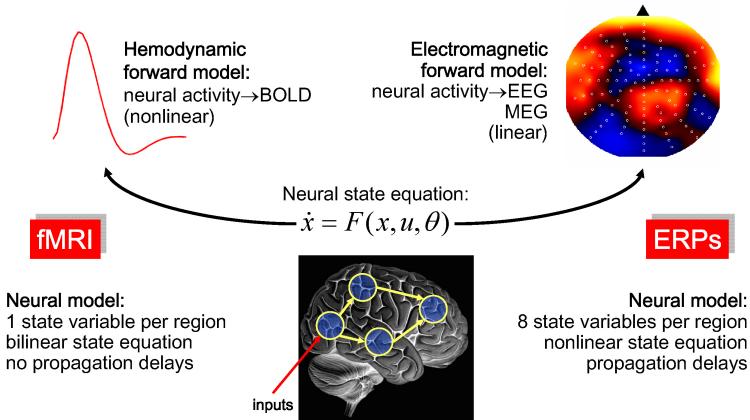


Figure 33.1: A schematic overview of the differences between the DCM implementations for fMRI and ERPs (as measured by EEG or MEG). Whereas the state equation of DCM for fMRI is bilinear and uses only a single state variable per region, that for ERPs is more complex and requires 8 state variables per 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  $F$ :

$$\begin{aligned} A &= \frac{\partial F}{\partial z} = \frac{\partial \dot{z}}{\partial z} \\ B_j &= \frac{\partial^2 F}{\partial z \partial u_j} = \frac{\partial}{\partial u_j} \frac{\partial \dot{z}}{\partial z} \\ C &= \frac{\partial F}{\partial u} \end{aligned} \quad (33.3)$$

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 [26]. 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

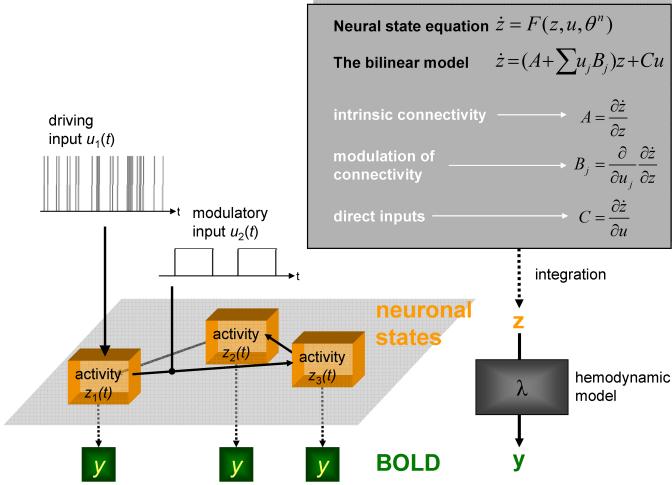


Figure 33.2: Schematic summary of the conceptual basis of DCM. The dynamics in a system of interacting neuronal populations (orange boxes), which are not directly observable by fMRI, is modeled using a bilinear state equation (grey box). Integrating the state equation gives predicted neural dynamics ( $z$ ) 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 intrinsic connections. The strengths of these connections can be changed by modulatory inputs ( $u_2$ ; e.g. changes in cognitive set, attention, or learning).

predicted BOLD signal  $y$  is a non-linear function of blood volume and deoxyhemoglobin content: . Details of the hemodynamic model can be found in other publications [26]. 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

$$\begin{aligned}\dot{x} &= F(x, u, \theta) \\ y &= \lambda(x)\end{aligned}\tag{33.4}$$

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):

$$y = h(u, \theta) + X\beta + e\tag{33.5}$$

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 (Friston et al. 2003). 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. 33.7). 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):

$$p(c^T \eta_{\theta|y} > \gamma) = \Phi_N \left( \frac{c^T \eta_{\theta|y} - \gamma}{c^T C_{\theta|y} c} \right) \quad (33.6)$$

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. 33.3).

Integration of a first-order linear differential equation gives an exponential function:

$$\frac{dz}{dt} = az \quad \rightarrow \quad z(t) = z_0 \exp(at)$$

Coupling parameter  $a$  is inversely proportional to the half life  $\tau$  of  $z(t)$ :

$$\begin{aligned} z(\tau) &= 0.5z_0 \\ &= z_0 \exp(a\tau) \\ \rightarrow a &= \ln 2 / \tau \end{aligned}$$

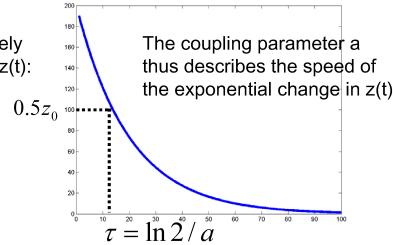


Figure 33.3: A short mathematical demonstration, using a simple linear first-order differential equation as an example, explaining why the coupling parameters in a DCM are inversely proportional to the half-life of the modelled neural responses and are therefore in units of  $1/s = Hertz$ .

## 33.2 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

$$p(\theta|y, m) = \frac{p(y|\theta, m)p(\theta|m)}{p(y|m)} \quad (33.7)$$

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

$$p(y|m) = \int p(\theta|y, m)p(\theta|m)d\theta \quad (33.8)$$

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.

In the context of DCM, one potential solution could be to make use of the Laplace approximation. As shown in [56], 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):

$$\begin{aligned} \ln p(y|m) &= \text{accuracy}(m) - \text{complexity}(m) \\ &= \left[ -\frac{1}{2} \ln |C_e| - \frac{1}{2} (y - h(u, \eta_{\theta|y}))^T C_e^{-1} (y - h(u, \eta_{\theta|y})) \right] \\ &\quad - \left[ \frac{1}{2} \ln |C_p| - \frac{1}{2} \ln |C_{\theta|y}| + \frac{1}{2} (\eta_{\theta|y} - \theta_p)^T C_p^{-1} (\eta_{\theta|y} - \theta_p) \right] \end{aligned} \quad (33.9)$$

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. This is problematic in the context of DCM for fMRI because this prior covariance is defined in a model-specific fashion to ensure that the probability of obtaining an unstable system is very small. Specifically, this is achieved by choosing the prior covariance of the intrinsic coupling matrix A such that the probability of obtaining a positive Lyapunov exponent of A is  $p \leq 0.001$ ; see [24] for details. Consequently, one cannot easily compare models with different numbers of connections. Therefore, alternative approximations to the model evidence are useful for DCMs of this sort. Suitable approximations are afforded by the Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC), which have different complexity terms. As shown in [56], these approximations are given by

$$\begin{aligned} \text{BIC} &= \text{accuracy}(m) - d_\theta \ln N \\ \text{AIC} &= \text{accuracy}(m) - d_\theta \end{aligned} \quad (33.10)$$

where  $d_\theta$  is the number of parameters and  $N$  is the number of data points (scans). If one compares the complexity terms of BIC and AIC, it becomes obvious that BIC pays a heavier penalty than AIC as soon as one deals with 8 or more scans (which is virtually always the case for fMRI data).

Therefore, BIC will be biased towards simpler models whereas AIC will be biased towards more complex models. This can lead to disagreement between the two approximations about which model should be favored. In DCM for fMRI, we have therefore adopted the convention that, for any pairs of models  $m_i$  and  $m_j$  to be compared, a decision is only made if AIC and BIC concur; the decision is then based on that approximation which gives the smaller Bayes factor:

$$BF_{ij} = \frac{p(y|m_i)}{p(y|m_j)} \quad (33.11)$$

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 intrinsic connections or which inputs affect the system and where they enter. 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.

### 33.3 Practical example

The following example refers to the "attention to motion" data set available from the SPM web site. This data set was obtained by Christian Buchel and is described in [13]. Note that the data available from the web have been pre-processed using SPM99, therefore ensure 'defaults.analyze.flip' is set to 1 in the `spm_defaults.m` file before proceeding further. 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.

#### 33.3.1 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. **photic**: 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. See chapters 8 and 9 on how to do this. Here are the relevant details for this data set that you need to set up the GLM (this information can also be found at

[http://www.fil.ion.ucl.ac.uk/~wpenny/datasets/attention/README\\_GLM\\_DCM.txt](http://www.fil.ion.ucl.ac.uk/~wpenny/datasets/attention/README_GLM_DCM.txt) note this web site describes the analysis for SPM2!).

- The onsets for the conditions can be found in the file `factors.mat`. They are named phot (photic), mot (motion) 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.

#### 33.3.2 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 inclusively with the attention-contrast when extracting time series for V5. You should also compute the usual "effects of interest" F-contrast, this is needed for mean-correcting the extracted time series (see below). Here is 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. Mask with other contrasts: Yes
5. Choose the t-contrast for the attention condition.
6. Mask inclusively and choose a threshold of  $p \leq 0.05$  uncorrected.
7. Select the local maximum of a blob that looks V5-ish, e.g. -45/-81/-9 (by overlaying the activations onto the normalised structural image you should be able to identify V5 more easily).
8. Press the "VOI" button.
9. Name of region: V5
10. Adjust data for: effects of interest (this mean-corrects the time series)
11. VOI definition: sphere
12. 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. You can now proceed to select time series for V1 (using the "photic" contrast) and SPC (using the "attention" contrast). For this example, we selected -6/-84/-6 for V1 and -18/-57/66 for SPC.

### 33.3.3 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 [24]. 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 33.4, and can be made as follows:

1. Press the "DCM" 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`
6. Include Photic: Yes

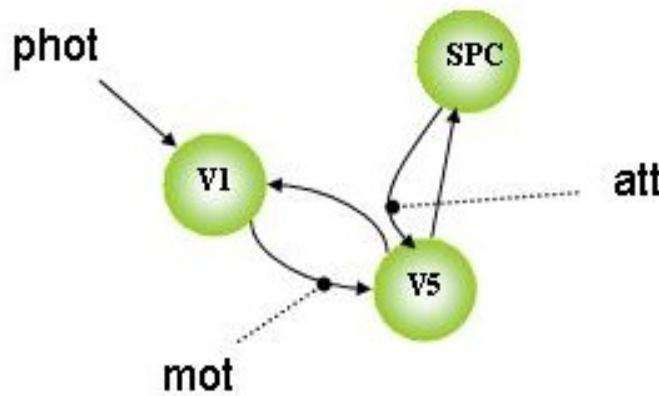


Figure 33.4: *DCM with attentional modulation of backwards connection. Dotted lines denote modulatory connections.*

7. Include Motion: Yes
8. Include Attention: Yes
9. Define the following intrinsic 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. 33.5.
10. Specify Photic as a driving input into V1. See Fig. 33.6
11. Specify Motion to modulate the connection from V1 to V5. See Fig. 33.7
12. Specify Attention to modulate the connection from SPC to V5. See Fig. 33.8
13. Specify slice timings for each area. This is a new option described in [45]. 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.

A polite "Thank you" completes the model specification process.

You can now estimate the model parameters, either by pressing the DCM button again and choosing "estimate", or by typing `spm_dcm_estimate('DCM_mod_bwd')` 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
4. Threshold: 0

Now you have multiple options, e.g. you can revisit the fit of the model ("Outputs") or look at the parameter estimates for the intrinsic connections ("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.A (intrinsic connections), DCM.B (modulatory inputs) and DCM.C (driving inputs).

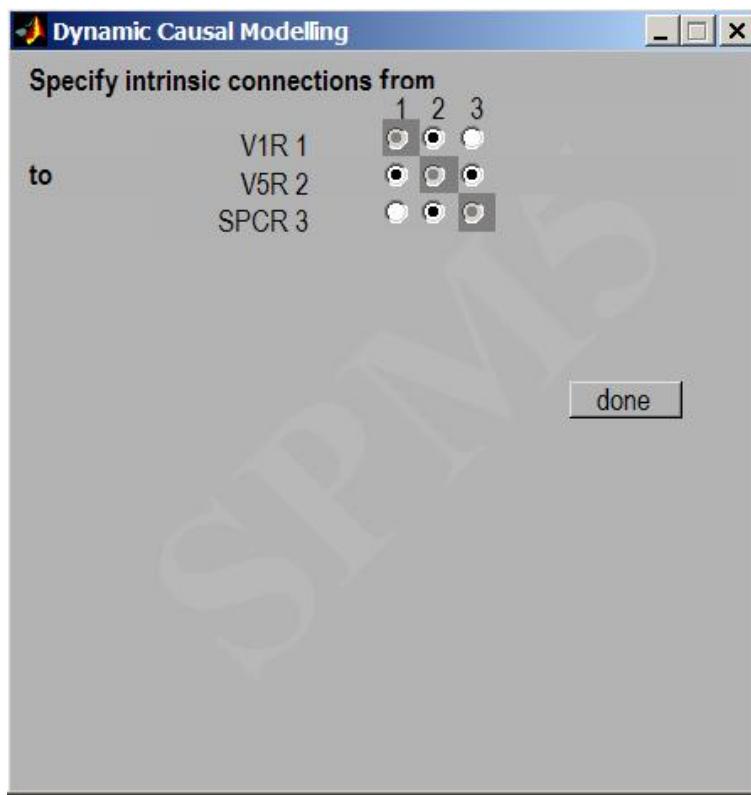


Figure 33.5: Filled circles define the structure of the intrinsic connections  $C$  such that eg. there are no connections from  $V1R$  to  $SPCR$  or from  $SPCR$  to  $V1R$ . See also Fig 33.4

### 33.3.4 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 differences 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 33.9. Once you have estimated this new model, you can perform a Bayesian model comparison as follows:

1. Press the "DCM" button.
2. Choose "compare".
3. Number of models to compare: 2
4. Select the two models, e.g. in the order `DCM_mod_bwd` and `DCM_mod_fwd`.

The graphics window will now show two plots of the model evidences, one based on the AIC approximation (upper panel) and another is based on the BIC approximation (lower panel). Fig. 33.10 shows this plot for AIC. Generally, a decision is only made if the two approximations concur see section 33.2 for details. In this example, AIC and BIC give identical results since the two models do not differ in complexity (we have only changed the position of one input). You can see that our second model is better than the first one. How much better precisely, is stated in the MATLAB command window where you find an exact breakdown of the different components (model fit and penalty terms for complexity) of the model comparison. In the example here the Bayes factor is 3.2532 in favour of the second model:

```
Model 1: C:\klaas\teaching\ExampleDataSet_Att2Mot\example_DCM\models\DCM_mod_bwd.mat
versus
```

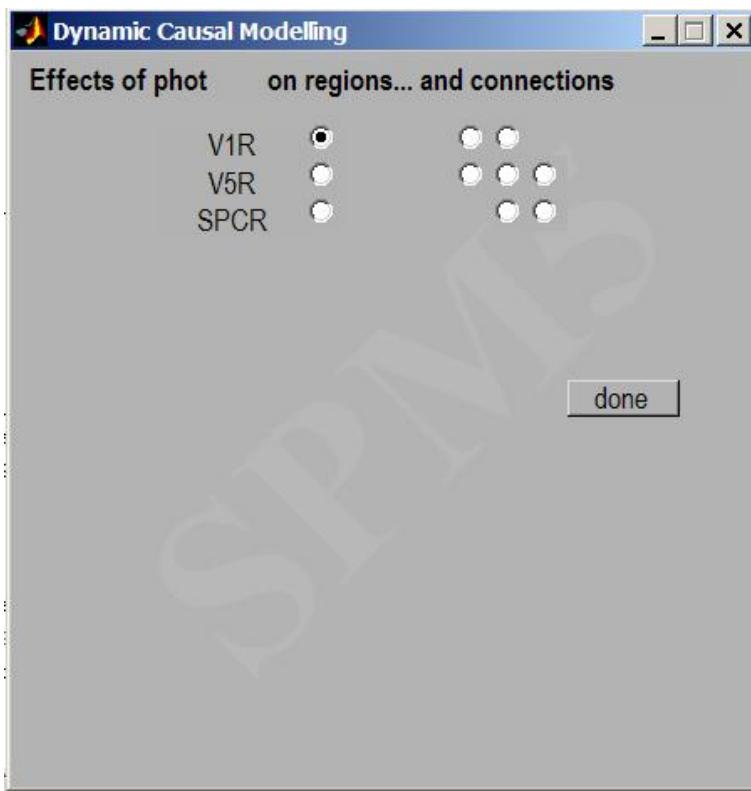


Figure 33.6: The filled circle specifies that the input ‘phot’ connects to region V1R. See also Fig 33.4

```
Model 2: C:\klaas\teaching\ExampleDataSet_Att2Mot\example_DCM\models\DCM_mod_fwd.mat
```

```
All costs are in units of binary bits
```

```
Region V1R: relative cost = -2.4170, BF= 5.3405
Region V5R: relative cost = 0.4477, BF= 0.7332
Region SPCR: relative cost = 3.6712, BF= 0.0785
AIC Penalty = 0.0000, BF = 1.0000
BIC Penalty = 0.0000, BF = 1.0000
AIC Overall = 1.7019, BF = 0.3074
BIC Overall = 1.7019, BF = 0.3074
```

```
Consistent evidence in favour of model 2
Bayes factor >= 3.2532
```

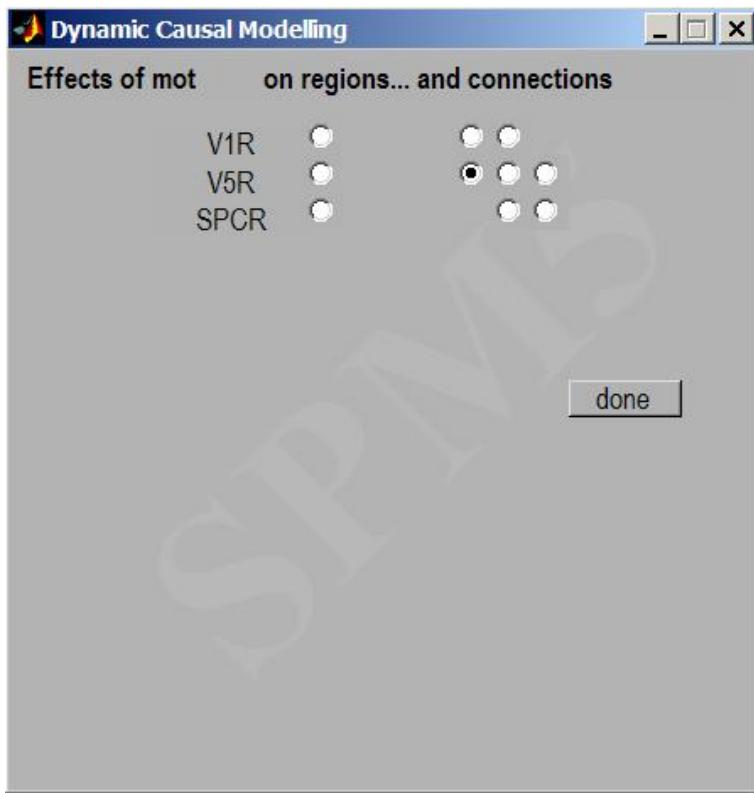


Figure 33.7: The filled circle indicates that the input variable ‘mot’ can modulate the connection from V1R to V5R. This specifies a ‘modulatory’ connection. See also Fig 33.4.

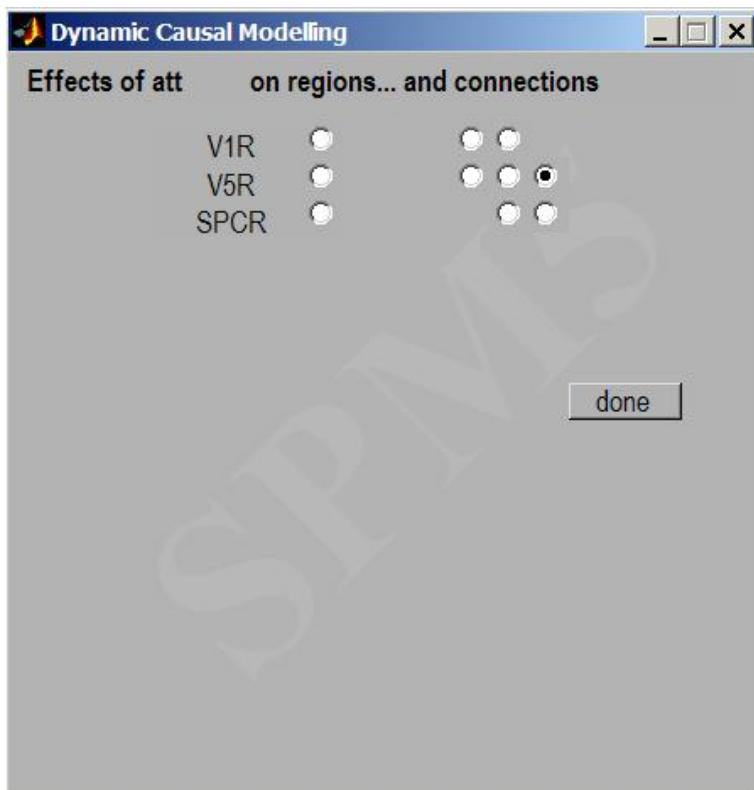


Figure 33.8: The filled circle indicates that attention can modulate the connection from SPCR to V5R. See also Fig 33.4

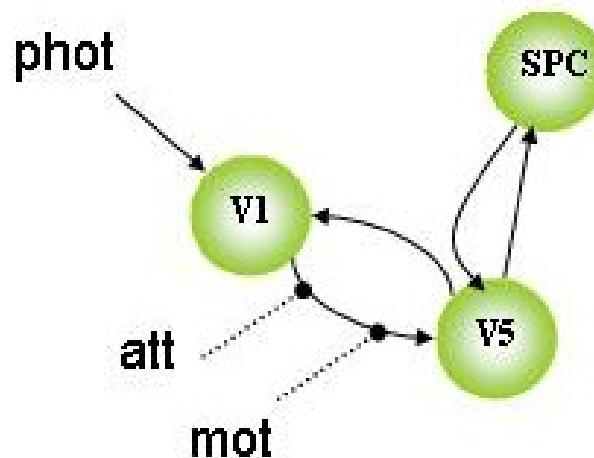


Figure 33.9: *DCM with attentional modulation of forwards connection. Dotted lines denote modulatory connections.*

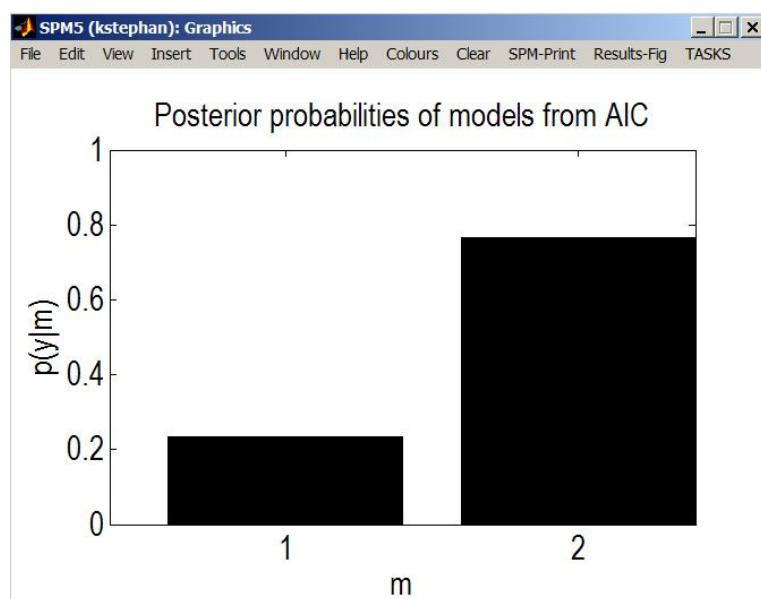


Figure 33.10: *Model 2 (shown in Fig 33.9) is preferred to model 1 (shown in Fig 33.4).*

# Chapter 34

## Multimodal face-evoked responses

### 34.1 Overview

This dataset contains EEG, MEG, functional MRI and structural MRI data on the same subject within the same paradigm, which allows a basic comparison of faces versus scrambled faces.

It can be used to demonstrate, for example, 3D source reconstruction of various electrophysiological measures of face perception, such as the "N170" evoked response (ERP) recorded with EEG, or the analogous "M170" evoked field (ERF) recorded with MEG. These localisations are informed by the anatomy of the brain (from the structural MRI) and possibly by functional activation in the same paradigm (from the functional MRI).

The demonstration below involves localising the N170 using a distributed source method (called an "imaging" solution in SPM) analogous to "weighted minimum L2-norm". The data can also be used to explore further effects, e.g. induced effects (Friston et al, 2006), effects at different latencies, or the effects of adding fMRI constraints on the localisation.

The EEG data were acquired on a 128 channel ActiveTwo system; the MEG data were acquired on a 151 channel CTF Omega system; the sMRI data were acquired using a phased-array headcoil on a Siemens Sonata 1.5T; the fMRI data were acquired using a gradient-echo EPI sequence on the Sonata. The dataset also includes data from a Polhemus digitizer, which are used to coregister the EEG and the MEG data with the structural MRI.

Some related analyses of these data are reported in Henson et al (2005a, 2005b, 2007), Kiebel and Friston (2004) and Friston et al (2006; in press-a).

The analysis below is best done in Matlab7, but all mat files should be in a format readable by Matlab6.5.

### 34.2 Paradigm and Data

The basic paradigm involves randomised presentation of at least 86 faces and 86 scrambled faces (Figure 34.1), based on Phase 1 of a previous study by Henson et al (2003). The scrambled faces were created by 2D Fourier transformation, random phase permutation, inverse transformation and outline-masking of each face. Thus faces and scrambled faces are closely matched for low-level visual properties such as spatial frequency power density. Half the faces were famous, but this factor is collapsed in the current analyses. Each face required a four-way, left-right symmetry judgment (mean RTs over a second; judgments roughly orthogonal to conditions; reasons for this task are explained in Henson et al, 2003). The subject was instructed not to blink while the fixation cross was present on the screen.

#### 34.2.1 Structural MRI

The T1-weighted structural MRI of a young male was acquired on a 1.5T Siemens Sonata via an MDEFT sequence with resolution  $1x1x1mm^3$  voxels, using a whole-body coil for RF transmission and an 8-element phased array head coil for signal reception.

The images are in Analyze format in the sMRI sub-directory, consisting of two files:

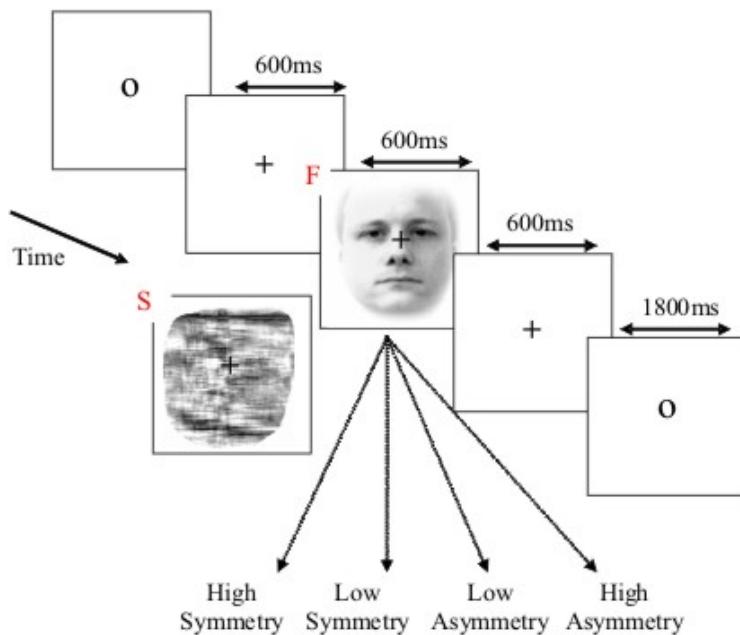


Figure 34.1: One trial in the experimental paradigm: Trials involved either a Face (F) or Scrambled face (S).

`sMRI/sMRI.img`  
`sMRI/sMRI.hdr`

The structural was manually-positioned to match roughly Talairach space, with the origin close to the Anterior Commissure, which produced the associated SPM Matlab file:

`sMRI/sMRI.mat`

The approximate position of 3 fiducials within this MRI space - the nasion, and the left and right peri-auricular points - are stored in the file:

`sMRI/smri_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). It doesn't matter that the positions are approximate, because more precise coregistration is done via digitised surfaces of the scalp ("head shape functions") that were created using the Polhemus 3D digitizer.

### 34.2.2 EEG data

The EEG data were acquired on a 128-channel ActiveTwo system, sampled at 2048 Hz (subsequently downsampled to 200Hz to reduce filesize), plus electrodes on left earlobe, right earlobe, and two bipolar channels to measure HEOG and VEOG. The data were referenced to the average of the left and right earlobes (for consistency with Henson et al, 2003). The 128 scalp channels are named: 32 A (Back), 32 B (Right), 32 C (Front) and 32 D (Left).

The original data were converted into SPM M/EEG format and epoched from -200ms to +600ms post-stimulus (baseline-corrected from -200 to 0ms), ie 161 samples:

`EEG/e_eeg.mat`  
`EEG/e_eeg.dat`

(using the `bdf_setup.mat` channel template provided with SPM5 in the EEGtemplates subdirectory). Other details about the data can be examined by typing:

```
D = spm_eeg_ldata
```

and selecting the `e_meg.mat` file. This will show the contents of the structure "D" that is loaded into the Matlab workspace, the various fields of which can be explored. Note that the data values themselves are memory-mapped from the `e_eeg.dat` file to the field D.data (e.g, D.data(1,2,3) returns the field strength in the first sensor at the second sample point during the third trial).

You will see that there are 344 events (D.Nevents), consisting of 172 faces (event code 1) and 172 scrambled faces (event code 2), which are randomly intermixed (see D.events.code)<sup>1</sup>. If you type D.channels.name, you will see the order and the names of the channels.

The EEG directory also contains a Polhemus sub-directory with the following files:

```
EEG/Polhemus/eeg_fids.mat
EEG/Polhemus/eeg_sens_loc.mat
EEG/Polhemus/eeg_hsf.mat
```

All files contain matrices, the three columns of which code location in a right-handed 3D space, the axes of which conform to the Talairach space, i.e, the first column is x (left-right), the second is y (anterior-posterior) and the third is z (inferior-superior). The units are mm.

The `eeg_fids.mat` file contains the position of 3 fiducial markers that were placed approximately on the nasion and peri-auricular points and digitised by the Polhemus digitizer. The digitizer was also used to locate the position of each electrode (in the `eeg_sens_loc.mat` file), and to trace out many points along the surface of the subject's scalp and nose (the "head shape function" in the `eeg_hsf.mat` file). Later, we will coregister the fiducial points and the head shape to map the electrode positions in the "Polhemus space" to the "MRI space".

### 34.2.3 MEG data

The MEG data were acquired on a 151 channel CTF Omega system, using second-order axial gradiometers sampled at 625 Hz (subsequently downsampled to 200Hz to reduce filesize). The original data were converted into SPM MEEG format and epoched from -200ms to +600ms post-stimulus (i.e, baseline-corrected from -200ms to 0ms), ie 161 samples:

```
MEG/e_meg.mat
MEG/e_meg.dat
```

The channel template for these data is also provided:

```
MEG/CTF151_setup.mat
```

(which may need to be copied to the EEGtemplates sub-directory of your local SPM5 installation directory, if not already there). The MEG data also contains a Polhemus sub-directory with the following files:

```
MEG/Polhemus/meg_fids.mat
MEG/Polhemus/meg_sens_loc.mat
MEG/Polhemus/meg_sens_or.mat
MEG/Polhemus/meg_hsf.mat
```

which are analogous to the corresponding MEG files described in the previous section<sup>2</sup>. More specifically, the `meg_fids.mat` contains the position of 3 "locator coils", positioned close to the fiducials<sup>3</sup>, the locations of which are measured by the CTF machine, and used to define the coordinates (in "CTF space") for the location of the 151 sensors (in the `meg_sens_loc.mat` file) and their (axial gradiometer) orientations (`meg_sens_or.mat`). The same three locator coils were digitised by a Polhemus digitizer outside the MEG machine, together with the head shape, to

<sup>1</sup>These data were actually concatenated from two separate runs on the same subject (using spm-eeg-merge), which is why there are twice as many events as with the MEG and fMRI data.

<sup>2</sup>These matrices are transformations from the original CTF / Polhemus files - in which x codes anterior-posterior and y codes left-right - i.e, a 90 degree clockwise rotation about the z-axis.

<sup>3</sup>Unlike the MRI and EEG data, these fiducials were not precisely the nasion and peri-auricular points. However, given that the coregistration of the MEG and MRI data is based mainly on the headshape (see later), this inaccuracy in the MEG fiducials does not matter.

define the "Polhemus space". Subsequently, the fiducials in the Polhemus space were coregistered with the fiducials in the CTF space, and the resulting rigid-body transformation applied to the Polhemus head shape. Thus the coordinates in all four files above are in alignment in the CTF space, which will subsequently be transformed into the "MRI space".

### 34.2.4 fMRI data

The fMRI data were acquired using a Trajectory-Based Reconstruction (TBR) gradient-echo EPI sequence (Josephs et al, 2000) on a 1.5T Sonata. There were 32, 3mm slices of 3x3 mm<sup>2</sup> pixels, acquired in a sequential descending order with a TR of 2.88s. There are 215 images in the 'Scans' sub-directory (5 initial dummy scans have been removed), each consisting of an Analyze image and header file:

```
fMRI/Scans/fM*.img  
fMRI/Scans/fM*.hdr
```

Also provided are the onsets of faces and scrambled faces (in units of scans) in the Matlab file:

```
fMRI/onsets.mat
```

and the SPM "Job" files (see Section 34.6):

```
fMRI/realign_job.mat  
fMRI/slicetime_job.mat  
fMRI/smooth_job.mat  
fMRI/stats_job.mat
```

## 34.3 Getting Started

You need to start SPM5 and toggle "EEG" as the modality (bottom-right of SPM main window), or start SPM5 with `spm eeg`. You will also need to 1) copy the MEG template file (`CTF151_setup.mat`) to the EEGtemplates sub-directory within your SPM5 installation directory, if it is not already there (see section 34.2.3 above), and 2) ensure this EEGtemplates directory is on your Matlab path.

## 34.4 EEG analysis

### 34.4.1 Preprocessing the EEG data

\* Change directory to the EEG subdirectory (either in Matlab, or via the "CD" option in the SPM "Utils" menu)

\* Press 'Artefacts', select the `e_eeg.mat` file, press 'no' to the 'read own artefact list?' question, 'no' to 'robust average?', 'yes' to 'threshold channels?', and enter 200 for the threshold

This will detect trials in which the signal recorded at any of the channels exceeds 200 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<sup>4</sup>. 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).

In this case, the Matlab window will show:

```
There isn't a bad channel.
```

```
45 rejected trials: [5 38 76 82 83 86 87 88 89 90 92 93 94 96 98 99  
100 101 104 105 106 107 108 112 117 118 119 120 122 124 126 130 137  
139 159 173 221 266 268 279 281 292 293 298 326]
```

---

<sup>4</sup>Channel-specific thresholds can be used by entering 130 thresholds, one per EEG/EOG channel, with a value of Inf for those channels that you do not want to threshold.

(leaving 299 valid trials). A new file will also be created, `ae_eeg.mat` (in which these artefacts are coded in the fields `D.events.reject` and `D.channels.thresholded`).

At this point, you may want to look at the data. Press "Display: M/EEG", and select the `ae_eeg.mat` file. After a short delay, the Graphics window should show the mean ERP (for trial 1) at each of the 130 channels (as in Figure 34.2). You can click on one of the channels (e.g, VEOG, on the top right of the display) to get a new window with the data for that channel expanded. You can alter the scaling or trial number using the sliders on the bottom of the Graphics window, or select a subset of channels to display by pressing the 'Channels' button.

### 34.4.2 Basic ERPs

\* Press the 'Averaging' button and select the `ae_eeg.mat` file. After a few moments, the matlab window will echo:

`ae_eeg.mat`: Number of replications per contrast: average 1: 151 trials, average 2: 148 trials (artefact trials are excluded from averaging) and a new file will be created in the MEG directory called `mae_eeg.mat` ("m" for "mean").

\* Press the 'Filtering' button, select the `mae_eeg.mat` file, select 'lowpass', and enter 40 (Hz) as the cutoff. This smooths the data to 40Hz, producing the file `fmae_eeg.mat` (using zero-phase-shift forward and reverse digital filtering with a 5th-order Butterworth filter)<sup>5</sup>.

You can display the mean ERPs using the "Display: M/EEG" menu option again. Once you have done so, press the "channels" button in the Graphics window, then "Deselect all", and then click only, eg channels 'a1', 'd7', 'a12', 'b9' and 'c7'. (You can save these selections as a file, and use this file to display only a subset of channels in future). After pressing "ok", you will now only see these 5 channels (which will also make the display much faster!). Once you hold SHIFT and select trial-type 2, you should see something like Figure 34.3.

\* Select "Contrast" from the "Other..." pulldown menu on the SPM window (or type `spm_eeg_weight_epoch` in the Matlab window). This function creates linear contrasts of ERPs/ERFs. Select the `fmae_eeg.mat` file, and enter  $[1 - 1; 1/21/2]$  as the contrast matrix. Press "no" to the question "weight by num replications". This will create new file `mfmae_eeg.mat`, in which the first trial-type is now the differential ERP between faces and scrambled faces, and the second trial-type is the average ERP for faces and scrambled faces.

To look at the differential ERP, again press 'Display: M/EEG', and select the `mfmae_eeg.mat` file. After a short delay, the Graphics window should show the ERP for each channel (for trial-type 1). Hold SHIFT and select trial-type 2 to see both conditions superimposed. Then click on one of the channels (e.g, 'B9' on the bottom right of the display) to get a new window with the data for that channel expanded, as in Figure 34.4.

The red line shows the average ERP evoked by faces and scrambled faces (at this occipitotemporal channel). A P1 and N1 are clearly seen. The blue line shows the differential ERP between faces and scrambled faces. This is approx zero around the P1 latency, but negative around the N1 latency. The latter likely corresponds to the "N170" (Henson et al, 2003). We will try to localise the cortical sources of the P1 and N170 in Section 34.4.4.

To see the topography of the differential ERP, press the "topography" button in the main graphics window, enter 165ms for the latency, and select "3D", to produce Figure 34.5. Choose the rotate3D cursor to surf.

### 34.4.3 3D SPMs (Sensor Maps over Time)

One novel 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.

---

<sup>5</sup>Note that (lowpass) filtering short epochs like this is not necessarily a good idea, since ringing or "end-effects" can result at the start and end of the epoch. Filtering is normally better performed on continuous data (or longer epochs). The filtering performed here is simply to demonstrate the option and for display purposes (though the averaging process also tends to act like a lowpass filter anyway).

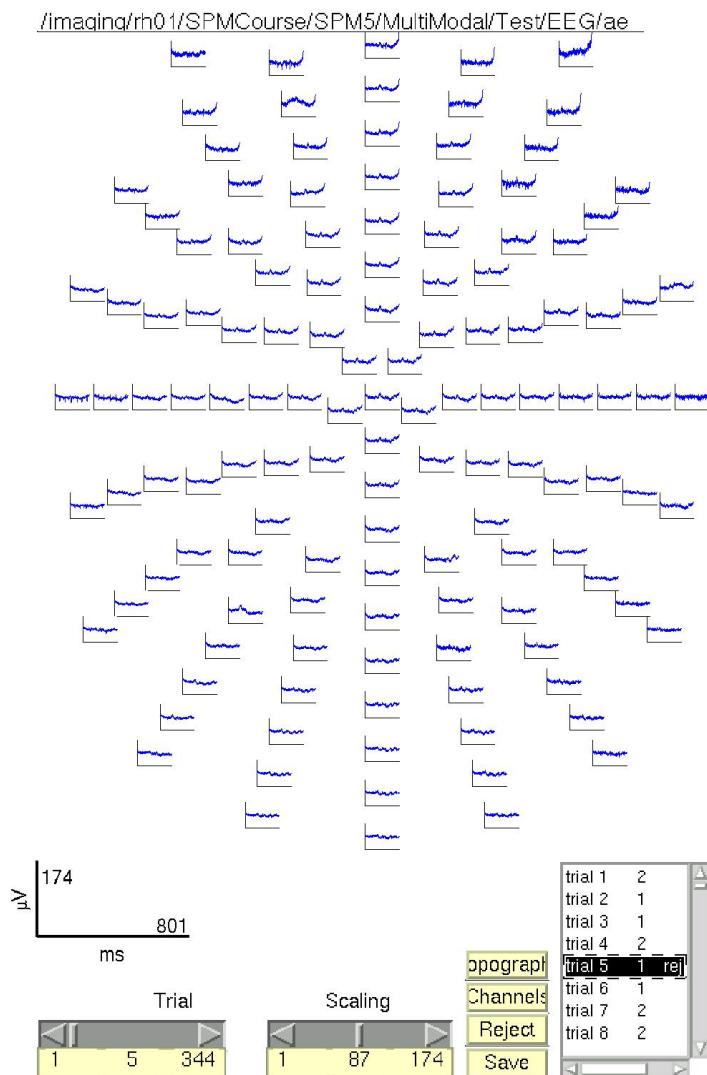


Figure 34.2: *SPM Display window for trial 5 for all 128 EEG plus 2 EOG channels (top left and top right) in ae-eeg.mat. Trial 5 is marked as an artefact because the VEOG channel (top right) exceeds the user-specified threshold of 200uV, most likely reflecting the onset of a blink towards the end of the epoch.*

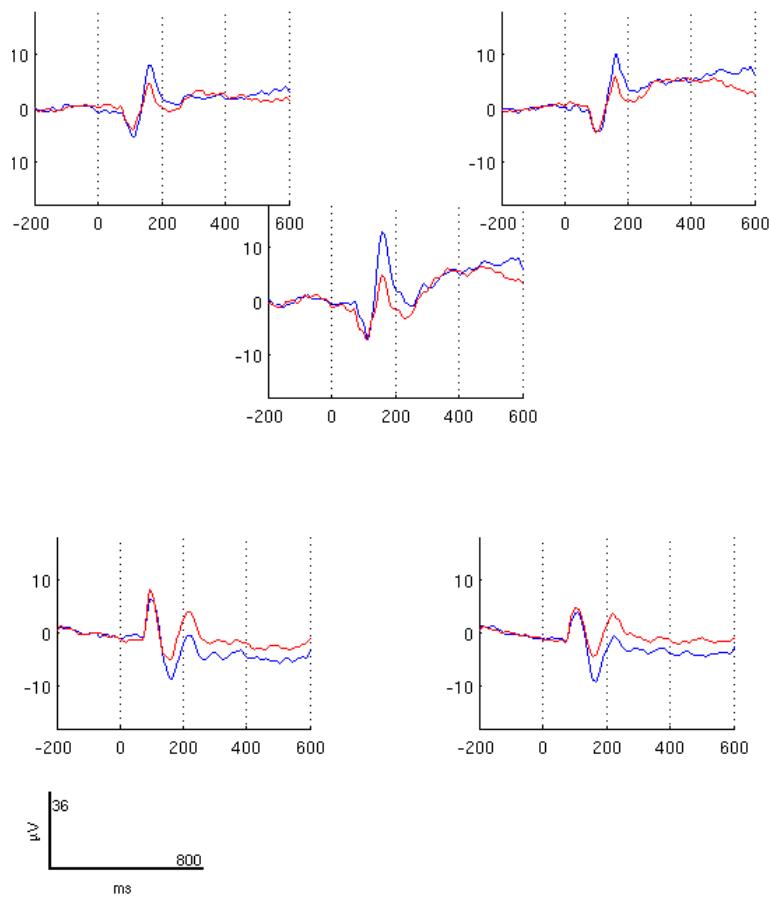


Figure 34.3: *SPM Display window for smoothed, average ERPs for faces (blue) and scrambled faces (red) for 5 selected channels in fmae-eeg.mat.*

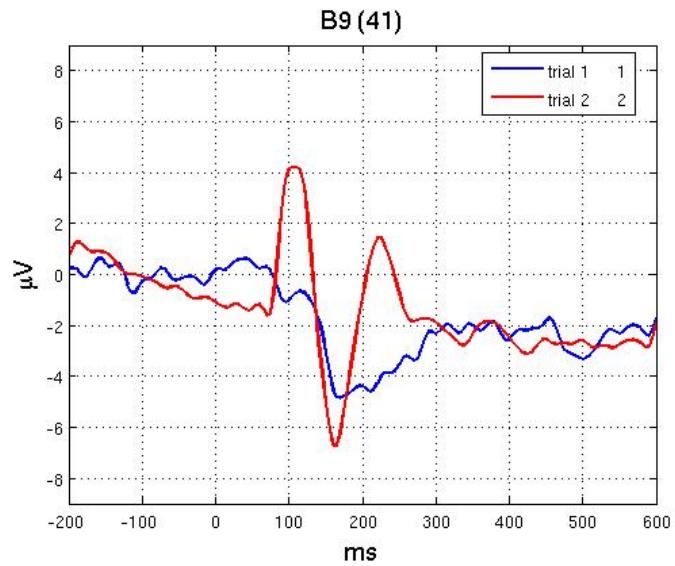


Figure 34.4: Average (red) and differential (blue) ERPs for faces and scrambled faces at channel B9 in *mfmae-eeg.mat*.

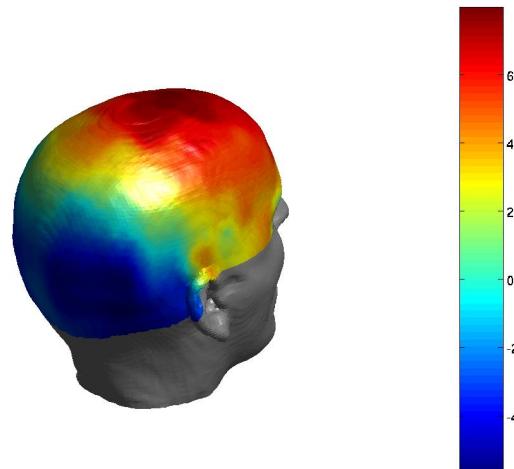


Figure 34.5: 3D topography for faces minus scrambled faces at 165ms.

That correction uses Random Field Theory, which takes into account the spatial correlation across pixels (i.e, that the tests are not independent). This kind of analysis is described earlier in the SPM manual, where a 1st-level design is used to create the images for a given weighting across timepoints of an ERP/ERF, and a 2nd-level design can then be used to test these images across subjects.

Here, we will consider a 3D example, where the third dimension is time, and test across trials within the single subject. We first create a 3D image for each trial of the two types, with dimensions MxMxS, where S=161 is the number of samples. We then take these images into an unpaired t-test across trials (in a 2nd-level model) to compare faces versus scrambled faces. 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 faces and scrambled faces would emerge<sup>6</sup>.

\* Select the "mat-2-3Dimage" option in the "Other..." menu in the Matlab window, and select the `ae_eeg.mat` file. You will then be prompted for "output image dimensions", for which you can accept the default of 32 (leading to a 32x32 pixel space), and a pixel dimension, which you can change to 5 (this is rather arbitrary, but will make the images easier to view). It will then ask whether you want to interpolate or mask out bad channels, for which you can select interpolate (though it will make no difference here because there are no bad channels).

This will take some time as it writes out an image for each trial (except rejected trials), in a new directory called `ae_eeg`, which will itself contain two subdirectories, one for each trialtypes. In each trialtypes subdirectory there will be image and header files for each non-rejected trial of that type, e.g. `trial02.img/hdr`. You can press "Display: images" to view one of these images - it will have dimensions 32x32x161(x1), with the origin set at [16 16 40] (where 40 samples is 0ms), as in Figure 34.6.

To perform statistics on these images, first create a new directory, eg. `mkdir XYTstats`.

\* Then press the "specify 2nd level" button, select "two-sample t-test" (unpaired t-test), and define the images for "group 1" as all those in the subdirectory "triaitype1" (using right mouse, and "select all") and the images for "group 2" as all those in the subdirectory "triaitype2". Finally, specify the new XYTstats directory as the output directory, and press "run"<sup>7</sup>.

This will produce the design matrix for a two-sample t-test.

\* Then press "Estimate", and when it has finished, press "Results" and define a new F-contrast as [1 -1] (for help with these basic SPM functions, see eg. chapter 26). Keep the default contrast options, but threshold at  $p < .05$  FWE corrected for the whole "image". Then press "volume", and the Graphics window should now look like that in Figure 34.7 (ignore the outline of the brain in the MIP!).

This will reveal "regions" within the 2D sensor space and within the -200ms to 600ms epoch in which faces and scrambled faces differ reliably, having corrected for multiple F-tests across pixels/time. There are a number of such regions, but we will concentrate on the first two (largest ones), with cluster maxima of [25 -55 200] and [10 5 160]. An F-test was used because the sign of the difference reflects the polarity of the ERP difference, which is not of primary interest (and depends on the choice of reference; see footnote 6). Indeed, if you plot the contrast of interest from the cluster maxima, you will see that the difference is negative for the first cluster (which is located posteriorly) but positive for the second cluster (which is more central, close to Cz). This is consistent with the polarity of the differences in Figure 34.3<sup>8</sup>.

If one had more constrained a priori knowledge about where and when the N170 would appear, one could perform an SVC based on, for example, a box around posterior channels and between 150 and 200ms poststimulus.

---

<sup>6</sup>Note that the 2D location in sensor space for EEG will depend on the choice of reference channel.

<sup>7</sup>Note that we can use the default "nonsphericity" selections, i.e, that the two trial-types may have different variances, but are uncorrelated.

<sup>8</sup>With a reference similar to the current earlobes, the former likely corresponds to the "N170", while the latter likely corresponds to the "VPP" (though we have no evidence here for a dissociation between them).

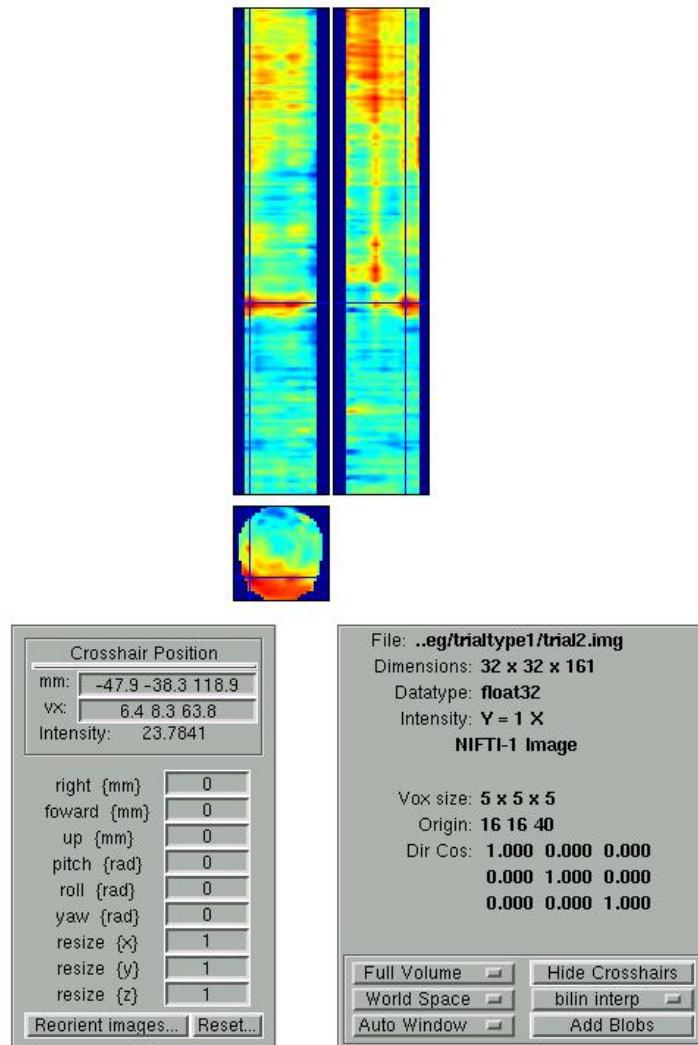


Figure 34.6: 3D image for trial 2 of ae-eeg.mat. The bottom image is a square x-y space interpolated from the flattened electrode locations (at one point in time). The two top images are sections through x and y respectively, now expressed over time (vertical (z) dimension). (Colormap changed to 'jet').

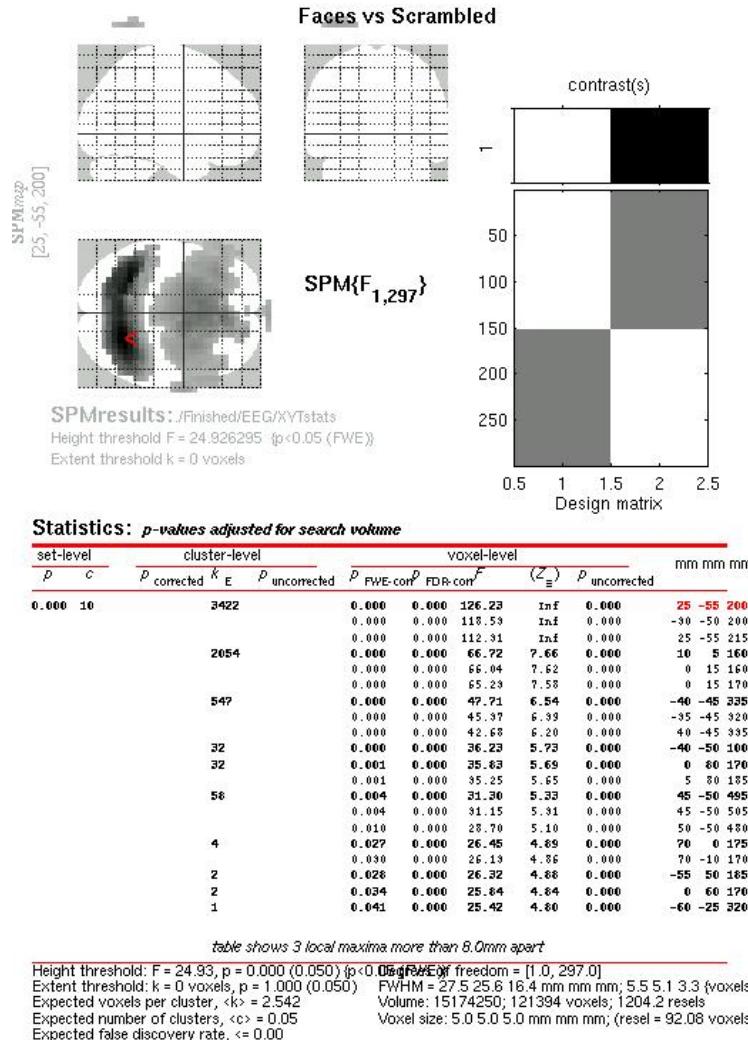


Figure 34.7: 3D sensor-time SPMF at  $p < .05$  FWE corrected for the amplitude difference between face and scrambled face trials. Note that the brain outline in the MIP should be ignored. The x, y coordinates refer to arbitrary units in the  $32 \times 32$  electrode plane (origin = [16 16]); the z coordinate refers to peristimulus time in ms (to the nearest sampling of 5ms).

### 34.4.4 3D "imaging" reconstruction

Here we will demonstrate a distributed source reconstruction of the N170 differential evoked response between faces and scrambled faces, using a grey-matter mesh extracted from the subject's MRI, and an L2-norm method in which multiple constraints on the solution can be imposed (Phillips et al, 2002; Mattout et al, 2005; Henson et al, 2007; Friston et al, in press-a).

\* Press the '3D source reconstruction' button, and press the "load" button at the top of the new window. Select the `mfmae_eeg.mat` file and type a label (eg "N170") for this analysis<sup>9</sup>.

\* Press the 'MRI' button, select the `smri.img` file within the sMRI sub-directory, press the "Imaging" button, and select 3000 for the number of vertices in the mesh...

The "imaging" option corresponds to a distributed source localisation, where current sources are estimated at a large number of fixed points (3000 here) within a cortical mesh, rather than approximated by a small number of equivalent dipoles (the ECD option). The imaging or distributed approach is better suited for group analyses and probably for later components; the ECD approach may be better suited for very early sensory components (when only small parts of the brain are active), or for DCM models of a small number of regions (Kiebel et al, 2006).

This will take some time while the MRI is segmented (and normalisation parameters determined). This will create the usual files, i.e, `c1/c2/c3smri.img/hdr`, for grey/white/CSF respectively, `msmri.img/hdr` for the attenuation-corrected image, and the normalisation and inverse normalisation parameters (`mri_vbm_sn_1.mat` and `smri_vbm_inv_sn_1.mat` respectively) in the sMRI directory (see Chapter 5 for further details).

This process will also create binary images of the cortex, inner skull surface and scalp, which are then used to create meshes (of 2002 vertices) for these surfaces, stored in the following files:

```
sMRI/smri_cortex.img
sMRI/smri_iskull.img
sMRI/smri_scalp.img
```

When meshing has finished, the cortex (blue), inner skull (red) and scalp (orange) meshes will also be shown in the Graphics window. The field `D.inv{1}.mesh` will be updated in matlab. Press "save" in top right of window to update the corresponding mat file on disk.

Note that the cortical mesh (and the distances within the mesh) are not created directly from the segmented MRI (like the skull and scalp meshes), but rather are determined from a template cortical mesh in MNI space via inverse spatial normalisation (Mattout et al, in press).

\* Press the 'Co-register' button, respond "no" to the 'Read Polhemus?' question (which is if you want to read in a Polhemus file directly), and then select the following files in response to each prompt (pressing "yes" to the 'Use headshape file' prompt):

```
EEG/Polhemus/eeg_sens_loc.mat
EEG/Polhemus/eeg_fids.mat
EEG/Polhemus/eeg_hsf.mat
sMRI/smri_fids.mat
```

This stage coregisters the EEG sensor positions with the structural MRI and cortical mesh, via an approximate matching of the fiducials in the two spaces, followed by a more accurate surface-matching routine that fits the head-shape function (measured by Polhemus) to the scalp that was created in the previous meshing stage via segmentation of the MRI.

When coregistration has finished, the field `D.inv{1}.datareg` will be updated in matlab. Press "save" in top right of window to update the corresponding mat file on disk. Finally, a figure like that in Figure 34.8 will also be produced, which you can rotate with the mouse (using the Rotate3D Matlab Menu option) to check all sensors.

\* Press 'Forward Model', and select "3 Berg".

This will create a forward model (lead field matrix) based on a three sphere model (using a subset of BrainStorm functions, packaged with SPM<sup>10</sup>). The Matlab window will output:

---

<sup>9</sup>Note that no new M/EEG files are created during each stage of the 3D reconstruction; rather, each step involves updating of the cell-array field `D.inv`, which will have one entry per analysis performed on that dataset (e.g, `D.inv{1}` in this case).

<sup>10</sup>Brainstorm is available from <http://neuroimage.usc.edu/ResearchMEGEEGBrainStorm.html>

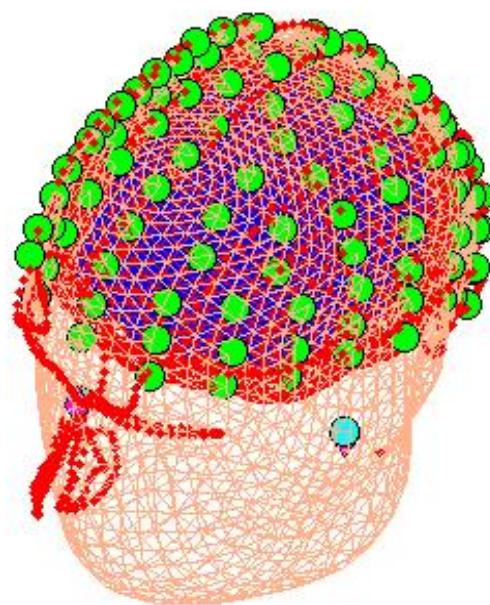


Figure 34.8: Graphical output of Co-registration of EEG data, showing (upper panel) cortex (blue), inner skull (red) and scalp (black) meshes, electrode locations (green), MRI/Polhemus fiducials (cyan/magenta), and headshape (red dots).

```

Scalp best fitting sphere computed (in 11 iterations)
Centre = [0.0001 -0.0218 0.0027] (Radius = 0.0774)
Computing EEG "BERG" Parameters. . .
Computing EEG "BERG" Parameters -> DONE

Computing the Image Gain Matrices. . .
Foward model complete - thank you

```

and a picture of the best-fitting sphere to the inner skull surface will be shown in the Graphics window (this defines the centre of the concentric spheres). The leadfield matrix (with source orientations fixed as normal to the cortical surface) is stored in the file:

**smri\_SPMgainmatrix\_1.mat**

(The file **smri\_SPMgainmatxyz\_1.mat** stores a version with three orthogonal orientations per source location).

\* Press 'Invert', select "Classical" (i.e, a distributed solution rather than DCM; Kiebel et al, 2006), select "yes" to include all conditions (i.e, both the differential and common effects of faces and scrambled faces), press "MSP" for the type of inversion, and then "Standard".

MSP stands for "Multiple Sparse Priors", and has been shown to be superior to standard minimum norm (the alternative MNM option) or a maximal smoothness solution (like LORETA; the COH option) - see Friston et al (in press-a). Note that by default, MSP uses a "Greedy Search" (Friston et al, in press-b), though the standard ReML (as used in Friston et al, in press-a) can be selected as a hidden option.

The "Standard" option uses default values for the MSP approach (to customise some of these parameters, press "Customise" instead).

\* Press "save" to save the results. You can now explore the results via the 3D reconstruction window. If you type 165 into the box in the bottom right (corresponding to the time in ms) and press "mip", you should see an output like in [34.9](#). This fit explains approx 97% of the data.

Note the hot-spots in the fusiform. The timecourses come from the peak voxel. The red line shows the condition currently being shown (corresponding to the "Condition 1" toggle bar in the reconstruction window); the grey line(s) will show all other conditions. Condition 1 is the differential evoked responses for faces vs scrambled; if you press the "condition 1" toggle, it will change to Condition 2 (average evoked response for faces and scrambled faces), then press "mip" again and the display will update (note the colours of the lines have now reversed from before, with red now corresponding to average ERP).

If you toggle back to condition 1 and press "movie", you will see the changes in the source strengths for the differential response over peristimulus time (from the limits 0 to 300ms currently chosen by default).

If you press "render" you can get a very neat graphical interface to explore the data (the buttons are fairly self-explanatory). However, we will concentrate on how one might perform statistics (eg with more subjects in a group analysis).

\* Press the "Window" button in the reconstruction window, enter "150 200" as the timewindow of interest and keep "0" as the frequency band of interest (0 means all frequencies). The Graphics window will then show the mean activity for this time/frequency contrast (and the contrast itself; note additional use of a Hanning window).

\* If you then press "Image", press "12" for the smoothing kernel, and SPM will write 3D Nifti images corresponding to the above contrast for each condition:

```

w_mfmae_eeg_1_1.nii
w_mfmae_eeg_1_2.nii
sw_mfmae_eeg_1_1.nii
sw_mfmae_eeg_1_2.nii

```

Note that the first two images are unsmoothed (but normalised); the latter two are smoothed by a 12mm isotropic Gaussian kernel. The last number in the file name refers to the condition number; the penultimate number refers to the reconstruction number (ie the number in red in the reconstruction window, i.e, D.val, here 1).

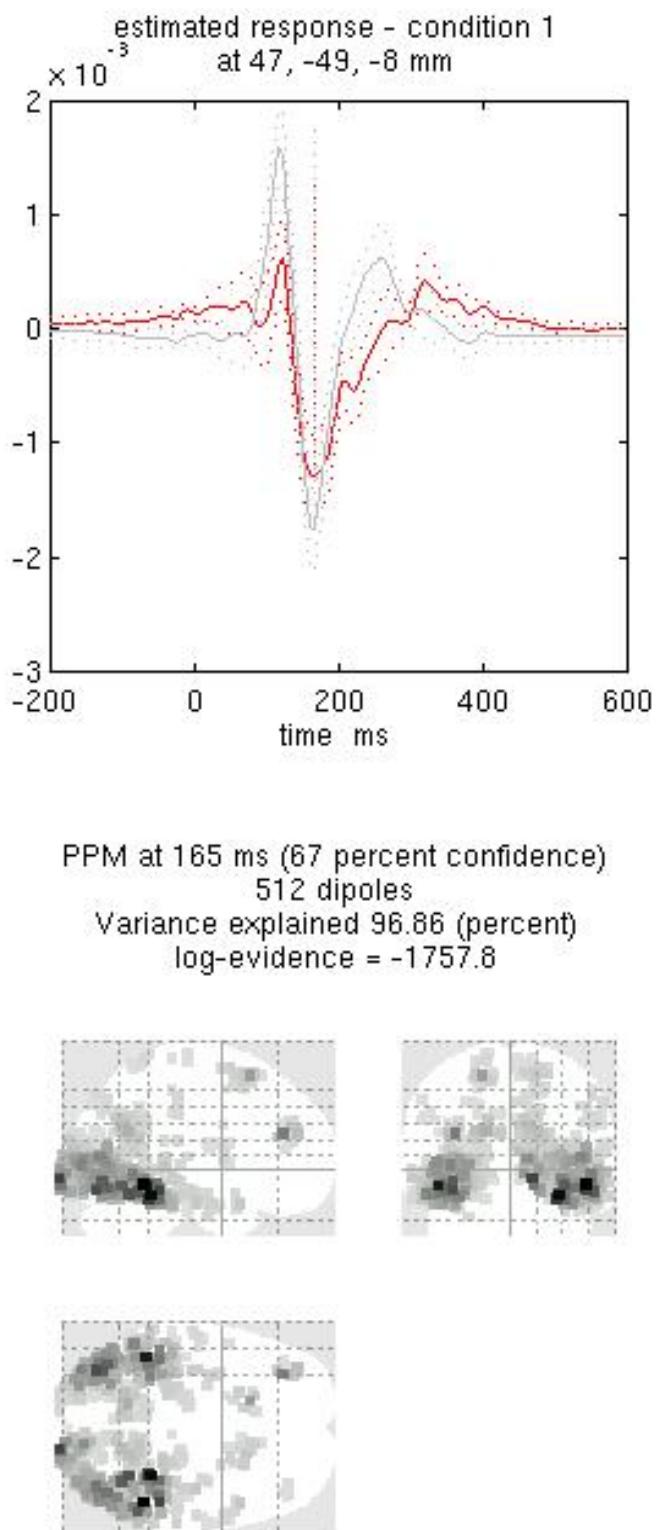


Figure 34.9: Graphical output of an MSP estimation of the differential ERP between faces and scrambled faces at 165ms.

The smoothed results for Condition 1 (i.e. the differential evoked response for faces vs scrambled faces) will also be displayed in the Graphics window, together with the normalised structural. Note that the solution image is in MNI (normalised) space, because the use of a canonical mesh provides us with a mapping between the cortex mesh in native space and the corresponding MNI space.

You can also of course view the image with the normal SPM "Display:image" option, and locate the coordinates of the "hotspots" in MNI space. Note that these images contain RMS (unsigned) source estimates (see Henson et al, 2007).

You could also explore the other inversion options, like COH and MNM, which you will note give more superficial solutions (a known problem with standard minimum norm). To do this quickly (without repeating the MRI segmentation, coregistration and forward modelling), press the "new" button in the reconstruction window, which by default will copy these parts from the previous reconstruction.

## 34.5 MEG analysis

### 34.5.1 Preprocessing the MEG data

\* Change directory to the MEG subdirectory (either in Matlab, or via the "CD" option in the SPM "Utils" menu)

\* Press 'Artefacts', select the `e_meg.mat` file, press 'no' to the 'read own artefact list?' question, but 'yes' to 'robust average?' and select the default 'offset weighting function' (3) and default FWHM residual smoothing (20), and 'no' to 'threshold channels?'

This will take a while. The new file produced, `ae_meg.mat`, will contain the same data, but a new field of "D.weights" will be created. These weights will be applied during the next step of averaging (see Kilner et al, in prep.):

\* Press the 'Averaging' button and select the `ae_meg.mat` file. After a few moments, the matlab window will echo:

```
e_meg.mat: Number of replications per contrast:  
average 1: 86 trials, average 2: 86 trials
```

and a new file will be created in the MEG directory called `mae_meg.mat` ("m" for "mean")

\* Press the 'Filtering' button, select the `mae_eeg.mat` file, select 'lowpass', and enter 40 (Hz) as the cutoff. This smooths the data to 40Hz, producing the file `fmae_eeg.mat` (see again footnote 5 about filtering).

As before, you can display these data by "Display: M/EEG" and selecting the `fmae_eeg.mat`. Hold SHIFT and select trial-type 2 with the mouse in the bottom right of the window to see both conditions superimposed (as Figure 34.10).

You can also press this 'Channels' button and in the new window, "deselect" all the channels, then select MRT24 and MLT24 (e.g. from the channel names on the right), and press 'ok'. (It will help if you save these selections as a file, which can be used later to display only a subset of channels). You will now only see these two channels in the SPM Graphics Window, which clearly show a difference between faces (trial-type 1, in blue) and scrambled faces (trial-type 2, in red) around approximately 170ms (the "M170"; Figure 34.11). The sign of this difference is reversed across left and right hemispheres, as is common for the axial components of the magnetic fields from tangential current sources.

\* Select "Contrast" from the "Other..." pulldown menu on the SPM window (or type `spm_eeg_weight_epoch` in the Matlab window). This function creates linear contrasts of ERPs/ERFs. Select the `fmae_meg.mat` file, and enter [1 -1; 1/2 1/2] as the contrasts. This will create new file `mfmae_meg.mat`, in which the first trial-type is now the differential ERF between faces and scrambled faces, and the second trial-type is the average ERF.

If you want to see the 2D topography of the differential ERF between faces and scrambled faces, you can Display the new file `mfmae_eeg.mat`, select trial-type 1, press "Topography" and in the new window, select "2D" and 165ms as the timepoint (Figure 34.12). This will show a bilinear interpolation of the difference across the 151 channels.

You can move the slider left and right to see the development of the M170 over time.

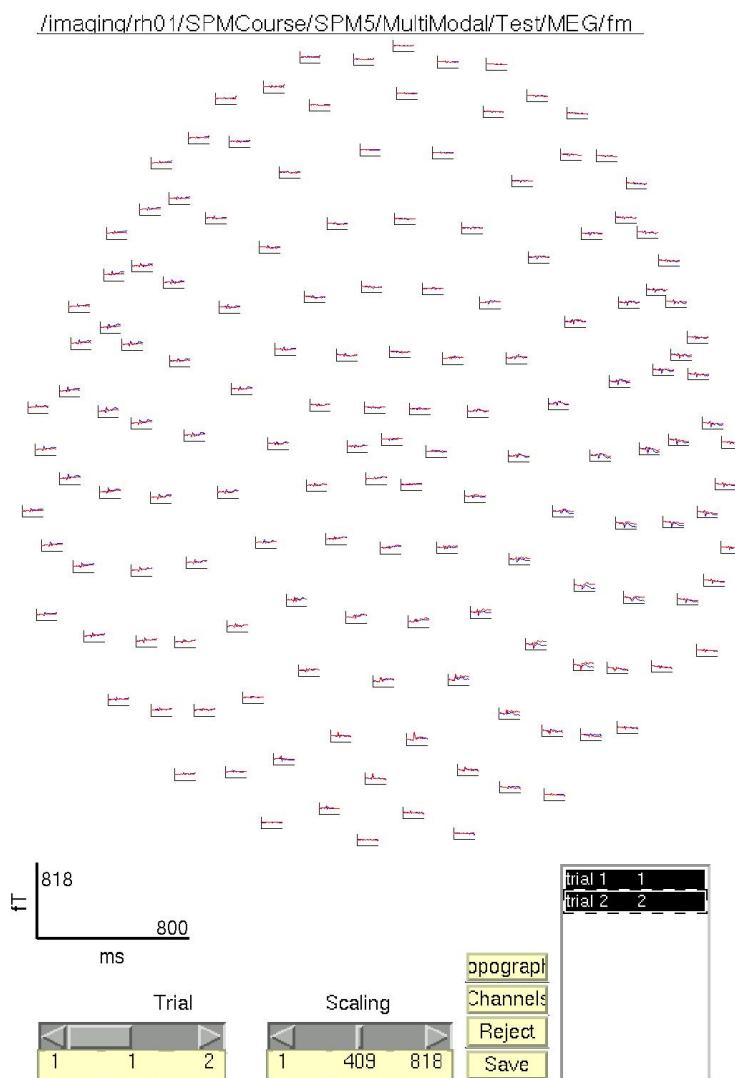


Figure 34.10: *SPM Display window for mean, smoothed ERF (fmae-meg.mat) for all 151 MEG channels.*

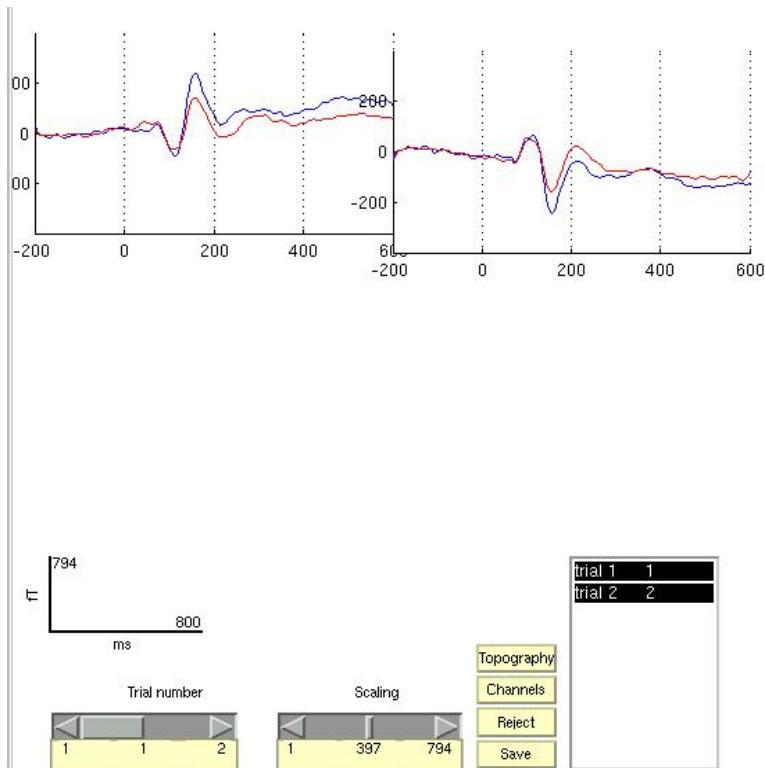


Figure 34.11: Two selected MEG channels (MLT24 and MRT24).

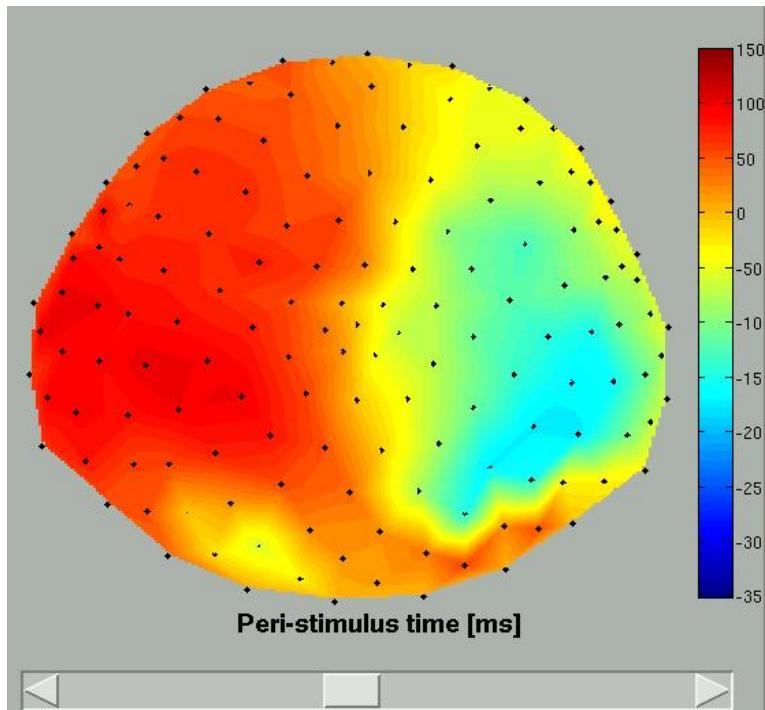


Figure 34.12: 2D topography of the ERF of faces minus scrambled faces at 165ms

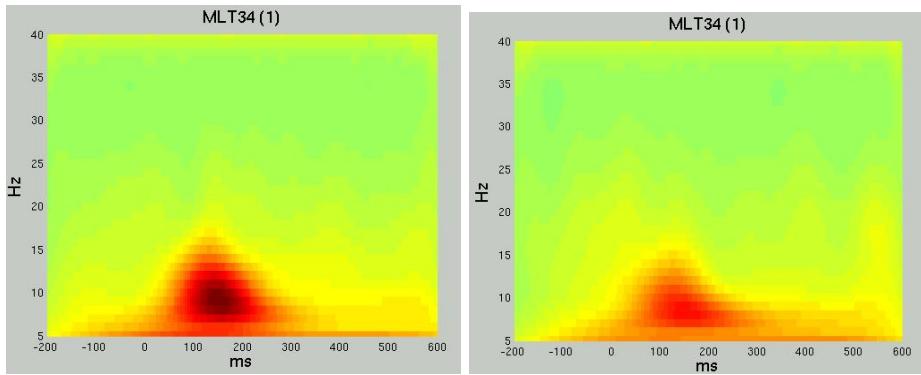


Figure 34.13: Total power spectra for faces (left) and scrambled faces (right) for channel MLT34

### 34.5.2 Time-Frequency Analysis

SPM uses Morlet wavelets to perform time-frequency analyses.

\* Select the 'time-frequency' option under the 'Other' pull-down menu, and select the `ae_meg.mat` file. SPM will then prompt you for the frequencies you wish to analyse, for which you can type [5:40] (Hz). To the question "remove baseline", press "no" (because for frequencies as low as 5Hz, one would need a longer pre-stimulus baseline, to avoid edge-effects<sup>11</sup>). Later, we will compare two trial-types directly, and hence any pre-stimulus differences will become apparent. Change the default Morlet wavelet order (N) from 7 to 5. This factor effectively trades off frequency vs time resolution, with a lower order giving higher temporal resolution. You will then be prompted to select channels, for which you can highlight and delete the default option of all channels, and type just 66 (which corresponds to channel 'MLT34', as can be confirmed by typing `D.channels.names` in the Matlab window)<sup>12</sup>.

This will produce two new files, `t1_e_eeg.mat` and `t2_e_eeg.mat`. The former contains the power at each frequency, time and channel; the latter contains the corresponding phase angles.

\* Press the 'Averaging' button and select the `t1_e_meg.mat` file. After a few moments, the matlab window will echo:

```
e_meg.mat: Number of replications per contrast:  
average 1: 86 trials, average 2: 86 trials
```

and a new file will be created in the MEG directory called `mt1_e_meg.mat`.

This contains the power spectrum averaged over all trials, and will include both "evoked" and "induced" power. Induced power is (high-frequency) power that is not phase-locked to the stimulus onset, which is therefore removed when averaging the amplitude of responses across trials (i.e., would be absent from a time-frequency analysis of the `mae_eeg.mat` file).

The power spectra for each trial-type can be displayed using the usual Display button and selecting the `mt1_e_eeg.mat` file. This will produce a plot of power as a function of frequency (y-axis) and time (x-axis) for Channel MLT34. If you use the "trial" slider to switch between trial(types) 1 and 2, you will see the greater power around 150ms and 10Hz for faces than scrambled faces (click on one channel to get scales for the axes, as in Figure 34.13). This corresponds to the M170 again.

We can also look at evidence of phase-locking of ongoing oscillatory activity by averaging the phase angle information. This time, we do not take the straight (arithmetic) mean, since the data are phase angles, and this average is not particularly meaningful. Instead we calculate their vector mean (when converting the angles to vectors in Argand space), which corresponds to a "Phase-Locking Value" (PLV) which lies between 0 (no phase-locking across trials) to 1 (perfect phase-locking).

<sup>11</sup>For example, for 5Hz, one would need at least  $N/2 \times 1000\text{ms}/5$ , where N is the order of the Morlet wavelets (i.e., number of cycles per Gaussian window), e.g., 600ms for a 6th-order wavelet.

<sup>12</sup>You can of course obtain time-frequency plots for every channel, but it will take much longer (and result in a large file).

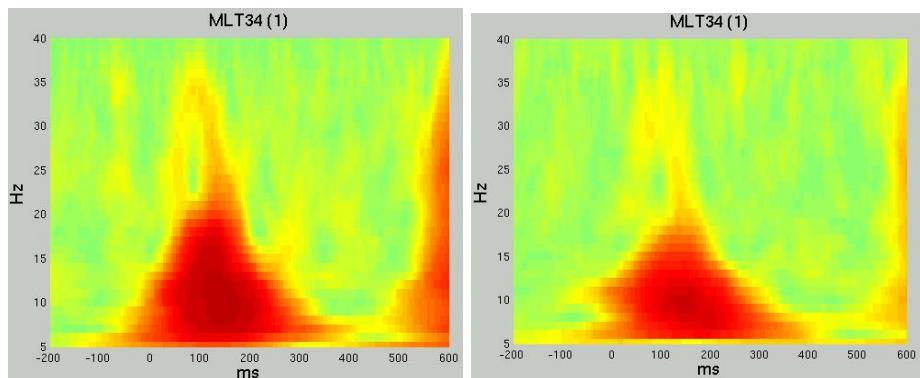


Figure 34.14: *Phase-Locking Values for faces (left) and scrambled faces (right)* for channel MLT34

\* Press the 'Averaging' button and select the `t2_e_meg.mat` file. This time you will be prompted for either a straight or a vector average, for which you should select "vector". The matlab window will echo:

```
e_meg.mat: Number of replications per contrast:  
average 1: 86 trials, average 2: 86 trials
```

and a new file will be created in the MEG directory called `mt2_e_meg.mat`.

If you now display the file `mt2_e_eeg.mat` file, you will see PLV as a function of frequency (y-axis) and time (x-axis) for Channel MLT34. Again, if you use the "trial" slider to switch between trial(types) 1 and 2, you will see greater phase-locking around 10Hz and 100ms for faces than scrambled faces, as in Figure 34.14. Together with the above power analysis, these data suggest that the M170 includes an increase both in power and in phase-locking of ongoing oscillatory activity in the alpha range (Henson et al, 2005b).

### 34.5.3 2D Time-Frequency SPMs

Analogous to Section 34.4.3, we can also use Random Field Theory to correct for multiple statistical comparisons across the 2-dimensional time-frequency space.

\* Type `spm_eeg_convertmat2ana3Dtf` in the Matlab window, and select the `t1_e_eeg.mat` file.

This will create time-frequency images for each trial of the two types, with dimensions 161x36x1, as for the example shown in Figure 34.15 from pressing "Display: images" on the main SPM window.

As in Section 34.4.3, we then take these images into an unpaired t-test across trials to compare faces versus scrambled faces. We can then use classical SPM to identify times and frequencies in which a reliable difference occurs, correcting across the multiple comparisons entailed (Kilner et al, 2005).

\* First create a new directory, eg. `mkdir TFstatsPow`.

\* Then press the "specify 2nd level" button, select "two-sample t-test" (unpaired t-test), and define the images for "group 1" as all those in the subdirectory "triatype1" (using right mouse, and "select all") and the images for "group 2" as all those in the subdirectory "triatype2". Finally, specify the new TFstatsPow directory as the output directory, and press "run". (Note that this will be faster if you saved and could load an SPM job file from Section 34.4.3).

This will produce the design matrix for a two-sample t-test.

\* The press "Estimate", and when it has finished, press "Results" and define a new T-contrast as [1 -1]. Keep the default contrast options, but threshold at  $p < .05$  FWE corrected for the whole "image". Then press "whole brain", and the Graphics window should now look like that in Figure 34.16 (ignore the glass brain MIP).

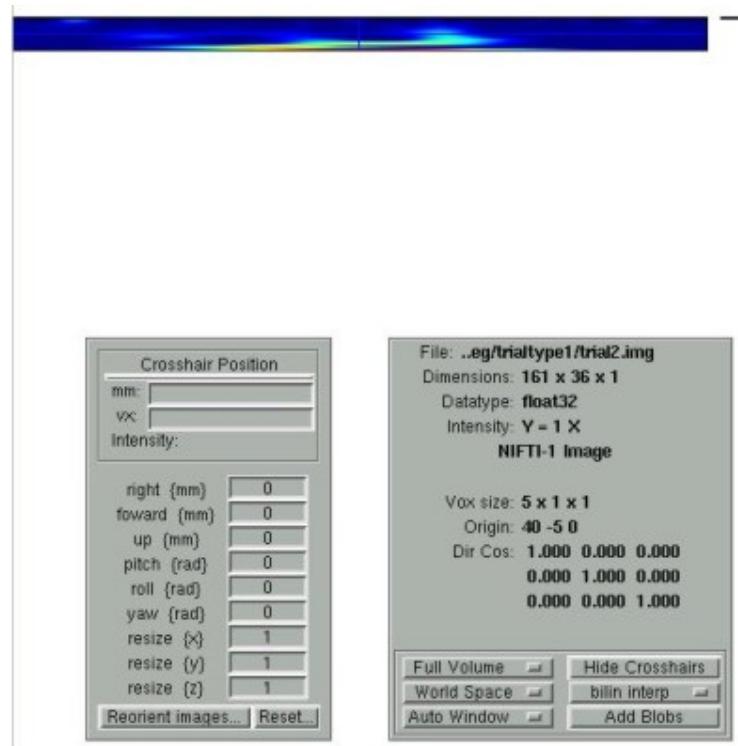


Figure 34.15: 3D image for trial 2 of *t1-e-eeg.mat*. The left section is through time (*x*) and frequency (*y*) (the right image is an *y-z* section, though there is only one value in *z*, i.e., it is really a 2D image).

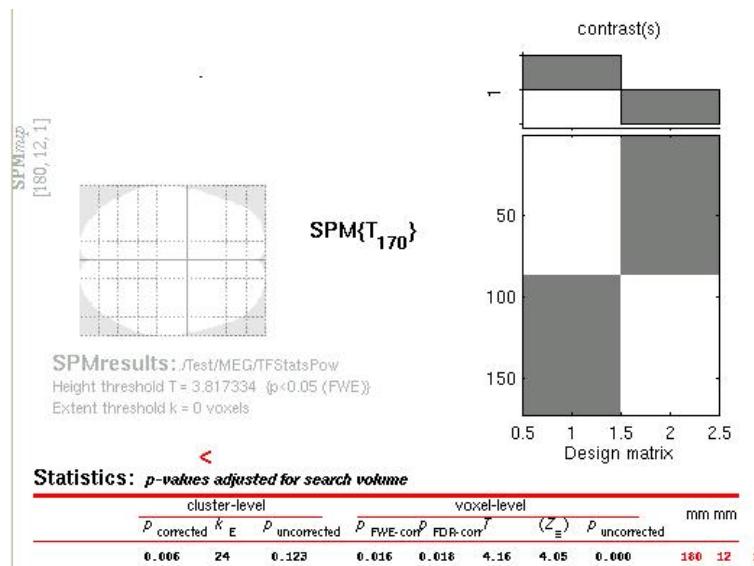


Figure 34.16: 2D time-frequency SPMt at  $p < .001$  uncorrected for the power difference between face and scrambled faces at Channel MLT34. Note that the brain outline in the MIP should be ignored. The *x* coordinates refer to time in ms; the *y* coordinates refer to frequency in Hz (the *z*-coordinate is always 1).

This will list one "region" within the 2D time-frequency space in which faces produce greater power than scrambled faces, having corrected for multiple T-tests across pixels. This has a maximum of [180 12 1], ie 12 Hz and 180ms post-stimulus.

If you repeat the above time-frequency analysis on the `e_meg.mat` file, but this time keep every channel and answer 'yes' to the "average over channels?" question, and then repeat the above statistical analysis of power, you will notice that there is also a reliable decrease in induced high-frequency power (around 400ms and 35 Hz) for faces vs scrambled faces, which could also be source-localised.

### 34.5.4 "Imaging" reconstruction of differential power

In Section 34.4.4 we localised the differential evoked potential difference in EEG data corresponding to the N170. Here we will localise the total power of faces vs scrambled faces in a timewindow corresponding to that of the M170, ie including potential induced components (see Friston et al, 2000).

\* Press the '3D source reconstruction' button, and press the "load" button at the top of the new window. Select the `e_meg.mat` file and type a label (eg "M170") for this analysis.

\* Press the 'MRI' button, select 3000 for the number of vertices in the mesh, and select the `smri.img` file within the sMRI sub-directory...

This will take some time while the MRI is segmented and binary images of the skull created (see Section 34.4.4 for more details on these files)<sup>13</sup>.

The choice of the minimum of 3000 vertices in the cortical mesh is simply to reduce computation time (the actual number of vertices resulting will be 3004).

Note that the cortical mesh (and the distances within the mesh) are not created directly from the segmented MRI (like the skull and scalp meshes), but rather are determined from a template cortical mesh in MNI space via inverse spatial normalisation (Mattout et al, in press).

When meshing has finished, the field `D.inv{1}.mesh` will be updated in matlab. Press "save" in top right of window to update the corresponding mat file on disk. The cortex (blue), inner skull (red) and scalp (orange) meshes will also be shown in the Graphics window.

\* Press the 'Co-register' button, respond "no" to the 'Read Polhemus?' question, and then select the following files in response to each prompt (pressing "yes" to the 'Use headshape file' prompt):

```
MEG/Polhemus/meg_sens_loc.mat
MEG/Polhemus/meg_fids.mat
MEG/Polhemus/meg_hsf.mat
MEG/Polhemus/meg_sens_or.mat
sMRI/smri_fids.mat
```

(like in Section 34.4.3, except now we also need to provide information about the orientation of each MEG sensor, as in the penultimate file here).

This stage coregisters the MEG sensor positions and orientations (in "MEG" space) with the structural MRI and solution mesh (in "MRI" space). This is done via an approximate matching of the fiducials in the two spaces, followed by a more accurate surface-matching routine that fits the head-shape function (in "MEG" space) to the scalp that was created in the previous meshing stage via segmentation of the MRI. The match will be shown in a window like that in Figure 34.17. (Note that the match of the MEG and MRI fiducials is poor because the MEG fiducials did not correspond exactly to the nasion and peri-auricular points (see footnote 3); this does not matter because the coregistration is dominated by the close fit of the digitized headshape to the scalp mesh).

When coregistration has finished, the field `D.inv{1}.datareg` will be updated in matlab. Press "save" in top right of window to update the corresponding mat file on disk. Finally, a figure like that in Figure 34.17 will also be produced, which you can rotate with the mouse (using the Rotate3D Matlab Menu option) to check all sensors.

\* Press the 'Forward Model' button. This assumes the sensors lie on a single best-fitting sphere, which allows analytical computation of the forward model (lead field) that maps each

---

<sup>13</sup>Note that this procedure can be shortened in the batch script included here, by loading the normalisation parameters and binary masks from previous segmentations of the structural MRI.

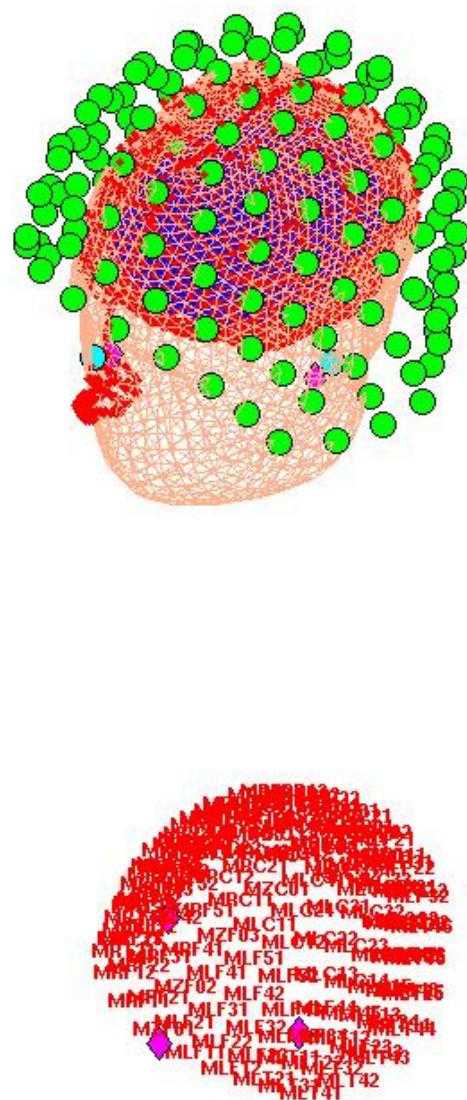


Figure 34.17: Graphical output of registration of MEG and sMRI data, showing (upper panel) cortex (blue) and scalp (black) meshes, sensor locations (green), MRI and Polhemus fiducials (cyan/magenta), and headshape (red dots).

”dipole” in the cortical mesh to each sensor, assuming that the orientation of the dipole at each vertex is normal to the surface of the mesh at that point. This stage uses BrainStorm functions <sup>14</sup>. The Matlab window will output:

```
Scalp best fitting sphere computed (in 11 iterations)
Centre = [0.0001 -0.0218 0.0027] (Radius = 0.0774)
```

\* Press ’Invert.’, select ’Classical’, select ’yes’ to ’All conditions or trials?’, select ’MSP’ (for Multiple Sparse Priors) for the type of inversion, ”Standard” for the model (i.e, to use defaults; you can customise a number of options if you press Custom instead) (see Friston et al, in press-a, for more details about these parameters).

Press ”save” to save the results. You can now explore the results via the 3D reconstruction window. If you type 165 into the box in the bottom right (corresponding to the time in ms) and press ”mip”, you should see an output like in Figure 34.18. This fit explains approx 87% of the data.

Note the hot-spots in the fusiform. The timecourses come from the peak voxel. The red line shows the condition currently being shown (corresponding to the ”Condition 1” toggle bar in the reconstruction window); the grey line(s) will show all other conditions. Condition 1 is faces; if you press the ”condition 1” toggle, it will change to Condition 2 (scrambled faces), then press ”mip” again and the display will update (note the colours of the lines have now reversed from before, with red now corresponding to scrambled faces).

If you toggle back to condition 1 and press ”movie”, you will see the changes in the source strengths over peristimulus time (from the limits 0 to 300ms currently chosen by default).

If you press ”render” you can get a very neat graphical interface to explore the data (the buttons are fairly self-explanatory). However, we will concentrate on how one might perform statistics.

\* Press the ”Window” button in the reconstruction window, enter ”150 200” as the timewindow of interest and ”5 40” as the frequency band of interest. The Graphics window will show the mean activity for this time/frequency contrast (for faces alone, assuming the condition toggle is showing ”condition 1”).

\* If you then press ”Image”, press ”12” for the smoothing kernel, and SPM will write 3D Nifti images corresponding to the above contrast for each condition:

```
w_e_meg_1_1.nii
w_e_meg_1_2.nii
sw_e_meg_1_1.nii
sw_e_meg_1_2.nii
```

Note that the first two images are unsmoothed (but normalised); the latter two are smoothed by a 12mm isotropic Gaussian kernel. The last number in the file name refers to the condition number; the penultimate number refers to the reconstruction number (ie the number in red in the reconstruction window, i.e, D.val, here 1).

The smoothed results for Condition 1 will also be displayed in the Graphics window, together with the normalised structural, as in Figure 34.19. Note that the solution image is in MNI (normalised) space, because the use of a canonical mesh provides us with a mapping between the cortex mesh in native space and the corresponding MNI space.

You can also of course view the image with the normal SPM ”Display:image” option, and locate the coordinates of the ”hotspots” in MNI space. Note that these images contain RMS (unsigned) source estimates (see Henson et al, 2007).

If you want to see where activity (in this time/freq contrast) is greater for faces and scrambled faces, you can use SPM’s ImCalc to create a difference image of **sw\_e\_meg\_1\_1.nii** - **sw\_e\_meg\_1\_2.nii** - you should see bilateral fusiform. For further discussion of localising a differential effect (as in Section 34.4.4 with ERPs), vs taking the difference of two localisations, as here, see Henson et al (2007).

You could also explore the other inversion options, like COH and MNM, which you will note give more superficial solutions (a known problem with standard minimum norm; see also Friston et al, in press-a). To do this quickly (without repeating the MRI segmentation, coregistration

---

<sup>14</sup>Brainstorm is available from <http://neuroimage.usc.edu/ResearchMEGEEGBrainStorm.html>

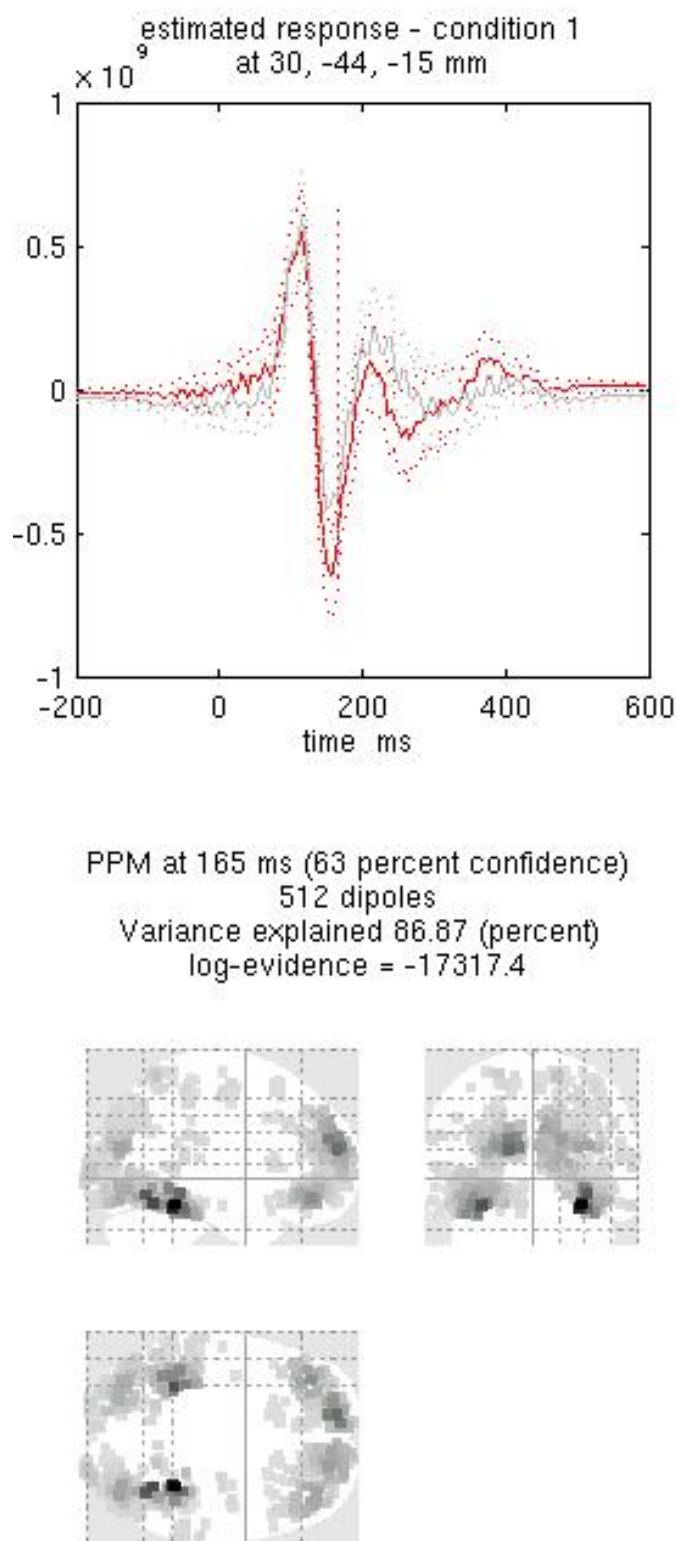


Figure 34.18: Graphic output for MSP-estimated activity at 165ms for faces.

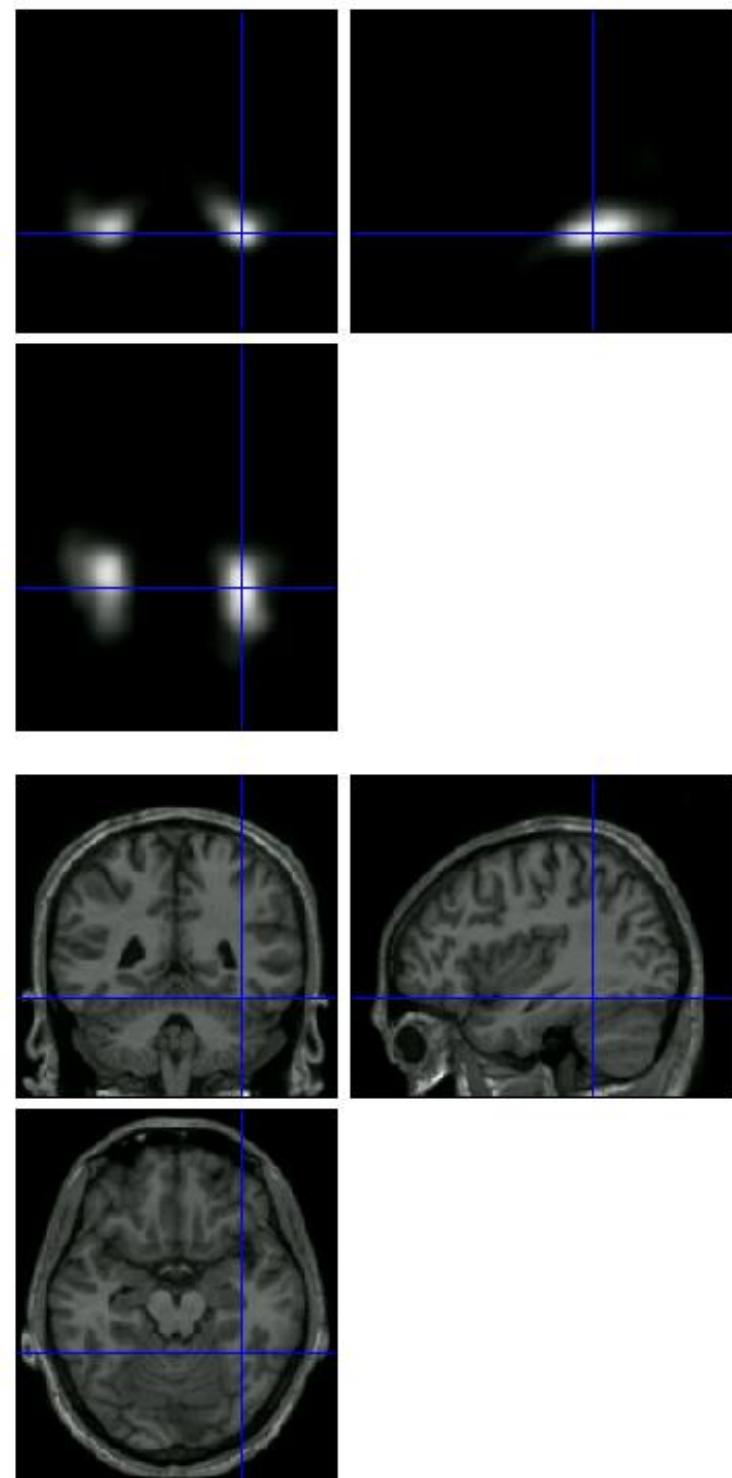


Figure 34.19: Display of the smoothed 3D image of the MSP-estimated activity between 150-200ms in the frequency band 5-40Hz for faces, together with the normalised structural. Note large hotspots in bilateral fusiform.

and forward modelling), press the "new" button in the reconstruction window, which by default will copy these parts from the previous reconstruction.

## 34.6 fMRI analysis

Only the main characteristics of the fMRI analysis are described below; for a more detailed demonstration of fMRI analysis, see Chapter 29.

Note that all the job files for each stage of preprocessing and analysis are also provided:

```
fMRI/realign_job.mat
fMRI/slicetime_job.mat
fMRI/smooth_job.mat
fMRI/stats_job.mat
```

These can be loaded and run, though of course the location of the files and the output directory will need to be changed.

### 34.6.1 Preprocessing the fMRI data

\* Toggle the modality from EEG to fMRI, and change directory to the fMRI subdirectory (either in Matlab, or via the "CD" option in the SPM "Utils" menu)

\* Select 'Realign' from the 'Realign and Unwarp' menu, click on 'Data', and select 'New Session'. Double-click on the new Session branch, and click on the 'Images' button, click on the 'specify files' and select all 215 fM\*.img files in the Scan directory (using the right mouse to 'select all', assuming the filter is set to ^f.\*img).

Realignment will produce a spm\*.ps postscript file in the current directory, which shows the estimated movement (like in Figure 34.20). Importantly, the resliced images will be output as rfM\*.img files. A mean image will also be written:

```
meanfMS02554-0003-000006.img
```

as will the movement parameters in the text file:

```
rp_fMS02554-0003-000006.txt
```

\* Press the 'slice-timing' button, select the functional images (filter set to ^rf.\* to avoid the mean image), enter 2.88 as the TR, 2.88\*31/32 as the TA, the slice-order as [1:32] (since the first slice in the file is the top slice, and this was the first slice acquired in the descending sequence), and the reference slice to 16. This will write out 215 images arfM\*.img, in which the data have been interpolated to match the acquisition time of the middle slice (in space and time, given the sequential acquisition).

\* Press the 'smooth' button and keep the default 10x10x10mm smoothing. This will produce 215 spatially-smoothed images sarfM\*.img.

Note that we will not normalise these images, in order to compare them with the MEG and EEG source reconstructions, which are in the native MRI space.

### 34.6.2 Statistical analysis of fMRI data

\* Load the onsets.mat file provided into the Matlab workspace

\* Press the 'specify 1st-level' button, change the microtime onset from 1 to 8, select the 215 'sarfM\*img' images, define two new conditions - condition 1 called "faces" with onsets set to onsets1 and condition 2 called "scrambled faces" with onsets set to onsets2 (all duration 0) - select the rp\_fMS02554-0003-000006.txt file as 'Multiple Regressors' (to covary out some of the residual movement-related effects), and select the fMRI/Stats as the output directory (loading and editing the stats\_job.mat file provided will save a lot of time here!). Keep the rest of the parameters (e.g. use of a canonical HRF basis function) at their default settings.

This will produce a design matrix like that in Figure 34.21, which is stored in the file:

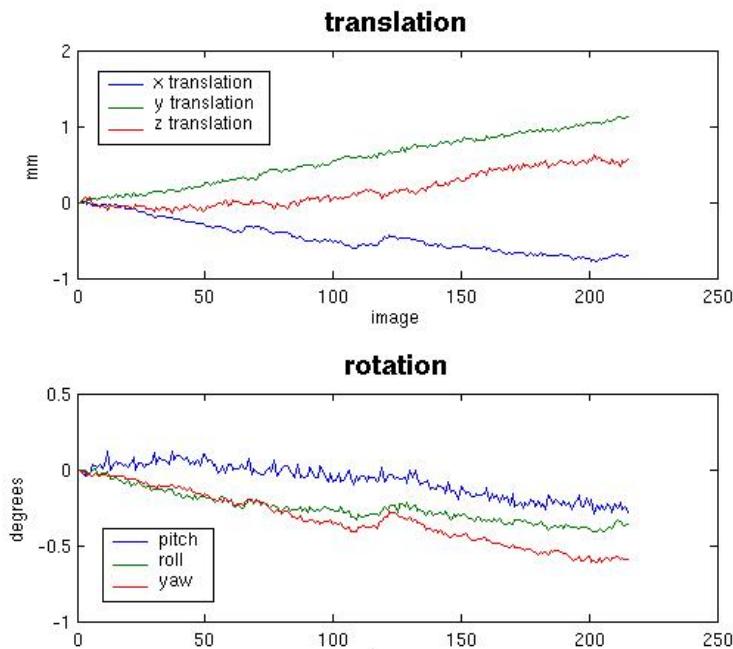


Figure 34.20: Movement parameters from Realignment of the fMRI data.

#### fMRI/Stats/SPM.mat

\* Then estimate the parameters of the design matrix by pressing the 'Estimate' button and selecting the SPM.mat file

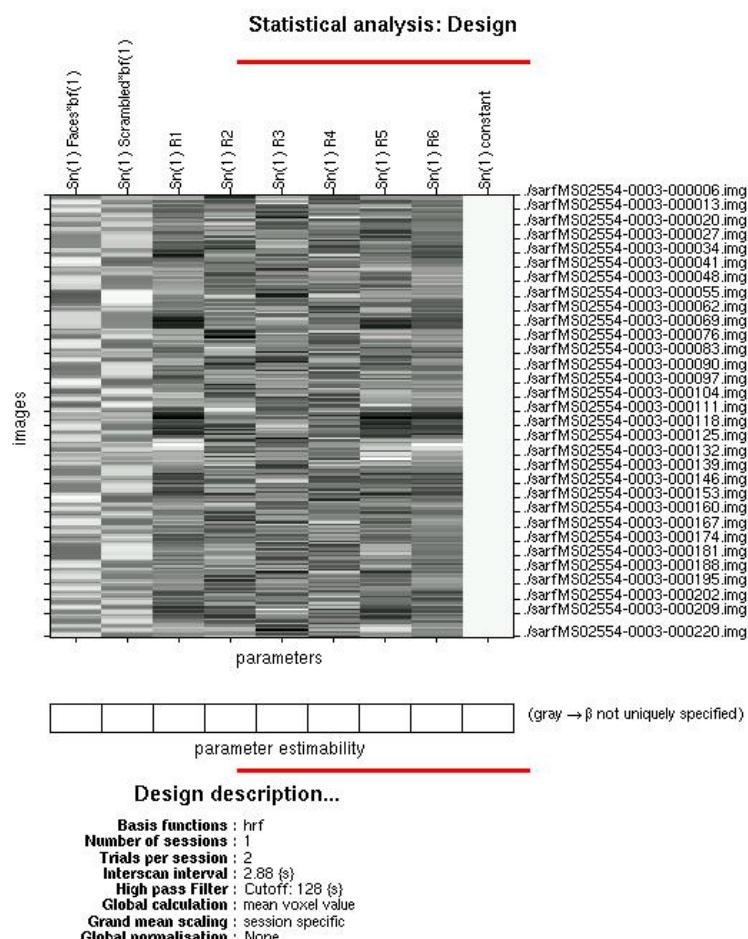
\* Finally, to see the regions that respond differentially between faces and scrambled faces, press 'Results' and define a new F-contrast (called, e.g, 'Faces - Scrambled') by typing the contrast weights [1 -1].

This will identify regions in which the parameter estimate for the canonical HRF differs reliably between faces and scrambled faces. This could include regions that show both a "greater" relative response for faces, and regions that show a "greater" relative response for scrambled faces (such a two-tailed test is used because we do not know the precise relationship between haemodynamic changes measured by fMRI and the synchronous current changes measured by EEG/MEG).

If the resulting SPMF is thresholded at  $p < .05$  FWE corrected, the resulting MIP and table of values should be like that in Figure 34.22. Only two regions survive correction: right fusiform and orbitofrontal cortex (note coordinates refer to native MRI space; not MNI space). These can be displayed on the (attenuation-corrected) structural msmri.nii. They are a subset of the regions identified by the same contrast in a group of 18 subjects in Henson et al (2003). At a lower threshold (e.g,  $p < .01$  uncorrected), one can see additional activation in left fusiform, as well as other regions.

There is some agreement between these fMRI effects and the localised EEG/MEG effects around the 170ms latency - eg in orbitofrontal and right fusiform - though of course the EEG dipoles were bilateral, and there were more extensive posterior occipitotemporal effects in the source-localised MEG data. Note of course that the fMRI data may include differences between faces and scrambled faces that occur at latencies other than the M170 (e.g, later), or differences in "induced" high-frequency M/EEG power that is not phase-locked to stimulus onset (Henson et al, 2005b).

One could use the unthresholded F-image as an additional continuous prior within the PEB L2-norm method offered by SPM5, or probably better, one could take a number of regions after thresholding the SPMF, and enter each as a separate prior on the PEB L2-norm method (this way, different regions can be up-weighted or down-weighted as a function of whether they are likely to be active during the critical timewindow being localised).

Figure 34.21: *Design matrix for the fMRI analysis.*

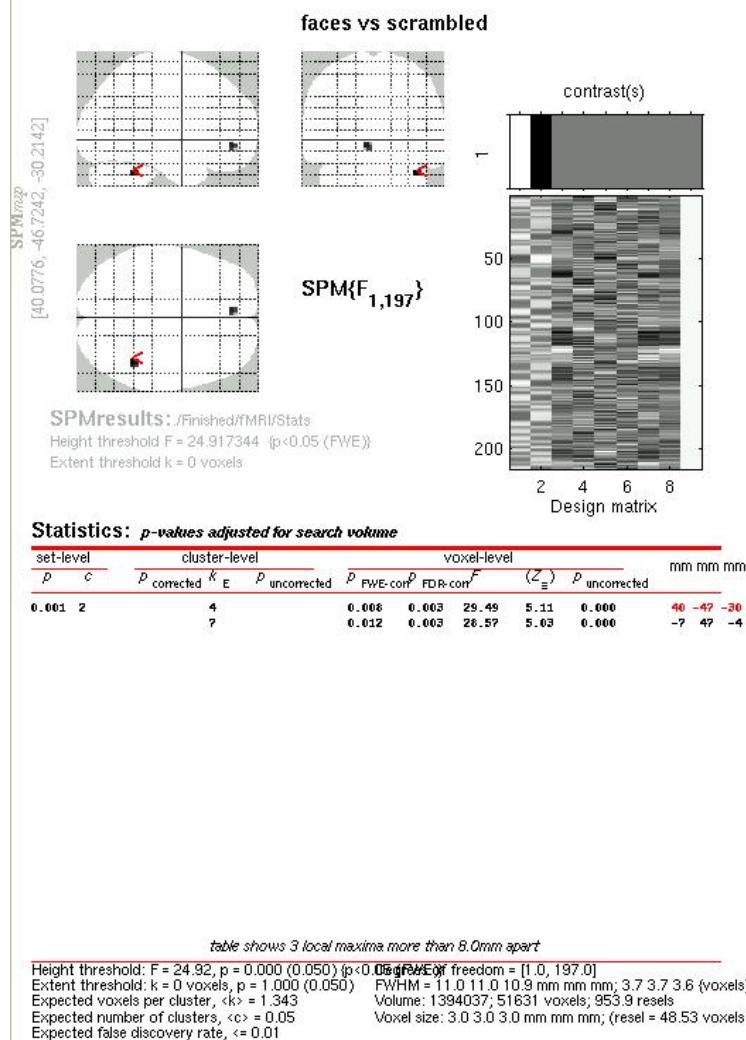


Figure 34.22: SPMF for faces vs scrambled faces. Note that the coordinates are in the MRI native space (no normalisation performed) so bear a close, but not exact, relationship with MNI coordinates (affecting brain outline in MIP too).

### 34.7 References

1. Friston, K, Daunizeau, J, Kiebel, S, Phillips, C, Trujillo-Barreto, N, Henson, R, Flandin, G, Mattout, J (in press-a). Multiple sparse priors for the M/EEG inverse problem. *Neuroimage*.
2. Friston, K, Carlton Chu, Janaina Mouro-Miranda, Oliver Hulme, Geraint Rees, Will Penny and John Ashburner (in press-b). Bayesian decoding of brain images. *NeuroImage*.
3. Friston K, Henson R, Phillips C, and Mattout J. (2006). Bayesian estimation of evoked and induced responses. *Human Brain Mapping*, 27, 722-735.
4. Henson, R, Goshen-Gottstein, Y, Ganel, T, Otten, L, Quayle, A. and Rugg, M. (2003). Electrophysiological and hemodynamic correlates of face perception, recognition and priming. *Cerebral Cortex*, 13, 793-805.
5. Henson R, Mattout J, Friston K, Hassel S, Hillebrand A, Barnes G and Singh K. (2005a) Distributed source localisation of the M170 using multiple constraints. *HBM05 Abstract*.
6. Henson R, Kiebel S, Kilner J, Friston K, Hillebrand A, Barnes G and Singh K. (2005b) Time-frequency SPMs for MEG data on face perception: Power changes and phase-locking. *HBM05 Abstract*.
7. Henson, R.N. Mattout, J, Singh, K, Barnes, G, Hillebrand, A. and Friston, K.J. (2007). Population-level inferences for distributed MEG source localisation under multiple constraints: Application to face-evoked fields. *Neuroimage*, 38, 422-438.
8. Josephs, O., Deichmann, R., Turner, R. (2000). Trajectory measurement and generalised reconstruction in rectilinear EPI. *NeuroImage* 11, S543.
9. Kilner, J., Kiebel, S and Friston, K. J. (2005). Applications of random field theory to electrophysiology. *Neuroscience Letters*, 374:174-178.
10. Kilner, J. and Penny. W. (2006). Robust Averaging for EEG/MEG. *HBM06 Abstract*.
11. Kiebel S and Friston K (2004). Statistical Parametric Mapping for Event-Related Potentials II: A Hierarchical Temporal Model. *NeuroImage*, 22, 503-520.
12. Kiebel, S.J., David, O. and Friston, K. J. (2006). Dynamic Causal Modelling of Evoked Responses in EEG/MEG with lead-field parameterization. *NeuroImage*, 30:1273-1284.
13. Mattout J, Pelegri-Issac M, Garnero L and Benali H. (2005a). Multivariate source prelocalization (MSP): use of functionally informed basis functions for better conditioning the MEG inverse problem. *Neuroimage*, 26, 356-73.
14. Mattout, J, Phillips, C, Penny, W, Rugg, M and Friston, KJ (2005b). MEG source localisation under multiple constraints: an extended Bayesian framework. *NeuroImage*.
15. Mattout, J., Henson, R N. and Friston, K.J. (in press). Canonical Source Reconstruction for MEG. *Computational Intelligence and Neuroscience*.
16. Phillips, C. M.D. Rugg, M and Friston, K (2002). Systematic Regularization of Linear Inverse Solutions of the EEG Source Localisation Problem. *NeuroImage*, 17, 287-301.
17. Spinelli, L, Gonzalez S, Lantz, G, Seeck, M and Michel, C. (2000). Electromagnetic inverse solutions in anatomically constrained spherical head models. *Brain Topography*, 13, 2.



# Chapter 35

## 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>.

### 35.1 Using DARTEL for VBM

The following procedures could be specified one at a time, but it is easier to use the *Batch* option of the *UTILS* pulldown to set up a job that will do all the processing. The sequence of jobs (use the *TASKS* pulldown from the *Graphics* window to select *Batch*) would be:

- SPM Jobs
  - Spatial
    - \* Segment: To obtain \*\_seg\_sn.mat files for “importing” the data into a form that DARTEL can use for registering the subject’s scans.
  - Tools
    - \* DARTEL Tools
      - Initial Import: Uses the \*\_seg\_sn.mat files to generate roughly (via a rigid-body) aligned grey and white matter images of the subjects.
      - Run DARTEL (create Template): Determine the nonlinear deformations for warping all the grey and white matter images so that they match each other.
      - Create Warped: Actually generate the “modulated” warped grey and white matter images.
  - Spatial
    - \* Smooth: Smooth the “modulated” warped grey and white matter, prior to performing the statistics.

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. The *TASKS*–>*Spatial*–>*Segment* pulldown can be used here:

- Segment

---

<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.

- **Data:** Select all the T1-weighted images, one per subject. It is usually a good idea to have roughly aligned them to MNI space first. The *Display* button can be used to reorient the data so that the *mm* coordinate of the AC is within about 3cm from [0, 0, 0], and the orientation is within about 15° of MNI space. The *Check Reg* button can be used to see how well aligned a number of images are.
- **Output Files:** No output files are needed. The Segmentation produces a *\*\_seg\_sn.mat* and a *\*\_seg\_inv\_sn.mat* for each image. It is the *\*\_seg\_sn.mat* files that are needed for the next step.
- **Custom:** Default settings can usually be used here.

The resulting *\*\_seg\_sn.mat* files encode various parameters that allow the data to be “imported” into a form that can be used by the main DARTEL algorithm. In particular, *procrustes* aligned maps of grey and white matter can be generated. Select *TASKS* –>*Tools* –>*Dartel Tools* –>*Initial Import*:

#### • Initial Import

- **Parameter Files:** Select all the *\*\_seg\_sn.mat* files generated by the previous step. The T1-weighted scans need not be selected, as the import routine will try to find them. If the image files have not been moved since the segmentation, then their location can be determined by the contents of the *\*\_seg\_sn.mat* files. If they have been moved, then the routine looks for the files in the current directory, or the output directory.
- **Output Directory:** Specify where the imported data should be written.
- **Bounding box:** This is the bounding box for the imported data. If the values are not finite (eg, if they are  $[NaN, NaN, NaN; NaN, NaN, NaN]$ ) then the bounding box for the tissue probability maps, used as priors for the segmentation, will be assumed. Note that the deformations that DARTEL estimates will wrap around at the boundaries, so it is usually a good idea to ensure that the whole brain is easily enclosed within the bounding box.
- **Voxel size:** These specify the resolution of the imported data. [1.5, 1.5, 1.5] are reasonable values here. If the resolution is finer than this, then you may encounter memory problems during the actual DARTEL registration. If you do want to try working at a higher resolution, then consider changing the bounding box (but allow for the strange behaviour at the edges).
- **Image option:** No imported image is needed - usually only the grey and white matter.
- **Grey Matter:** Yes, you need this.
- **White Matter:** Yes, you also need this.
- **CSF:** The CSF is not usually segmented very reliably because the segmentation only has tissue probability maps for GM WM and CSF. Because there are no maps for bone and other non-brain tissue, it is difficult for the segmentation algorithm to achieve a good CSF segmentation. Because of the poor CSF segmentation, it is not a good idea to use this tissue class for the subsequent DARTEL registration.

The output of the importing step are a series of rigidly aligned tissue class images (grey matter is encoded by *rc1\*.nii* and white matter by *rc2\*.nii* – see Fig 35.1). 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 *TASKS* –>*Tools* –>*Dartel Tools* –>*Run DARTEL (create Template)*.

#### • Run DARTEL (create Template)

##### – Images

- \* **Images:** Select all the *rc1\*.nii* files generated by the import step.

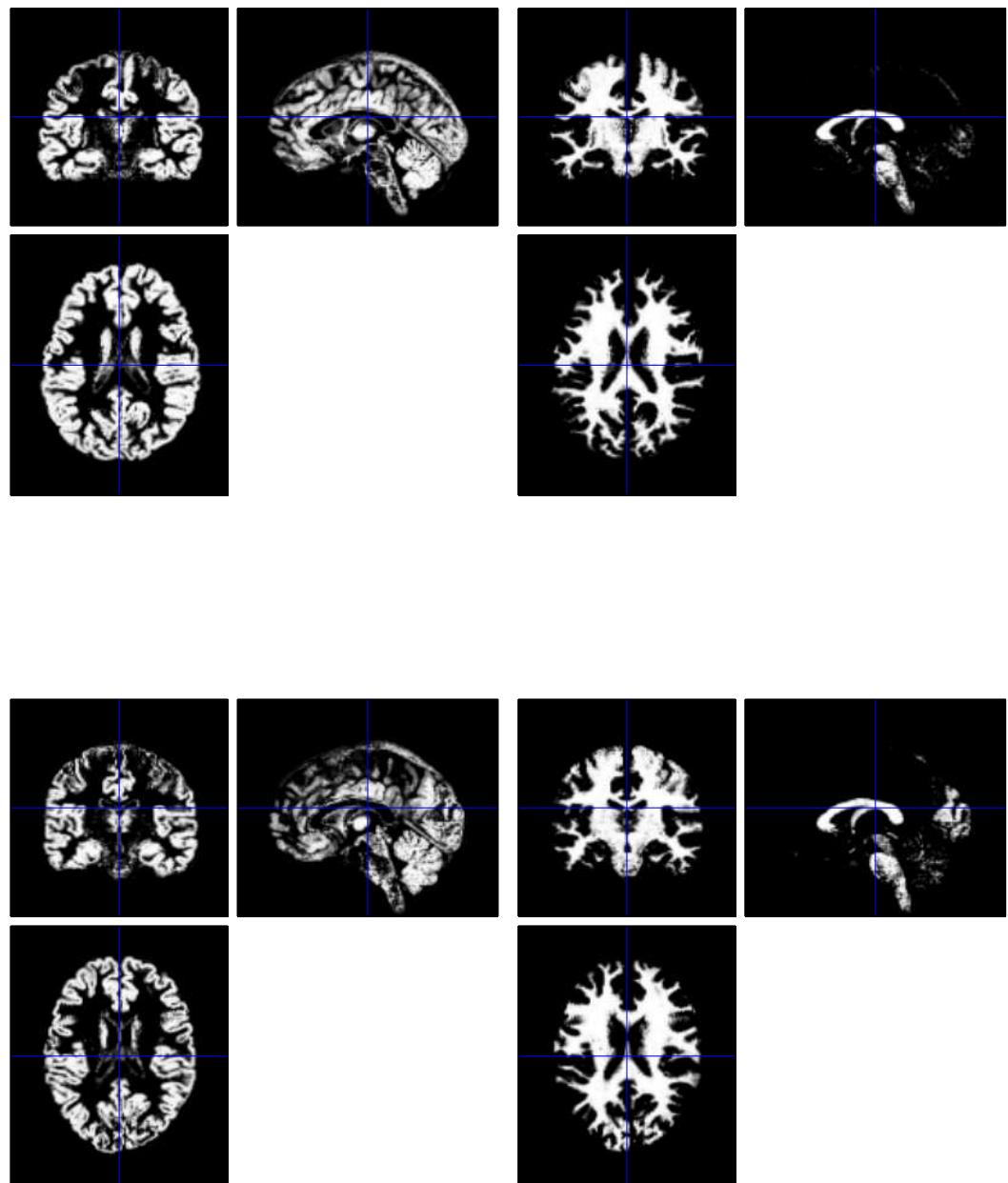


Figure 35.1: Imported data for two subjects (A and B). Top row: rc1A.nii and rc2A.nii. Bottom row: rc1B.nii and rc2B.nii.

- \* **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 35.2), and a series of u\_rc1\*.nii files. The first template is based on the average<sup>5</sup> of the original imported data, whereas 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”).

The next step is to create the Jacobian scaled (“modulated”) warped tissue class images, by selecting *TASKS* –> *Tools* –> *DARTEL Tools* –> *Create Warped*.

#### • Create Warped

- **Flow Fields:** Specify the flow fields (u\_rc1\*.nii) generated by the nonlinear registration.
- **Images**
  - \* **Images:** Select the rc1\*.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 rc2\*.nii files.
- **Modulation:** Specify “Modulation” in order to have Jacobian transformed tissue probability maps.
- **Time Steps:** Specify 64, which is the default number of time steps used to generate the deformations from the flow fields.
- **Interpolation:** Usually, you would specify “Trilinear”.

The end result should be a bunch of mwrc1\*.nii files (possibly with mwrc2\*.nii if white matter is also to be studied). These are then smoothed (*TASKS* –> *Spatial* –> *Smooth*).

#### • Smooth

- **Images to Smooth:** Select the mwrc1\*.nii (and possibly the mwrc2\*.nii) files to smooth.

---

<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.

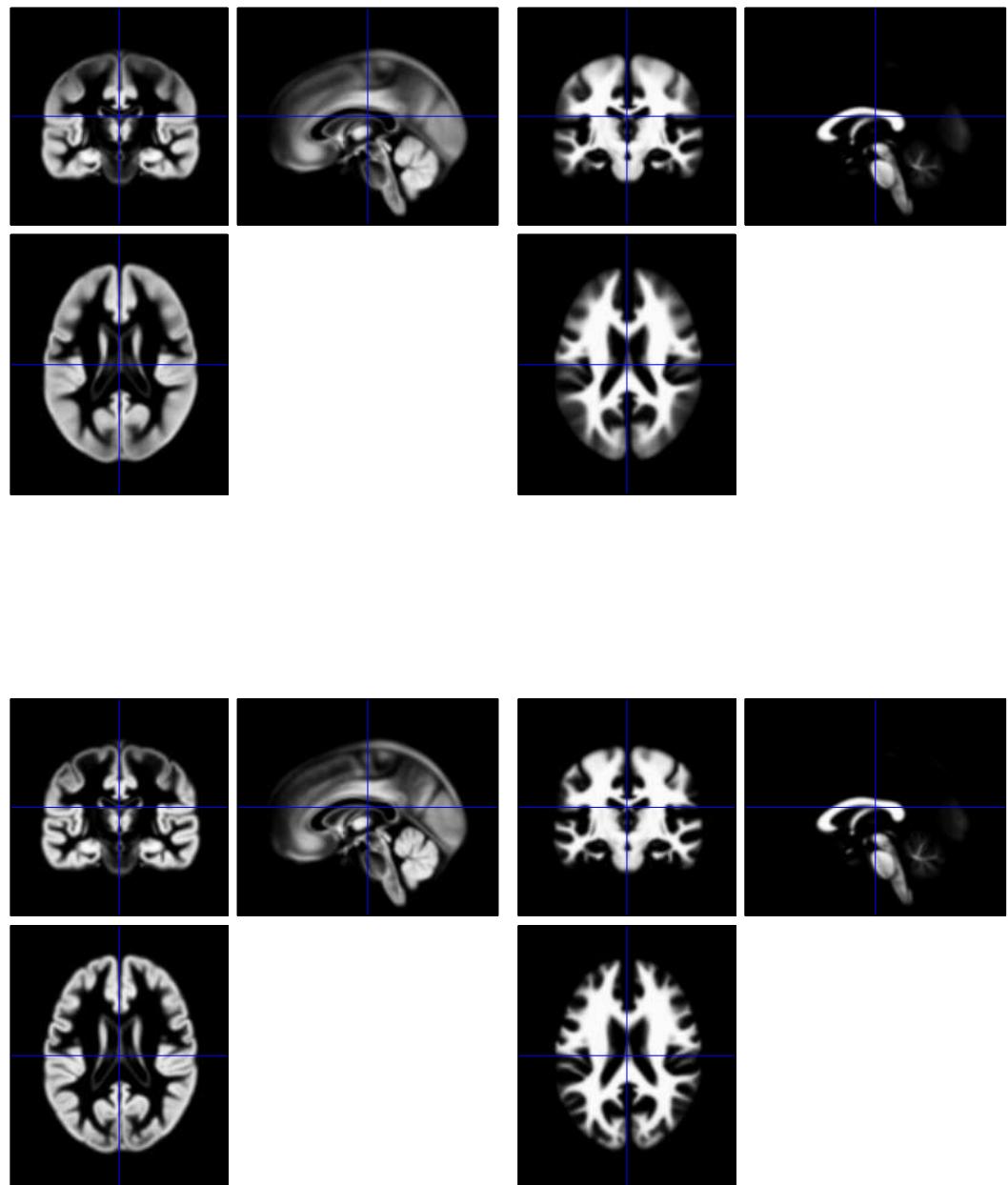


Figure 35.2: Different stages of template generation. Top row: an intermediate version of the template. Bottom row: the final template data.

- **FWHM:** 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.
- **Data Type:** There is little to gain by saving the smoothed images with greater precision, because the mwrc\*.nii files are already stored as floating point. The suggested option here is to set the data type to “SAME”, which results in the smoothed data having the same data-type as the images specified for smoothing.

The final step is to perform the statistical analysis on the preprocessed data (smwrc1\*.nii files). Note that results are not in MNI space, but rather in a coordinate system that represents the average shape and size of the subjects included in the study. The next section says a little about how data from a small number of subjects could be warped to MNI space.

## 35.2 Using DARTEL to Spatially Normalise 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 intersubject 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.

Note that DARTEL has not been thoroughly evaluated for spatially normalising fMRI. During spatial normalisation of a brain image, some regions need to expand 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.

### 35.2.1 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**

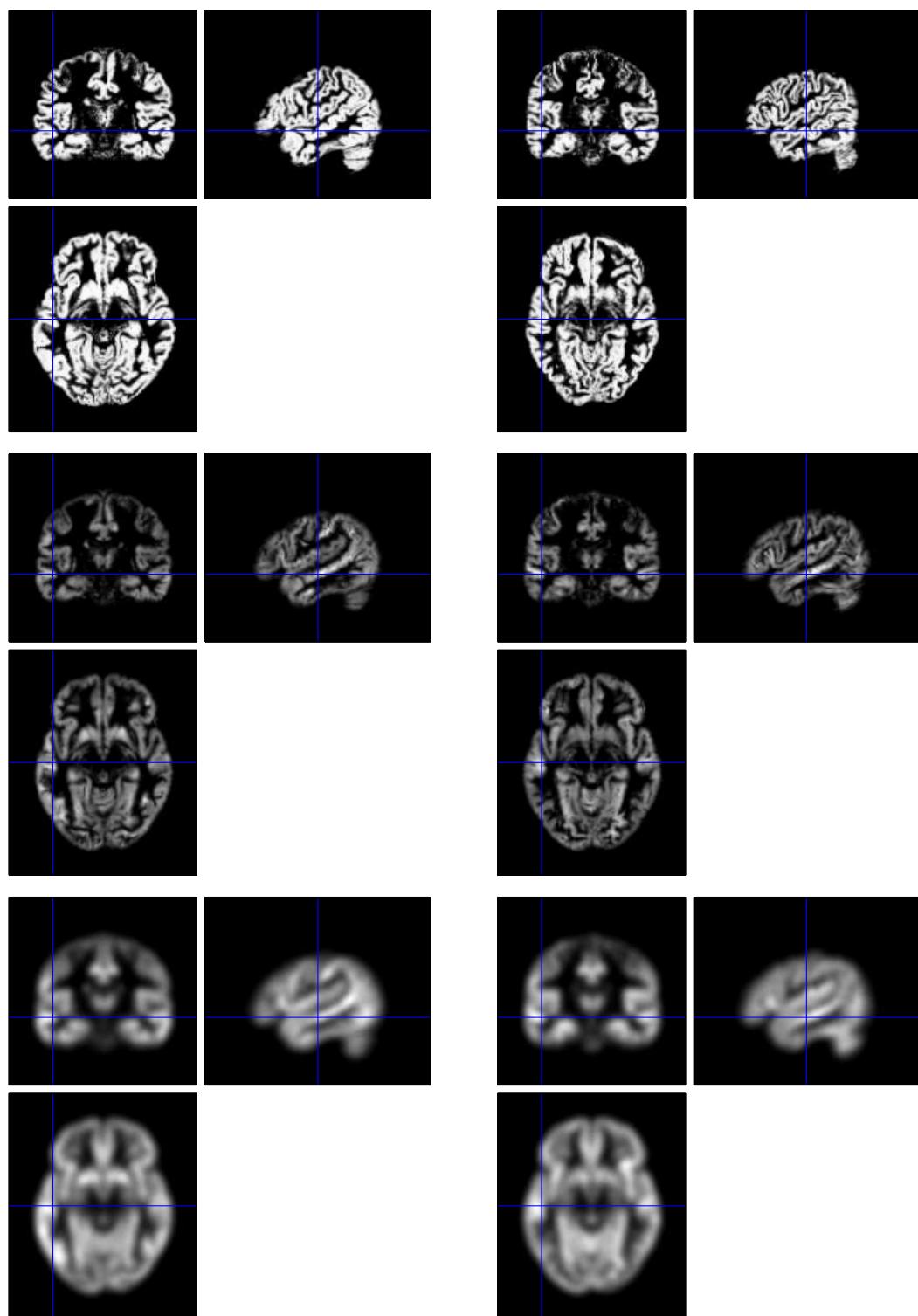


Figure 35.3: Pre-processing for VBM. Top row: Imported grey matter (rc1A.nii and rc1B.nii). Centre row: Warped and “modulated” (mwrc1A.nii and mwrc1B.nii). Bottom row: Smoothed by 8mm (smwrc1A.nii and smwrc1B.nii).

- \* **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.

### 35.2.2 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 from the *TASKS* pulldown (it is in *Utils*).

- **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).

- **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).

- **Save as:** You can save the composed deformations as a file. This would be called  $y\_*$ .nii, which contains three volumes that encode the x, y and z components of the mapping. Note that only the first (x) component can be visualised in SPM. These things were not really designed to be visualised as images anyway.
- **Apply to:** Specify the images for that subject that you would like spatially normalised. Note that the spatially normalised images are not masked (see the Chapter on Realignment for more information here). If realignment parameters are to be incorporated into the transformation, then this could cause problems at the edges. These can be avoided by reslicing after realignment (which is the default option if you “Realign Unwarp”). Alternatively, some form of additional masking could be applied to the spatially normalised images, prior to smoothing.
- **Interpolation:** Specify the form of interpolation.

The above procedure would be repeated for each subject in the study.

### 35.3 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 *TASKS –> 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).

### 35.4 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 *TASKS –> 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 flow**

- **Flow Field:** Specify the u\_rc1B\_Template.nii flow field.
  - **Forward/Backwards:** Forward.
  - **Time Steps:** Usually 64.
- \* **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).
  - **Save as:** You can save the composed deformations as a file. This would be called y\_\*.nii, which contains three volumes that encode the x, y and z components of the mapping.
  - **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.*
  - **Interpolation:** Specify the form of interpolation.

Suppose the image of one subject has been manually labeled, then this option is useful for transferring the labels on to images of other subjects.

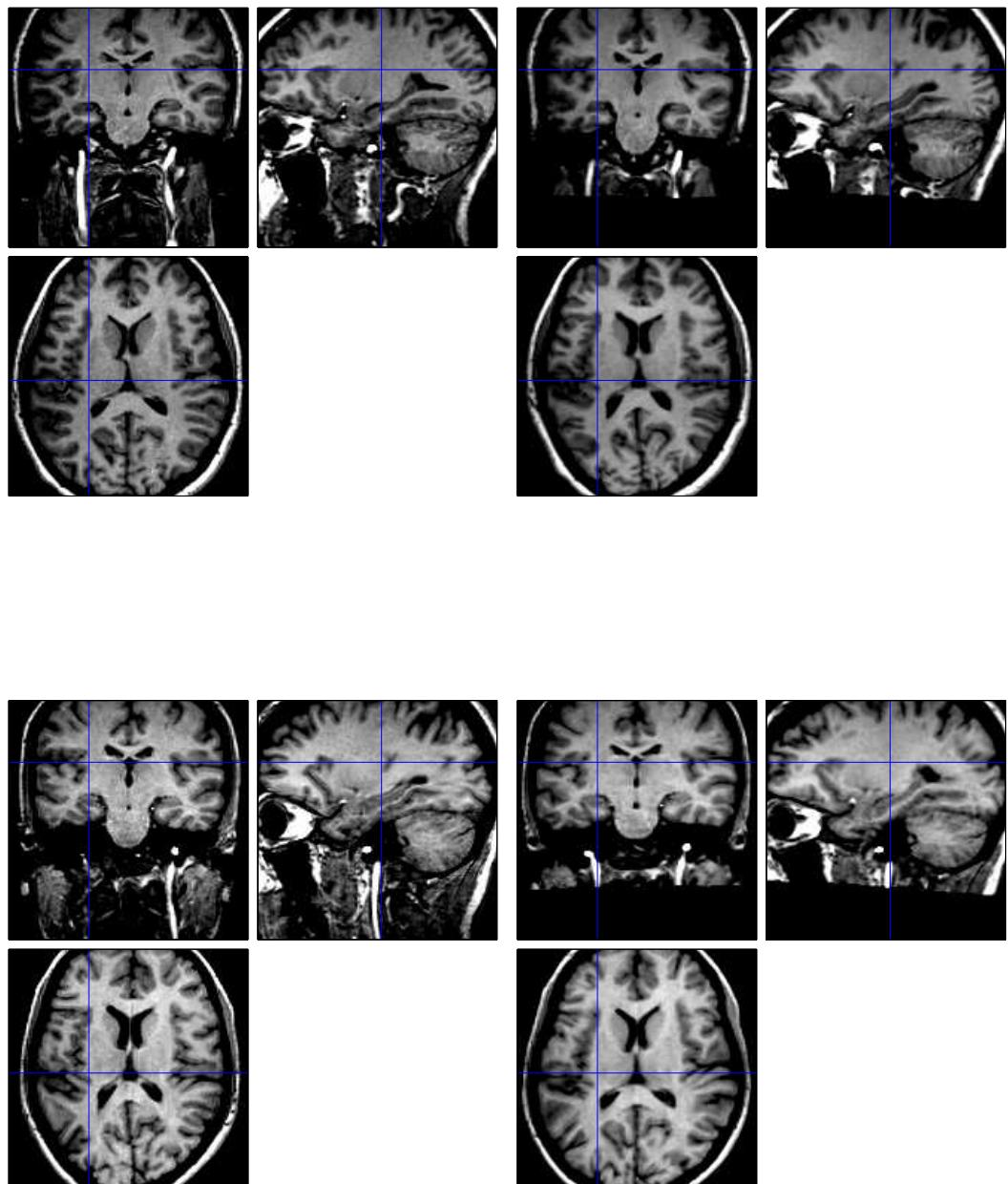


Figure 35.4: Composition of deformations to warp one individual to match another. Top-left: Original A.nii. Top-right: A.nii warped to match B.nii. Bottom-left: Original B.nii. Bottom-right: B.nii warped to match A.nii.



## **Part VIII**

# **Batching and User Interface**



# Chapter 36

## Batch interface

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. SPM8 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.

### 36.1 Batch tutorial

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 30). 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.

Subject specific input data (original functional and structural MRI data, subject specific experiment parameters) should be entered after the batch template has been saved.

---

<sup>1</sup><http://sourceforge.net/projects/matlabbatch>

### 36.1.1 Study specific input data

This dataset consists of fMRI data acquired in a single session and a structural MRI. See section 36.2 to learn how to deal efficiently with multi-session data. MRI parameters and experiment details are described in chapter 30.

### 36.1.2 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

### 36.1.3 Add modules to the batch

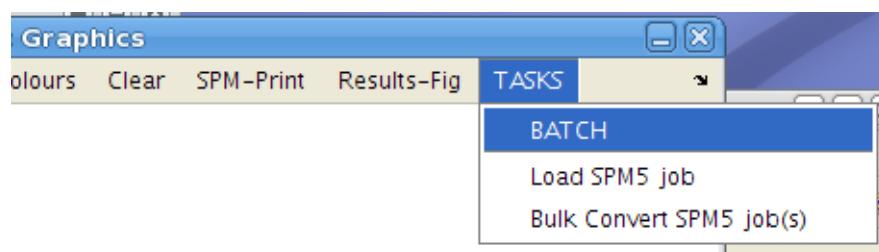


Figure 36.1: TASKS menu in “Graphics” window

The helper modules and the SPM processing modules can be assembled using the GUI. Locate the Graphics window and open the batch editor by selecting “BATCH” from the “TASKS” menu (fig. 36.1). 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.

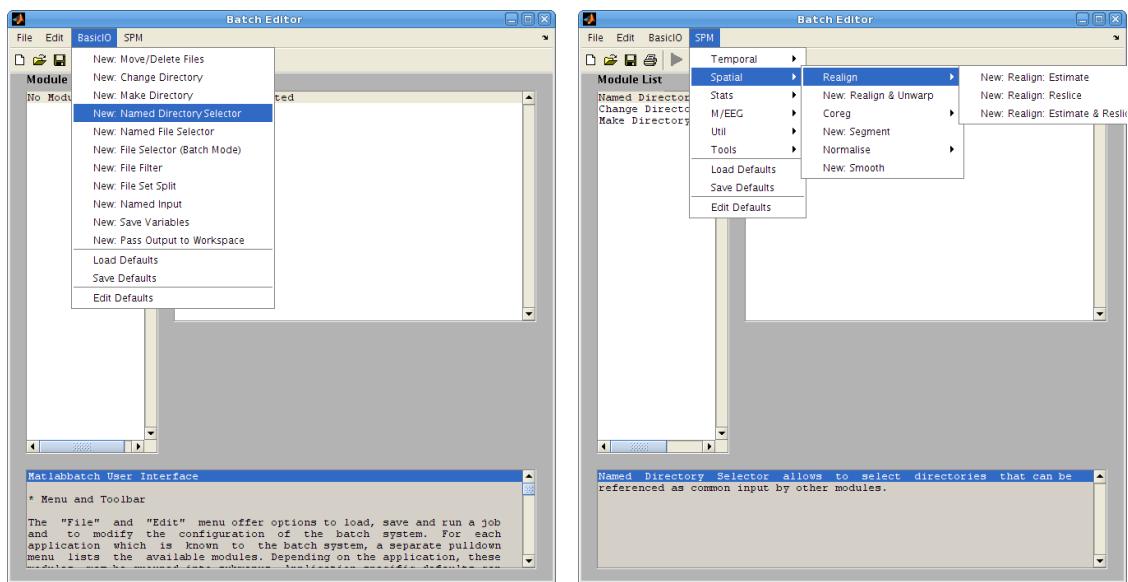


Figure 36.2: BasicIO and SPM application menus

### 36.1.4 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 30. 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 30.

The file `man/batch/face_template_single_subject_nodeps.m` contains the batch after all modules have been added and subject-independent data has been entered.

#### 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”, “Micrometre resolution”, “Micrometre 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  $\langle -X \rangle$ , 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.

### 36.1.5 Data flow

In chapter 30, 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 36.3. 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 36.3, modules in the batch can now be connected. The batch containing all dependencies can be found in `man/batch/face_template_single_subject.m`.

Again, start editing at the top of the batch:

#### Named Directory Selector

Nothing to enter now.

#### 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 “categorial” 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)”

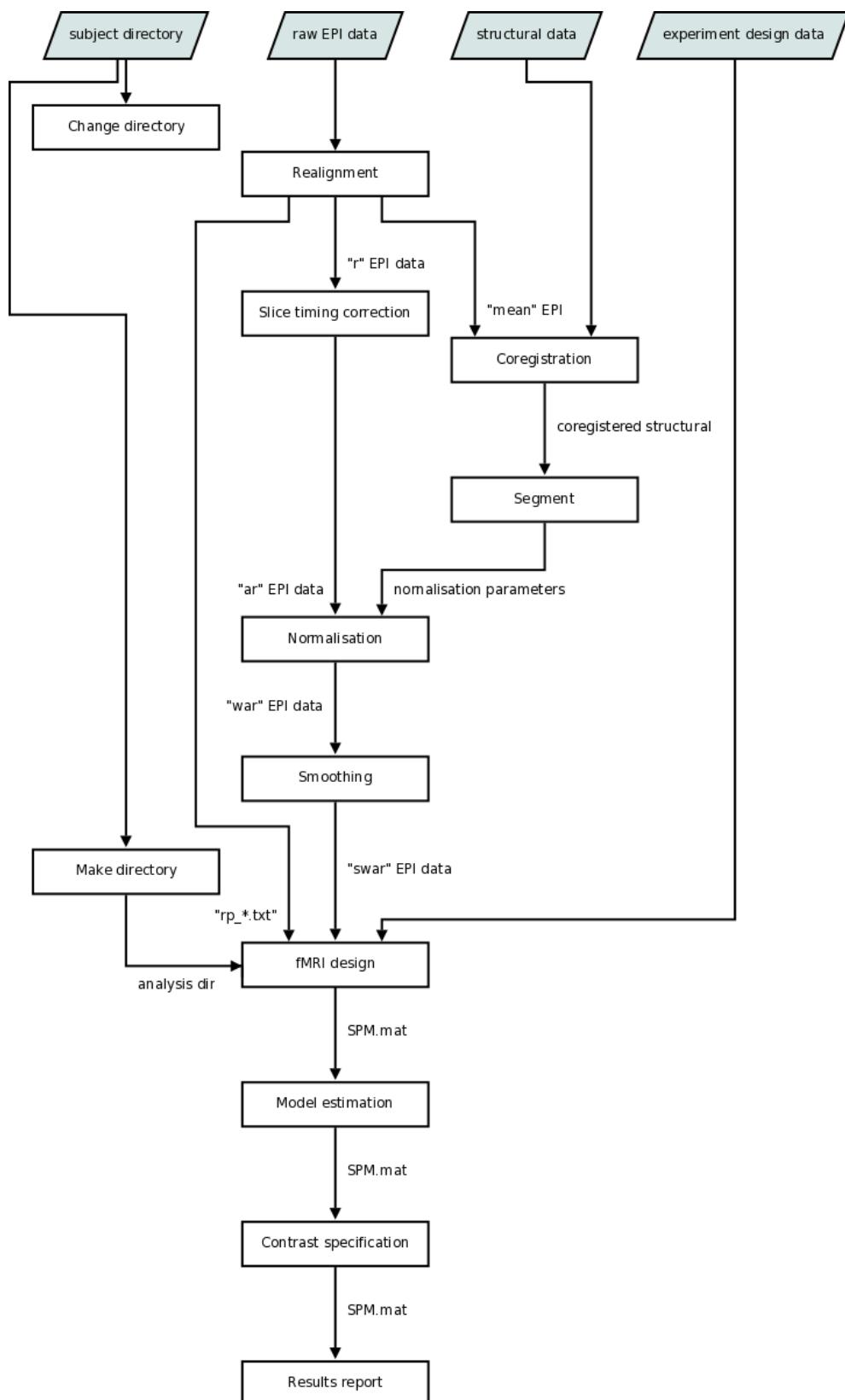


Figure 36.3: Flow chart for batch

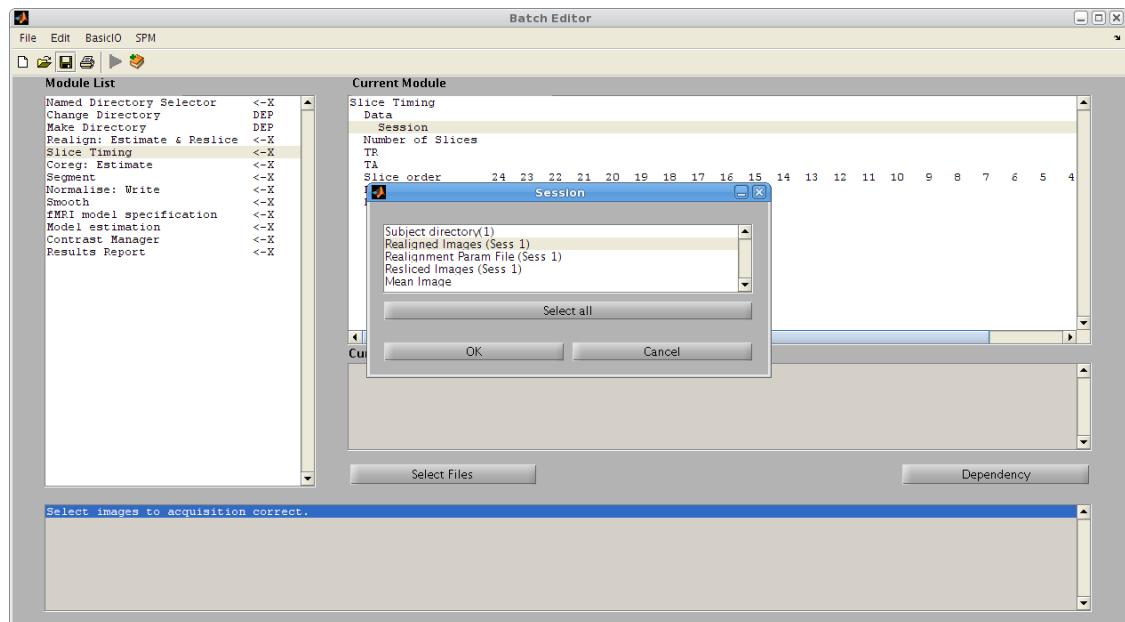


Figure 36.4: Dependency selection

### 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 36.2 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)”.

### 36.1.6 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 36.2.2 for details.

## 36.2 Advanced features

### 36.2.1 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:

**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.

### 36.2.2 Command line interface

#### Complete and run a pre-specified job

```
cfg_serial(guifcn, job[, input1, input2 ...])
```

This interface is called the “serial” interface. It takes a job, and asks for the input to any open configuration items one after another. If a list of inputs is supplied, these will be filled in (if they are appropriate). After all inputs are filled, the job will be run. The first argument, `guifcn` is a function handle that will be called to run a GUI if there are insufficient inputs or inputs that do not match a configuration item specification. For unattended mode, this may be left empty.

The following MATLAB code snippet can be used to fill in and run the `face_template_single_subject.m` batch. Instead of using `cfg_getfile` as file selector, files could be read from e.g. a MATLAB variable:

```
% Collect missing inputs
subjdir = cellstr(cfg_getfile([1 1], 'dir', 'Subject Dir'));
subjepi = cellstr(cfg_getfile([1 inf], 'image', 'Raw EPI images'));
subjana = cellstr(cfg_getfile([1 1], 'image', 'Anatomy image'));
subjcon = cellstr(cfg_getfile([1 1], 'mat', 'Multiple conditions'));
% Run batch, assuming face_template_single_subject.m is in your
% MATLAB path or working directory
cfg_serial([], 'face_template_single_subject', ...
    subjdir, subjepi, subjana, subjcon);
```

## 36.3 SPM5 to matlabbatch transition guide

This is a short overview to describe code organisation and interfaces between SPM and the batch system.

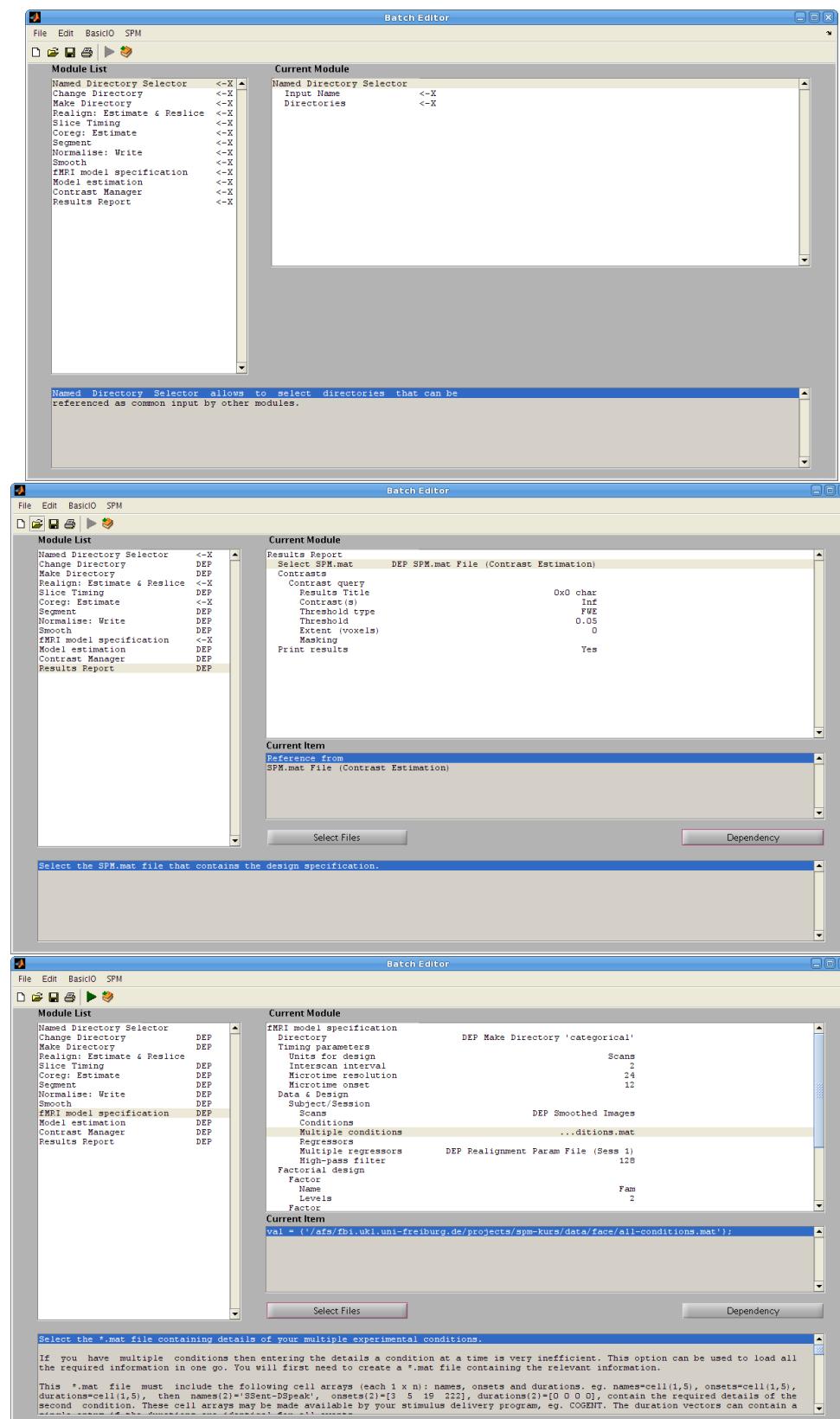


Figure 36.5: All stages of batch entry

### 36.3.1 Code Reorganisation

The following paths have changed:

- `fullfile(spm('dir'), 'matlabbatch')` Core batch system.
- `fullfile(spm('dir'), 'config')` New SPM config files.
- `fullfile(spm('dir'), 'oldconfig')` Old SPM config files (unused)
- `spm_jobman.m` and `spm_select.m` replaced with compatibility code
- `spm_Menu.fig` Callbacks adapted

Configuration code has been generated automatically from the existing SPM configuration using `cfg_struct2cfg` and `gencode`. This sometimes results in redundant/duplicate code. Also, conditional constructs like `if`, `case` may not have been considered.

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.

For details, see section [36.4](#).

### 36.3.2 Interfaces between SPM and Matlabbatch

**Unchanged** harvested job structure.

**Changed** Top-level node in SPM config now called `spmjobs` instead of `jobs`. New overall top-level node `matlabbatch`. `spm_jobman` will convert and load SPM5 style batch jobs into the new batch system.

**Changed** Configuration file syntax - instead of structs, configuration items are now objects. Structs of type `<type>` are now represented as objects of class `cfg_<type>`. Existing SPM5 configuration can be imported using `cfg_struct2cfg`. There is a new class `cfg_exbranch` which is used for branches that have a `.prog` field.

**Deprecated** Virtual files have been replaced by dependencies. These 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. Virtual files are treated as a special output argument.

**Added** 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` GUI - inspired by `spm_jobman`, but modified to work around some MATLAB GUI “features” (like input widgets loosing focus before editing has finished).

## 36.4 Configuration Code Details

Configuration code has been 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.

### 36.4.1 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 evalution of `.vout`.

The “target” part will be filled by the configuration classes, the `src_exbranch` field is set in `cfg_util`.

### 36.4.2 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. Otherwise, if there are only SPM5-style `*_config_*.m` files, the configuration will be converted at run-time using `cfg_struct2cfg`.

### 36.4.3 Defaults Settings

Due to automatic code generation from the previous configuration, some defaults are coded in the configuration files. This code will be removed, all defaults will be read from `spm_def.m` in the configuration folder. This file currently links to `spm_defaults.m` and contains code to include toolbox-specific defaults in a way similar to toolbox autodetection (i.e. they appear under `defaults.tools.<toolbox_tag>` in the defaults structure).

### 36.4.4 Toolbox Migration

In the `fullfile(spm('dir'), 'toolbox')` folder there exists a migration utility `spm_tbx_config2cfg.m`. This utility will create a `*_cfg_*.m` and a `*_def_*.m` file based on the configuration tree given as input argument.

## 36.5 Utilities

### 36.5.1 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 as dependency source if multiple functions require the same set of files as 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 generate a batch configuration file with the batch system itself.

### 36.5.2 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.

### 36.5.3 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.

# 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*, 38:95–113, 2007.
- [4] J. Ashburner, J. Andersson, and K.J. Friston. Image registration using a symmetric prior - in three-dimensions. *Human Brain Mapping*, 9(4):212–225, 2000.
- [5] 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.
- [6] J. Ashburner and K.J. Friston. Nonlinear spatial normalization using basis functions. *Human Brain Mapping*, 7(4):254–266, 1999.
- [7] J. Ashburner and K.J. Friston. Voxel-based morphometry – the methods. *NeuroImage*, 11:805–821, 2000.
- [8] J. Ashburner and K.J. Friston. Why voxel-based morphometry should be used. *NeuroImage*, 14(6):1238–1243, 2001.
- [9] J. Ashburner and K.J. Friston. Unified segmentation. *NeuroImage*, 26:839–851, 2005.
- [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.*, 14:239–256, 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, 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.
- [16] 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.

- [17] R. W. Cox and A. Jesmanowicz. Real-time 3D image registration for functional MRI. *Magnetic Resonance In Medicine*, 42:1014–1018, 1999.
- [18] 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.
- [19] 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.
- [20] W. F. Eddy, M. Fitzgerald, and D. C. Noll. Improved image registration by using Fourier interpolation. *Magnetic Resonance in Medicine*, 36:923–931, 1996.
- [21] J.J. Ermer, J.C. Mosher, S. Baillet, and R.M. Leahy. Rapidly recomputable EEG forward models for realistic head shapes. *Phys. Med. Biol.*, 46(4):1265–1281, 2001.
- [22] K.J. Friston, C. Frith, R.S.J. Frackowiak, and R. Turner. Characterizing dynamic brain responses with fMRI. *NeuroImage*, 2:166–172, 1995.
- [23] 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.
- [24] K.J. Friston, L. Harrison, and W.D. Penny. Dynamic Causal Modelling. *NeuroImage*, 19(4):1273–1302, 2003.
- [25] 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.
- [26] 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.
- [27] K.J. Friston and W.D. Penny. Posterior probability maps and SPMs. *NeuroImage*, 19(3):1240–1249, 2003.
- [28] K.J. Friston, W.D. Penny, and D.E. Glaser. Conjunction revisited. *NeuroImage*, 2005.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] R.N.A. Henson and W.D. Penny. ANOVAs and SPM. Technical report, Wellcome Department of Imaging Neuroscience, 2003.
- [33] R.N.A. Henson, C.J. Price, M.D. Rugg andR. Turner, and K.J. Friston. Detecting latency differences in event-related bold responses: application to words versus non-words and and initial versus repeated face presentations. *NeuroImage*, 15(1):83–97, 2002.
- [34] R.N.A. Henson, M.D. Rugg, and K.J. Friston. The choice of basis functions in event-related fMRI. *NeuroImage*, 13(6):127, June 2001. Supplement 1.
- [35] 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.
- [36] M.X. Huang, J.C. Mosher, and R.M. Leahy. A sensor-weighted overlapping-sphere head model and exhaustive head model comparison for MEG. *Physics in Medicine and Biology*, 44(2):423–440, 1999.

- [37] 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.
- [38] 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.
- [39] M. Jenkinson. Fast, automated, N-dimensional phase-unwrapping algorithm. *Magnetic Resonance in Medicine*, 49:193–197, 2003.
- [40] 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.
- [41] 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.
- [42] 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.
- [43] S.J. Kiebel and K.J. Friston. Statistical Parametric Mapping for Event-Related Potentials I: Generic Considerations. *NeuroImage*, 22(2):492–502, 2004.
- [44] S.J. Kiebel and K.J. Friston. Statistical Parametric Mapping for Event-Related Potentials II: A Hierarchical Temporal Model. *NeuroImage*, 22(2):503–520, 2004.
- [45] S.J. Kiebel, Stefan Klppel, N. Weiskopf, and K.J. Friston. Dynamic causal modeling: A generative model of slice timing in fMRI. *NeuroImage*, 34:1487–1496, 2007.
- [46] J. Kilner, S.J. Kiebel, and K.J. Friston. Applications of random field theory to electrophysiology. *Neuroscience Letters*, 374:174–178, 2005.
- [47] 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.
- [48] J. Mattout, M. Péligrini-Issac, L. Garnero, and H. Benali. Multivariate Source Prelocalization (MSP): use of functionally informed basis functions for better conditioning the MEG inverse problem. *NeuroImage*, 26(2):356–373, 2005.
- [49] J. Mattout, C. Phillips, W.D. Penny, M. Rugg, and K.J. Friston. Meg source localisation under multiple constraints: an extended Bayesian framework. *NeuroImage*, 2006.
- [50] 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.
- [51] 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.
- [52] W.D. Penny and G. Flandin. Bayesian analysis of single-subject fMRI: SPM implementation. Technical report, Wellcome Department of Imaging Neuroscience, 2005.
- [53] W.D. Penny, G. Flandin, and N. Trujillo-Bareto. Bayesian Comparison of Spatially Regularised General Linear Models. *Human Brain Mapping*, 28:275–293, 2005.
- [54] 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.
- [55] W.D. Penny, S.J. Kiebel, and K.J. Friston. Variational Bayesian Inference for fMRI time series. *NeuroImage*, 19(3):727–741, 2003.

- [56] W.D. Penny, K.E. Stephan, A. Mechelli, and K.J. Friston. Comparing Dynamic Causal Models. *NeuroImage*, 22(3):1157–1172, 2004.
- [57] W.D. Penny, N. Trujillo-Bareto, and K.J. Friston. Bayesian fMRI time series analysis with spatial priors. *NeuroImage*, 24(2):350–362, 2005.
- [58] 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.
- [59] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C (Second Edition)*. Cambridge, Cambridge, 1992.
- [60] L. Spinelli, S.G. Andino, G. Lantz, M. Seeck, and C.M. Michel. Electromagnetic inverse solutions in anatomically constrained spherical head models. *Brain Topography*, 13(2):115–125, 2000.
- [61] 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.
- [62] P. Thévenaz, T. Blu, and M. Unser. Interpolation revisited. *IEEE Transactions on Medical Imaging*, 19(7):739–758, 2000.
- [63] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing: Part I – theory. *IEEE Transactions on Signal Processing*, 41(2):821–833, 1993.
- [64] 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.
- [65] 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.
- [66] K.J. Worsley. Developments in random field theory. 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.
- [67] I.C. Wright, P.K. McGuire, J.B. Poline, J.M. Travere, 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.