SPM8 Manual

The FIL Methods Group
(and honorary members)

John Ashburner
Gareth Barnes
Chun-Chuan Chen
Jean Daunizeau
Guillaume Flandin
Karl Friston
Darren Gitelman
Stefan Kiebel
James Kilner
Vladimir Litvak
Rosalyn Moran
Will Penny
Klaas Stephan

Darren Gitelman
Rik Henson
Chloe Hutton
Volkmar Glauche
Jérémie Mattout
Christophe Phillips

Contents

Part I

Temporal processing

Part II

Spatial processing

Part III

fMRI Statistics

Chapter 1

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

1.1 Timing parameters

Specify various timing parameters needed to construct the design matrix. This includes the units of the design specification and the interscan interval.
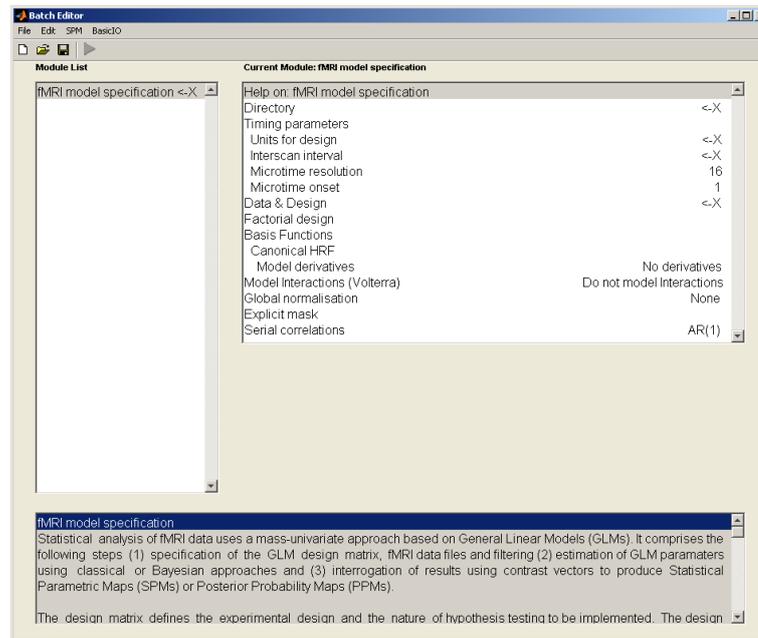
Figure 1.1: *After starting SPM in fMRI mode and pressing the "Specify 1st-level" button, the SPM batch editor window should appear as above. The options for "fMRI model specification" can be examined by clicking on them. A single click will bring up some help text in the lower subwindow (not shown in the above graphic). Options highlighted with a "<-X" are mandatory and must be filled in by the user. Each of the options shown above is described in this chapter.*

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

1.1.1   Units for design

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

1.1.2   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.

1.1.3   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.

1.1.4   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

Statistical analysis: Design
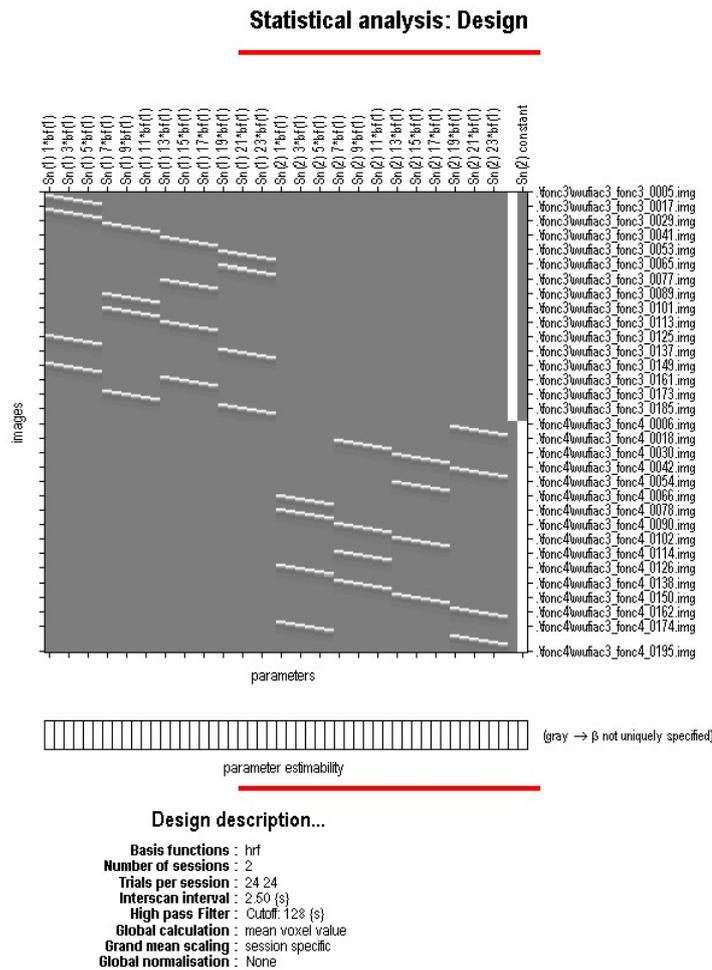
Figure 1.2: *Design matrix for fMRI data from two sessions. There are 24 experimental conditions for each session. The last two columns model the average activity in each session, giving a total of 50 regressors. There are 191 fMRI scans for each session. The overall design matrix therefore has 382 rows and 50 columns.*

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.

1.2 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 1.2 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.

1.2.1   Subject/Session

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

Scans

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

Conditions

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

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

   Name   Condition Name

   Onsets   Specify a vector of onset times for this condition type. This can be entered using
the keyboard eg. typing in "100 300" and then hitting return or "100;300" or "[100,300]" or
"[100,300]".
   More usually, however, this specification takes place using variables that have been created
before and loaded into matlab. For example, an `my_onsets` cell array[1] might exist in a file
you created earlier called `my_design.mat`. You would then type `load my_design` at the matlab
command prompt before pressing the 'Specify 1st-level' button.
   You could then specify the onsets for condition 2 by typing in eg. `my_onsets{2}` instead of
entering the numbers via the keyboard.

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

   Time Modulation   This option allows for the characterisation of nonstationary responses.
Specifically, you can model either linear or nonlinear time effects. For example, 1st order modu-
lation 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.

---

[1]Cell arrays are usually used in preference to matrices as different event types can then have different numbers
of events.

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

See [8, 6] for further details of parametric modulations.

Multiple conditions

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

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

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

Regressors

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

Regressor

Name Enter name of regressor eg. First movement parameter

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

Multiple regressors

If you have 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 1.10. 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.

1.3   Factorial design

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

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

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

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

See [33] for more information on SPM and factorial designs.

1.3.1   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

1.4   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.

1.4.1   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 [35].

1.4.2   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.

1.5   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 occuring in isolation. This type of 'sub-linear' response can be modelled using Volterra kernels. See [22] for further details.

1.6   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.

1.7   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.

1.8   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.

1.9 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 [25, 59] for further discussion of these issues.

1.10 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 1.3.
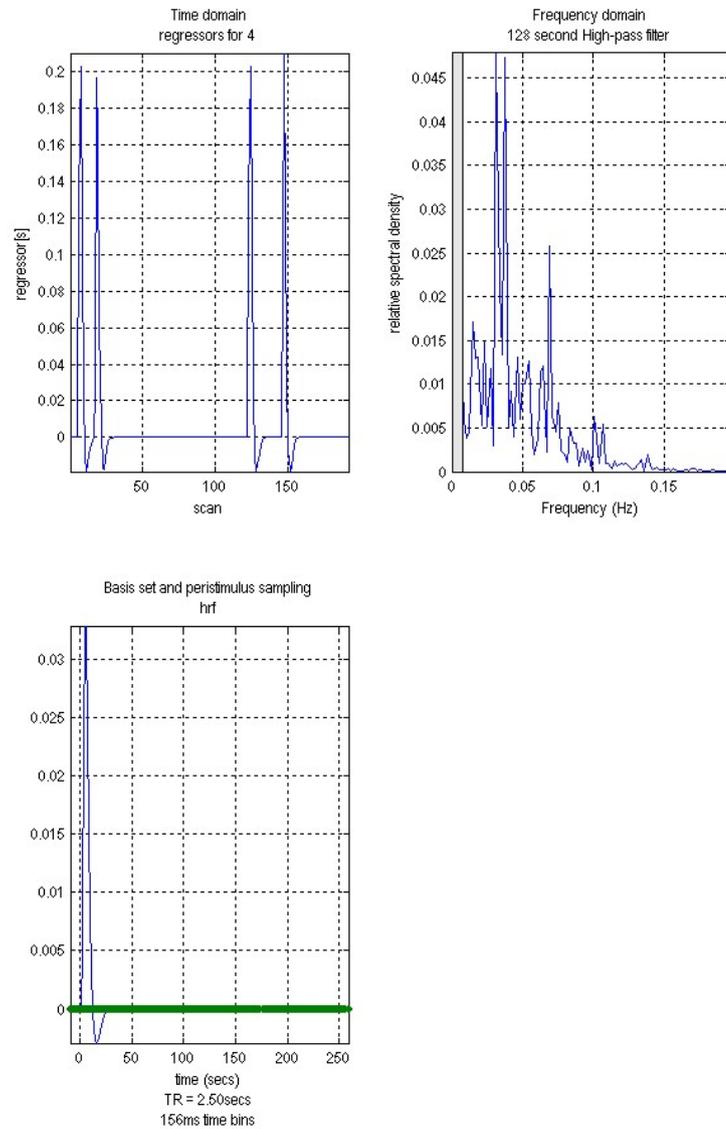
Figure 1.3: *After pressing "Review", selecting the pull-down 'Design' menu, Explore->Session, and selecting the regressor you wish to look at, you should get a plot similar to the one above. The top row shows time and frequency domain plots of the time-series corresponding to this regressor. In this 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 2

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.

2.1 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.

2.2 Method

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

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

2.2.1 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 [25, 18] 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.
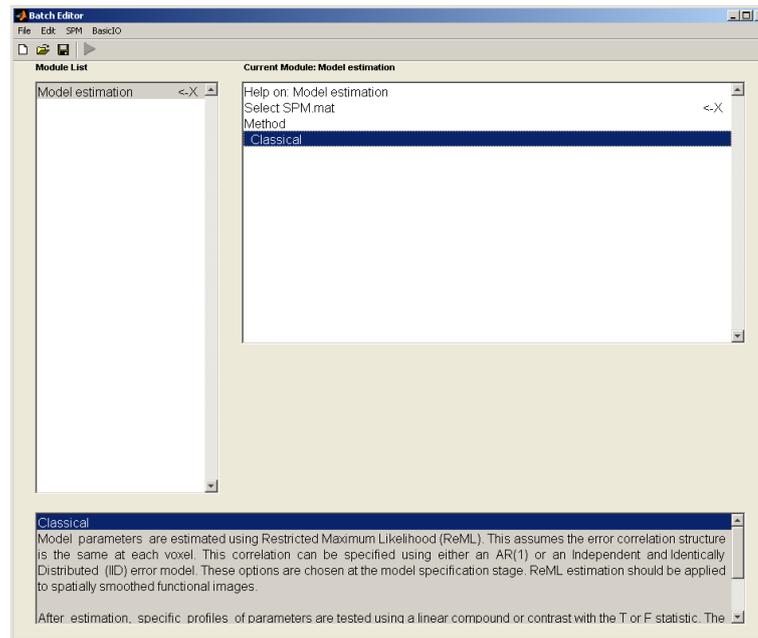
Figure 2.1: *After starting SPM in fMRI mode and pressing the "Estimate" button, the SPM batch editor window should appear as above. The options for "fMRI model estimation" can be examined by clicking on them. A single click will bring up some help text in the lower subwindow (not shown in the above graphic). Options highlighted with a '<-X' are mandatory and must be filled in by the user. Each of the options shown above is described in this chapter.*

2.2.2   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 [59, 63, 56, 57].

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) [23].

Analysis Space

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

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

    Block type   Enter the block type, i.e. "Slices" or "Subvolumes".

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

    Slice numbers   Enter Slice Numbers.
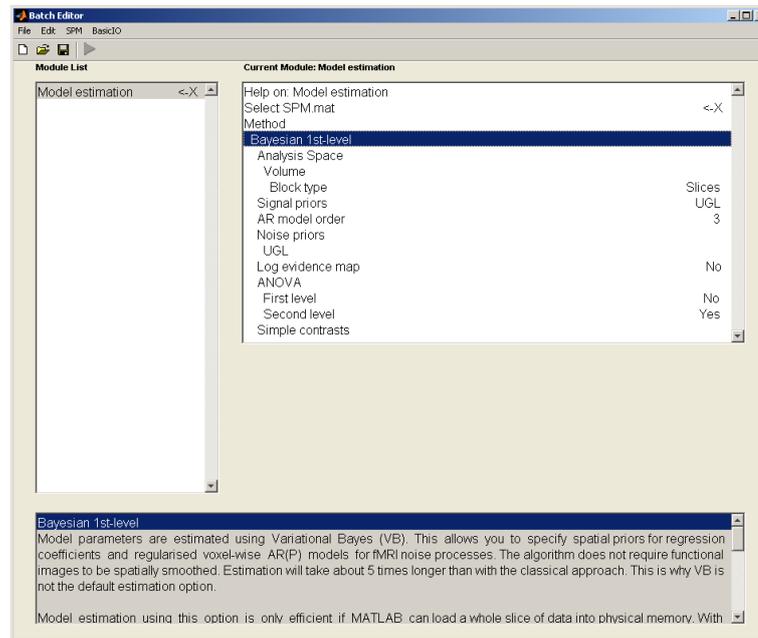
Figure 2.2: *After choosing "Bayesian 1st-level" under "Method", the SPM batch editor window should appear as above. Each of the options shown above is described in this chapter.*

Block type   Enter the block type, i.e. "Slices" or "Subvolume".

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

Cluster mask   Select cluster image.

Block type   Enter the block type, i.e. "Slices" or "Subvolumes".

Signal priors

- **[UGL] Unweighted Graph Laplacian.** This spatial prior is the recommended option. Regression coefficients at a given voxel are (softly) constrained to be similar to those at nearby voxels. The strength of this constraint is determined by a spatial precision parameter that is estimated from the data. Different regression coefficients have different spatial precisions allowing each putative experimental effect to have its own spatial regularity.

- **[GMRF] Gaussian Markov Random Field.** This is equivalent to a normalized UGL.

- **[LORETA] Low resolution Tomography Prior.** This is equivalent to UGL squared. It is a standatd choice for EEG source localisation algorithms.

- **[WGL] Weighted Graph Laplacian.** This is a generalization of the UGL, where weights can be used to preserve "edges" of functional responses.

- **[Global] Global Shrinkage prior.** This is not a spatial prior in the sense that regression coefficients are constrained to be similar to neighboring voxels. Instead, the average effect over all voxels (global effect) is assumed to be zero and all regression coefficients are shrunk towards this value in proporation to the prior precision. This is the same prior that is used for Bayesian estimation at the second level models, except that here the prior precision is estimated separaetly for each slice.

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

AR model order

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

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

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

Noise priors

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

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

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

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

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

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

Robust   Robust GLM. Uses Mixture of Gaussians noise model.

Log evidence map

Computes the log evidence for each voxel

ANOVA

Perform 1st or 2nd level Analysis of Variance.

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

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

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

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

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

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

Simple contrasts

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

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

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

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

Simple contrast

Name Name of contrast eg. "Positive Effect".

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

2.2.3 Bayesian 2nd-level

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

2.3   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

2.3.1   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 kth 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 ith 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.

2.3.2   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 kth 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 pth 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 ith 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.

2.4   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 [57] 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 2.3 shows an example output from the function indicating that, for the data in this brain region, an informed basis set has the highest model evidence.
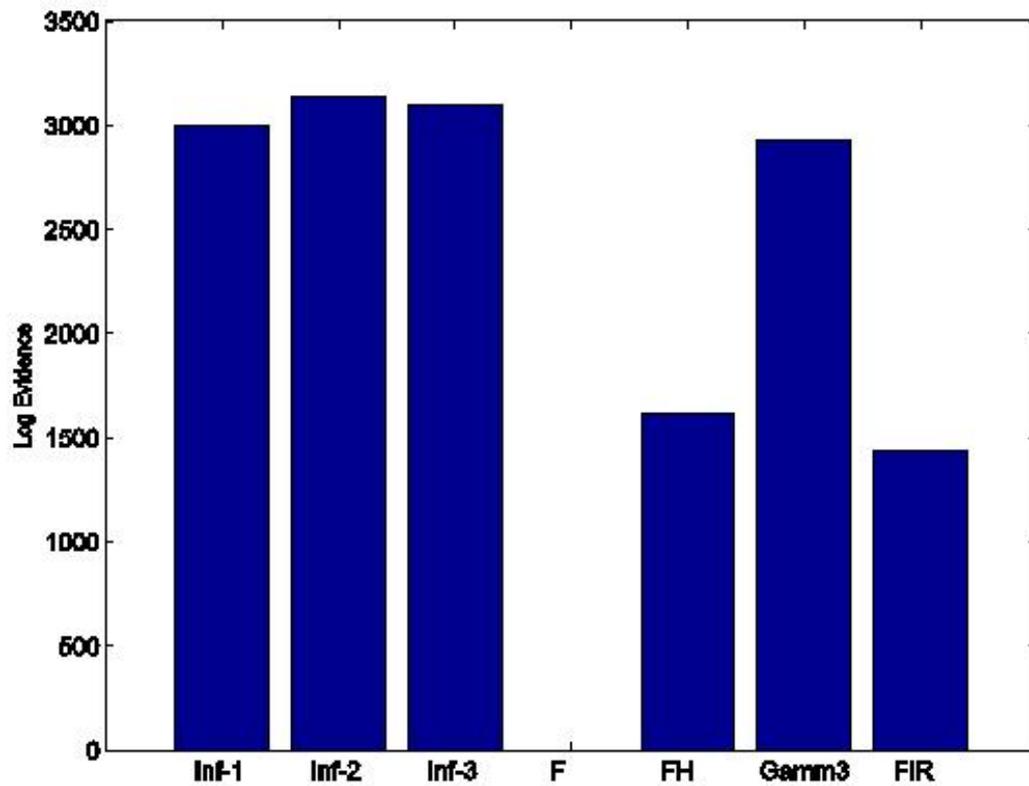
Figure 2.3: *This plot shows the model evidence for a number of different hemodynamic basis sets: Inf1 - Canonical HRF, Inf2 - Canonical plus temporal derivative, Inf3 - Canonical plus temporal and dispersion derivatives, F - Fourier, FH - Fourier with a Hanning Window, Gamm3 - 3 Gamma basis functions and FIR - a Finite Impulse Response function. An informed basis set provides the best model of the data for the selected region.*

Part IV

EEG/MEG

Chapter 3

SPM for MEG/EEG overview

3.1 Welcome to SPM for M/EEG

With SPM8 you can analyze all kinds of MEG and EEG data. Our group has recently published many articles about M/EEG analysis, in particular about Source Reconstruction[1] and Dynamic Causal Modelling[2] (DCM), which is a spatio-temporal network model to estimate effective connectivity in a network of sources. All these new methods, mostly based on a Bayesian approach, have been implemented in SPM8. We provide a range of tools for the full analysis pipeline, i.e., you can take your raw data from the MEG or EEG machine, and put it through SPM, starting from the conversion of the data through to a statistical analysis of sensor level or source reconstructed multi-subject data or Dynamic Causal Modelling.

Our overall goal is to provide an academic M/EEG analysis software package that can be used by everyone to apply the most recent methods available for the analysis of M/EEG data. As you may guess, this goal is quite ambitious because there is a large number of different M/EEG formats available, plus there are literally dozens of different analysis strategies that researchers would like to use. Clearly, our rather small group doesn't have the resources to cover all these different approaches. However, we made SPM for M/EEG as open as it possibly can be to allow researchers to use their favourite analysis software for specific processing steps. For example, it is possible to convert raw data to SPM8, then convert them to FieldTrip[3] or EEGLAB[4] (using an SPM conversion routine), use a couple of functions in these packages, convert back to SPM, and do source reconstruction or DCM. Any combination of processing steps should be possible, and we expect that this software-interoperability among analysis software packages (each with its own area of expertise) will lead to a boost of M/EEG researchers trying out new ways of analysing their data with a wide range of sophisticated methods. We are pleased to say that we have a formal collaboration with the excellent FieldTrip package (head developer: Robert Oostenveld, F.C. Donders centre in Nijmegen/Netherlands) on many analysis issues. For example, SPM and FieldTrip share routines for converting data to MATLAB, forward modelling for M/EEG source reconstruction and the SPM8 distribution contains a version of FieldTrip so that you can combine FieldTrip and SPM functions in your custom scripts. SPM and FieldTrip complement each other well, as SPM is geared toward very specific analysis tools as will be described below, whereas FieldTrip is a more general repository of different methods that can be put together in flexible ways to perform a variety of analyses. This flexibility of FieldTrip, however, comes at the expense of accessibility to a non-expert user. FieldTrip does not have a graphical user interface (GUI) and its functions are used by writing custom MATLAB scripts. By combining SPM8 and FieldTrip the flexibility of FieldTrip can be complemented by SPM's GUI tools and batching system. Within this framework, power users can easily and rapidly develop specialized analysis tools with GUIs that can then also be used by non-proficient MATLAB users. Some examples of such tools are available in the MEEG toolbox distributed with SPM. We will also be happy to

---

[1]Source Reconstruction: http://www.fil.ion.ucl.ac.uk/spm/doc/biblio/Keyword/EEG.html
[2]Dynamic Causal Modelling: http://www.fil.ion.ucl.ac.uk/spm/doc/biblio/Keyword/DCM.html
[3]FieldTrip: http://www.ru.nl/neuroimaging/fieldtrip/
[4]EEGLAB: http://sccn.ucsd.edu/eeglab/

include in this toolbox new tools contributed by other users as long as they are of general interest
and applicability.

SPM's speciality is, of course, the statistical analysis of voxel-based images. For statistical
analysis, we use exactly the same routines as SPM for fMRI users would. These are robust
and validated functions based on the General Linear Model[5] (GLM) and Random Field Theory[6]
(RFT). These routines have been developed and used in the fMRI field over many years and are
equally applicable to multi- (or single-) subject M/EEG studies.

Furthermore, our group has invested heavily in establishing Bayesian approaches to the source
reconstruction of M/EEG data. Good source reconstruction techniques are vital for the M/EEG
field, otherwise it would be very difficult to relate sensor data to neuroanatomy or findings from
other modalities like fMRI. Bayesian source reconstruction provides a principled way of incorpo-
rating prior beliefs about how the data were generated, and enables principled methods for model
comparison. With the use of priors and Bayesian model comparison, M/EEG source reconstruc-
tion is a very powerful neuroimaging tool, which has a unique macroscopic view on neuronal
dynamics.

    In addition, we have taken the idea of Dynamic Causal Modelling (DCM) from the fMRI
domain, and applied it to the M/EEG field. For M/EEG, DCM is a powerful technique, because
the data are highly resolved in time and this makes the identifiability of neurobiologically inspired
network models feasible. This means that DCM can make inferences about temporal precedence
of sources and can quantify changes in feedforward, backward and lateral connectivity among
sources on a neuronal time-scale of milliseonds. Note that DCM/fMRI won't do this for you;
DCM/fMRI (or any other connectivity analysis in fMRI) looks at the modulation of connectivity
by task, on a time-scale of seconds.

3.2   Changes from SPM5 to SPM8

As in SPM5, SPM8 provides tools for the analysis of EEG and MEG data. However, the SPM8
release is much more robust than the previous version, in many aspects such as conversion of
data, source reconstruction, and dynamic causal modelling.

For three years, we have collected valuable experience for the analysis of M/EEG data, and
received much valuable feedback from both FIL and external collaborators and users. We had
plenty of opportunity to see which things worked well and what can be improved. One of our
major insights was that writing a general routine for conversion of M/EEG data from their native
to our SPM-format is a major effort. This is simply because there are so many different formats
around and it is quite an undertaking for a small development team like ours to write stable
software which can read all formats, some of which we have never seen ourselves. This had two
consequences. The first is that we now collaborate with the developers of FieldTrip, who had
already made available a wide range of MATLAB code to convert M/EEG data, and we thought
it a good idea to let both SPM and FieldTrip use and develop the same library. The second
major difference is that we changed the internal M/EEG format of SPM in many ways to make
reading/writing and manipulating M/EEG data more robust and straightforward for the user.
Effectively, we invested a lot of effort into rebuilding the SPM for M/EEG machinery almost from
scratch.

There are a couple of other major changes from SPM5 to SPM8.

    First, based on our source reconstruction work, we have implemented a number of new routines
which provide for a robust and efficient source reconstruction, using Bayesian approaches. The
resulting, voxel-based source reconstructions can then be analysed, at the group level, with the
same well-tested routines that are used for fMRI data.

    Second, Dynamic Causal Modelling, a network analysis for spatiotemporal M/EEG data,
has been developed further over the past three years. The DCM routines for modelling evoked

---

[5]GLM: http://www.fil.ion.ucl.ac.uk/spm/doc/biblio/Keyword/GLM.html
[6]RFT: http://www.fil.ion.ucl.ac.uk/spm/doc/biblio/Keyword/RFT.html

responses or fields have been significantly improved both in functionality and speed. We now provide DCMs for modelling induced responses, phase coupling and local field potentials.

Third, there are now three ways of implementing M/EEG analyses in SPM. These are the graphical user interface, and two different scripting methods (batch analysis). These batch facilities come in handy for multi-subject studies. As in fMRI analysis, many processing steps are repetitive and it is now quite straightforward to automatize the software to a high degree.

Fourth, it is now possible to convert, in working memory, SPM data to FieldTrip or EEGLAB, and back. This feature makes it possible to use, within SPM, many FieldTrip and EEGLAB functions. For example, it is quite straightforward, using a script, to work within SPM, and use FieldTrip functions to do parts of the preprocessing.

The following chapters go through all the EEG/MEG related functionality of SPM8. Most users will probably find the tutorial (chapter 18) useful for a quick start. A more extensive tutorial demonstrating many new features of SPM8 on both EEG and MEG data can be foind in 19. A further detailed description of the conversion, preprocessing functions, and the display is given in chapter 4. In chapter 5, we explain how one would use SPM's statistical machinery to analyse M/EEG data. The 3D-source reconstruction routines, including dipole modelling, are described in chapter 6. Finally, in chapter 8, we describe the graphical user interface for dynamical causal modelling, for evoked responses, induced responses, and local field potentials.

Chapter 4

EEG/MEG preprocessing −
Reference

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

These functions can be called either from SPM's graphical user interface (GUI), from the MATLAB command line, or via the batch input system. We recommend beginners use the GUI first, because this will prompt SPM to ask for all relevant information needed to process the data. The batch input system is designed for repetitive analyses of data (eg. from multiple subjects) once the user knows what should be done, and in which order. The command line facilities are very useful for writing scripts, or using SPM's history-to-script functionality to generate scripts automatically. The command line use of SPM for M/EEG will require some MATLAB knowledge.

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 approach 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 (e.g., with a script), you can specify all arguments in advance and effectively use SPM/MEEG functions in batch mode. We provide some sample batch scripts (`history_subject1.m`) in the `man\example_scripts` folder of the SPM8-distribution.

In the following we will go through the conversion of the data, specifics of the M/EEG format in SPM8, how to properly enter additional information about the channels, how to call FieldTrip-functions from SPM, a complete reference of all methods and functions, how to use the display, and finally how to script and batch the preprocessing.

4.1   Conversion of data

The first step of any analysis is the conversion of data from its native machine-dependent format to a MATLABbased, common SPM format. This format stores the data in a `*.dat` file and all other information in a `*.mat` file. The `*.mat` file contains the data structure `D` and the `*.dat` is the M/EEG data. The conversion facility of SPM is based on the "fileio" toolbox[1], which is shared between SPM8, FieldTrip and EEGLAB toolboxes and jointly developed by the users of these toolboxes. At the moment most common EEG and MEG data formats are supported. For some cases, it might be necessary to install additional MATLAB toolboxes. In this case an error message will be displayed with a link where the appropriate toolbox can be downloaded.

---

[1] fileio: http://fieldtrip.fcdonders.nl/development/fileio

If your data format is not recognized by "fileio", it will still try to read the file using the Biosig toolbox[2], if available. This can work for some EEG systems, particularly clinical ones. One of the consequences of using Biosig as 'fallback' is that you may see error messages from Biosig or mentioning Biosig if your data format is not presently supported. In such a case you can extend the "fileio" toolbox and contribute your code to us. See "fileio" page for details.

In the following we will describe the GUI-version for conversion. One could also convert data using the batch or a script. After clicking on the CONVERT button of the M/EEG GUI you will be asked to select the file to be converted. As a rule of thumb, if the dataset consists of several files, the file containing the data (which is usually the largest) should be selected. SPM can usually automatically recognize the data format and apply the appropriate conversion routine. However, in some cases there is not enough information in the data file for SPM to recognize the format. This will typically be the case for files with non-specific extensions (*.dat, *.bin, *.eeg, etc). In these cases the header-, and not the data-, file should be chosen for conversion and if it is recognized, SPM will locate the data file automatically. Note that SPM8 can also convert data in SPM5 format, where the the *.mat file should be selected. In some rare cases automatic recognition is not possible or there are several possible low-level readers available for the same format. For these cases there is an option to force SPM to use a particular low-level reader available with the batch tool or in a script (see below).

After the file is chosen, you will be asked ("Define settings?") to choose whether to define some settings for the conversion or "just read". The latter option was introduced to enable a simple and convenient conversion of the data with no questions asked. The resulting SPM M/EEG data file can then be explored with SPM's reviewing tool to determine the appropriate conversion parameters for the future. If the "just read" option is chosen, SPM will try to convert the whole dataset preserving as much data as possible. The other option - "yes" - lets you control all features of the conversion, to convert only the data that will be used in subsequent processing.

If this option is chosen, the next question will be whether to read the data as "continuous" or as "trials". Note that some datasets do not contain continuous data to begin with. These datasets should usually be converted with the "trials" option.

If the "continuous" option is chosen you will be asked ("Read everything?") whether to convert the whole file ("yes") or a subset of it ("no"). If the answer is "no" you will be asked to specify the time window in seconds. Note that if a data file contains several concatenated long segments (e.g., if the recording was paused and then resumed) only one of these segments at a time can be converted as continuous. Therefore you should specify a time window which does not cross the boundaries between segments.

If the "trial" option is selected, the next question will be where to retrieve the information about trials. There are three options. If "data" is chosen, SPM will attempt to look for information about trials in the dataset. This option is suitable for datasets that are already epoched or datasets which contain information about trials. If "define" is selected, trials can be defined based on information about events which appears in the file. The routine used for this option is identical to the one used in epoching (see below) and the resulting SPM file will be already epoched. The advantage of defining trials at conversion is that only the necessary subset of the raw data is converted. This is useful when the trials of interest are only a small subset of the whole recording. After the trial definition is completed, the results can be saved into a file. This file can be later used instead of repeating the trial definition again. This is done by selecting the third trial definition option - "file".

The next question will be about which channels should be converted. Five options are available. "all" - convert all the available channels. "meg", "eeg" - automatically detect and choose MEG and EEG channels respectively. These options may not work correctly for some EEG systems because many data formats do not provide information about what data were acquired by a specific channel. However, the most common cases are supported. "gui" - choose the channels to convert using a graphical interface. The overall selection of channels can be saved in a file and this file can later be used by choosing the "file" option.

The final question is by which name the new SPM EEG files should be written to disk. By default SPM will add the prefix spm8_ to the name of the raw data file if the data is read as continuous and espm8_ if the data is read by trials.

---

[2]Biosig: http://biosig.sourceforge.net/

SPM will now convert the data. This may take some time depending on file size and a red bar will inform you about the progress.

4.2 Converting arbitrary data

It might be the case that your data is not in any standard format but is only available as an ASCII or Excel file or as a variable in the MATLAB workspace. Then you have two options depending on whether you would be willing to use a MATLAB script or want to only use the GUI. If you want to only use the GUI, we suggest that you assemble your dataset in EEGLAB and then save it and convert to SPM as any other format. The developers of EEGLAB have invested a lot of effort into making it possible to build a dataset from scratch without using the command line or scripts and we see no reason for reproducing the same functionality in SPM.

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

FieldTrip raw struct must contain the following fields:

- `.fsample` - sampling rate (Hz)

- `.trial` - cell array of trials containing matrices with identical dimensions (channels × time).

- `.time` - cell array of time vectors (in sec) - one cell per trial, containing a time vector the same length as the second dimension of the data. For SPM, the time vectors must be identical.

- `.label` - cell array of strings, list of channel labels. Same length as the first dimension of the data.

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

An example script for converting LFP data can be found under `toolbox\Neural_Models\spm_lfp_txt2mat.m`

4.3 The M/EEG SPM format

The M/EEG format has changed from SPM5 to SPM8. There were many reasons why we decided that it was time to radically change the format. If you still have some data from your SPM5 analyses, you can convert them to SPM8 (see above).

Here are the reasons why we changed the format. Previously, we placed all header information into a struct, which was then stored in a MATLAB mat-file. When the data were read, the struct-file was put into working memory, and the data, contained in the `dat`-file, was linked to the struct, using memory-mapping. We found that making a struct available in working memory was highly problematic. This was because users would manipulate the struct but sometimes introduced an error to the format (e.g., they removed a few trials from the data but did not update all fields relating to the total number of trials). This would generate hard-to-resolve errors when trying to further process the now inconsistent data. To avoid this in the future, we have introduced two changes. The first is that SPM8 now always does a consistency check when loading a file. This means, if, for whatever reason, the data format was made inconsistent, SPM8 will now report this as soon the user tries to load these data. SPM8 will also report where the check flagged an inconsistency. If there is enough information available to fix the inconsistency, it will be fixed on the fly. For instance, you will usually see some messages about missing information starting with 'checkmeeg:' when converting a file to SPM8 format. These messages are normal and if no error is generated they do not indicate a problem. Second, when reading the data, we now convert the header struct to an object and only then make it available in working memory. Generally, this MATLAB object can only be manipulated using standardized functions (called methods), which makes it very hard to introduce any inconsistency into SPM M/EEG data in the first place. Also, using methods simplifies internal book-keeping, which makes it much easier to program functions

operating on the M/EEG object. For example, while the SPM5 format kept a variable which contained the number of trials, the SPM8 object does not have this variable but a method that returns the number of trials by simply counting how many trials of data there are. This makes it easy for a programmer to, for example, remove trials. There is no longer a need to update a variable for the number of trials. Also there is no need to check for the presence of particular fields in the struct in every SPM function. All the methods are guaranteed to work on a valid object. There are many more simplifications along these lines due to the use of "objects" and "methods". In summary, the change in format has resulted in a much more robust, and usable data format.

4.4   Preparing the data after conversion

SPM tries to do its best to extract information automatically from the various data formats. In some cases it can also supplement the converted dataset with information not directly present in the raw data. For instance, SPM can recognize common EEG setups (extended 1020, Biosemi, EGI) based on channel labels and assigns 'EEG' channel type and default electrode locations for these cases. However, there are data types which are either not yet supported in this way or do not contain sufficient information for SPM to make the automatic choices. Also the channel labels do not always correctly describe the actual electrode locations in an experiment. In these cases, further information needs to be supplied by the user. Reading and linking this additional information with the data is the purpose of the `Prepare` interface. This interface is accessed by selecting `Prepare` from the "Other" drop-down menu in the GUI. A menu (easily overlooked) will appear at the top of SPM's interactive window. The same functionality can also be accessed by pressing "Prepare SPM file" in the SPM M/EEG reviewing tool.

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

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

SPM file using the "Load EEG sensors" menu. There are 3 options:

- "Assign default" - assigning default sensor positions. If this is possible, it will be done automatically at conversion but this option can be used to revert to default sensor positions after making some changes.

- "From a `*.mat` file" - this option is for the kind of files that were used in SPM5 and can also be used for any kind of locations without trying to get them into one of the standard formats. SPM will ask for two files. The sensors file should contain an $N \times 3$ matrix, where N is the same as the number of channels whose type is set to "EEG" and the order of the rows matches the order of these channels in the SPM file. The fiducials file should contain a $K \times 3$ matrix, where K (usually 3) is the number of fiducials. You will then be asked to provide labels for these fiducials. They should appear in the same order as the rows in the file.

- "Convert locations file" - this option uses a function from the internal "fileio" toolbox that supports several common formats for EEG channel position specification such as `*.sfp` and BESA's `*.elp`. It can also read Polhemus files from FIL and FCDC. In general Polhemus devices do not have a standard data format so if you are using Polhemus at a different site is is most likely that your Polhemus file will not be recognized by SPM directly. You will need to convert it to another format. An `*.sfp` file is the easiest to create (for instance in Excel). It is just an ASCII file containing a column of channel labels and 3 columns of cartesian coordinates. Check "fileio" website[3] for a complete list of supported formats. The file you are importing can also contain positions of fiducial points or any other named points that do not necessarily correspond to channels. You can also include multiple headshape points with the label "headshape". The important thing is that there are coordinates for each channel that was assigned "EEG" type.

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

If you did not use default 3D positions, after loading the sensor positions you can perform coregistration of your sensors with SPM's template head model. This initial alignment is helpful to verify that the sensor information you supplied were interpreted correctly and should also be done if you would like to generate a 2D sensor template based on your 3D sensor positions (see below). The 2D-coordinates will be used for displaying the data in a topologically meaningful way. This is implemented using the "Coregister" option. For details of how this option works see the 3D source reconstruction chapter 6.

---

[3]fileio: http://fieldtrip.fcdonders.nl/dataformat

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

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

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

A channel template file contains four variables:

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

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

4.5   Integration of SPM and Fieldtrip

The SPM8 distribution includes the latest version of the FieldTrip toolbox[4]. FieldTrip is a MATLAB toolbox for MEG and EEG analysis that is being developed at the F.C. Donders Centre (FCDC) together with collaborating institutes. FieldTrip functions can be used for many kinds of analysis which are not supported in SPM proper. However, FieldTrip does not have extensive graphical user interface and its functionality should be accessed by writing scripts. Full reference documentation for FieldTrip including example scripts is available at the FieldTrip website. The

---

[4]FieldTrip: http://fieldtrip.fcdonders.nl/

SPM distribution also contains some documentation, contained as help comments in FieldTrip functions. These can be found in the directory `external\fieldtrip`.

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

4.6 Reading of data

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

An SPM8 for M/EEG file can be read using the `spm_eeg_load` function. Without any arguments a file requester asks for the name of the file. With a string argument P, `spm_eeg_load(P)` will attempt to read a file with the name `P`. The behavior in this case is identical to the "Just read" option in the GUI. The SPM-format stores the binary data in a `*.dat` file. All header information are stored in a `*.mat` file. This `*.mat` file contains a single struct named `D` which contains several fields. When using `spm_eeg_load`, the struct is transformed into an object, and the data are linked into this object. The linking is done via memory mapping using `file_array` objects. Note that the data should always be read using the routine `spm_eeg_load`. The memory mapped data can be addressed like a matrix (see below) which is convenient for accessing the data in a random access way. However, a word of caution: If you write new values to this matrix, the matrix is not only changed in the object (in memory), but also physically on the hard disk.

You can also load the header struct using a simple MATLAB `load` but this just returns the header struct, without the data linked in, and SPM functions won't know what to do with this struct. In the following, we will describe the methods that one can use on an M/EEG-object. Note that we will not describe the internal format of the data here. This would be helpful only for programmers who want to write new methods for `meeg` object, because when simply analyzing M/EEG or even writing SPM functions there should never be a need to access the internal format. If you think that the existing methods do not provide the functionality you need, please let us know. For interested power users, there is documentation about the internal format within the `meeg` class constructor. *meeg.*

4.6.1 Syntax

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

4.7 Methods for the `meeg` object

`meeg` methods are functions that operate on an `meeg` object, loaded with `spm_eeg_load`. These methods should not be used, if you just want to analyze your data using the GUI or simple scripts. However, if you write your own scripts or high-level functions that need to read or manipulate the object, you need these methods. In the following, we will provide details about most of the methods. The code for all methods can be found in the `@meeg` SPM directory. Most methods provide some minimalist help text. In the following, we will assume that the object variable is called D, i.e. previously loaded by using `D = spm_eeg_load;`.

4.7.1 Constructor `meeg`

The `meeg` method is a constructor. Called without any arguments it will produce a consistent, but empty object. In SPM, the constructor is called when a struct has been loaded into memory by *spm_eeg_load*, and is transformed into an `meeg` object. Importantly, the constructor also checks the consistency of the object.

4.7.2 display

This method will return, in the MATLAB window, some information about the object, e.g., `display(D)`.

4.7.3 Number methods

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

4.7.4 Reading and manipulation of information

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

selectdata

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

chanlabels

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

chantype

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

conditions

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

events

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

- `.type` - string (e.g. "front panel trigger")

- `.value` - number or string, can be empty (e.g. "Trig 1").

- `.time` - in seconds in terms of the original file

- `.duration` - in seconds (optional)

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

fname

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

fnamedat

This method reads or writes the name of the `dat`-file, in which the data are stored.

frequencies

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

fsample

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

history

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

path

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

reject

This method reads or writes the indices of rejected (bad) trials.

timeonset

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

trialonset

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

transformtype

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

type

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

units

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

4.7.5   Reading of information

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

condlist

This method returns a list of condition labels. Multiple entries of labels have been removed.

coor2D

This method returns the 2D-coordinates used for displaying or writing sensor data to voxel-based images.

dtype

This method returns the type under which the data are stored in the `file_array` object.

eogchannels

This method returns which of the channels are EOG channels.

indsample

This method returns the index of the sample which is closest to a specific time point (ms).

indfrequency

For time-frequency datasets this method returns the index of the frequency closest to the given value in Hz.

indchannel

This method returns indices of channels based on their labels.

indtrial

This method returns indices of trials based on condition labels.

meegchannels

This method returns the indices of all channels that are either of the MEG or EEG type.

modality

This method returns the modality of channels (MEG, EEG, etc.).

pickconditions

This method returns the indices of trials of a certain condition. The condition must be supplied by its label (string).

repl

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

time

This method returns the time (ms) of the samples.

sensors

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

fiducials

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

ftraw

This method converts an object to a FieldTrip structure. An additional argument can be supplied to indicate whether the data is memory mapped (1: default) or loaded into memory (0). Note that not all FieldTrip functions can properly handle memory mapped data. In order to avoid corrupting the data, `ftraw` sets a read-only flag on it so in the worst case you might encounter errors in Fieldtrip functions. Please report such errors on the SPM or Fieldtrip mailing list so that we can fix them. `ftraw(0)` is safer in this respect but inefficient in its memory use.

fttimelock

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

4.7.6 Changing the information saved on the disk

save

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

delete

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

4.7.7 struct-like interface

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

4.8 SPM functions

In this section we will describe the high-level SPM functions which are used for preprocessing M/EEG data. These functions are fairly standard and should allow a simple preprocessing of the data (e.g., epoching, filtering, averaging, etc.). Here, we will just describe what each function roughly does and what the input arguments mean. More detailed information about the syntax can be found in the help text of the code. For example, to get detailed help on epoching, type `help spm_eeg_epochs`. The general syntax is the same for all functions. If called from the command-line, and if no input arguments are specified, the function will behave exactly as if you called the function from the GUI by pressing a button or choosing it from the "Other" menu. However, on the command line, or from a script, you can supply input arguments, up to the point when all required input arguments are specified, so that the function will run without any user interaction. In this way, one can write a script that runs without user interaction. See the folder `man\example_scripts` for an example. Input arguments are provided in a struct `S`, whose fields contain the arguments. A typical call, e.g., from a script would be: `D = spm_eeg_epochs(S)`, where S is the input struct, and `D` contains the return argument, the epoched `meeg` object. Note that, with all SPM functions, the object is also always written to hard disk. The filenames of the `mat`- and `dat`-files are generated by prepending a single letter to the old file name. In the example of epoching this would be an 'e'. The idea is that by calling a sequence of functions on a file, the list of first letters of the file name shows (roughly) which preprocessing steps were called to produce this file. Note that another way of calling SPM functions and specifying all input parameters is to use the new batch interface.

4.8.1 Epoching the data: `spm_eeg_epochs`

Epoching cuts out little chunks of continuous data and saves them as "single trials". In M/EEG research, this is a standard data selection procedure to remove long gaps between each single trial. 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 will also be 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. The prepended output letter is *'e'*.

The epoching function can deal with two different ways of specifying trials you want to epoch. The first is to specify explicitly where each trial is located in the measured time-series. The second is to specify trials using the labeled events stored in the file. For most users, the second way is the most convenient, but sometimes, when the stored triggers in the file do not relate to what you want to epoch or there is no event information available in the raw data (e.g. when you have an external stimuli file), you should use the first.

For the first method, you specify a $N \times 2$ so-called `trl` matrix, where each row contains the start and end of a trial (in samples). Optionally, there can be a third column containing the offset of the trigger with respect to the trial. An offset of 0 (default) means that the first sample of the trial corresponds to the trigger. A positive offset indicates that the first sample is later than the trigger, a negative offset indicates that the trial begins before the trigger. In SPM the offset should be the same for all trials. The need to specify a whole column is for interoperability with FieldTrip where trials can have different time axes. In addition you have to specify `conditionlabels` (a single string or a cell array of strings), either one for each trial or one for all trials. You can also enter a `padding` which will add time points before and after each trial to allow the user to later cut out this padding again. This is useful, e.g., for filtering epoched data, where one would otherwise, without padding experience "edge effects".

For the second method, one should define the pre- and post-stimulus interval, and also specify the events (triggers) around which the epochs will be "cut". SPM identifies events by their "event type" and "event value". These are strings or numbers which the software run by the EEG or MEG vendor uses when generating the measurement file. If you don't specify parameters for the epoching function, a GUI will pop up, and present the found triggers with their type and value entries. These can sometimes look strange, but if you want to run a batch or script to do the epoching, you have to first find out what the type and value of your event of interest are. Fortunately, these tend to be the same over scanning sessions, so that you can batch multi-subject epoching using the types and values found in one subject. You also have to come up with a "condition label" for each trial type, which can be anything you choose. This is the label that SPM will use to indicate the trial type of a trial at later processing stages. It is possible to use several types of triggers for defining trials with the same label - in the GUI, just select several events using `Shift` or `Ctrl` key.

For both methods of input you also have to set a (0/1)-flag (no/yes) indicating whether you want to review the information for all trials after selecting them. This allows you to make sure that all your trials are there. You should set the review-flag to 0, if you write a non-interactive script. In the GUI you can review a list of the epochs you defined and exclude some of them by hand (e.g. only take the triggers from the first 5 min of recording). You can also choose to save the trial definitions, for example, for re-use of another epoching of the same data.

4.8.2 Filtering the data: `spm_eeg_filter`

Continuous or epoched data can be filtered, over time, with a low-, high-, stop- or bandpass-filter. SPM uses a Butterworth filter to do this. Note that SPM uses the signal processing toolbox. This means that you have to have this toolbox to filter data in SPM. Phase delays are minimised by using MATLAB 's `filtfilt` function which filters the data twice, forwards and backwards. The prepended output letter is *'f'*.

When you use the function in GUI mode, it will automatically use the Butterworth-filter. You can then choose among the four different types of filtering: "lowpass", "highpass", "bandpass", and "stopband". Depending on your choice, SPM will ask for the cutoff(s) in Hz.

4.8.3   Baseline correction: `spm_eeg_bc`

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

4.8.4   Artefact detection and rejection: `spm_eeg_artefact`

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

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

The GUI for artefact detection is based on the batch interface of SPM. When you click the ARTEFACTS button, window of the SPM8 batch interface will open. Click on "File name" and select the dataset. Double click "How to look for artefacts" and a new branch will appear. It is possible to define several sets of channels to scan and one of the several different methods for artefact detection. For each detection method there are specific configuration parameter (e.g. for thresholding - the threshold value).

4.8.5   Downsampling: `spm_eeg_downsample`

The data can be downsampled to any sampling rate. This is useful if the data was acquired at a higher sampling rate than one needs for making inferences about low-frequency components. For example, resampling from 1000 Hz to 200 Hz would cut down the resulting file size to 20% of the original file size. Note that SPM's downsampling routine uses the MATLAB function `resample`, which is part of the signal processing toolbox. The prepended output letter is *'d'*.

Here, you choose the new sampling rate (Hz) which must be smaller than the old sampling rate.

4.8.6   Rereferencing: `spm_eeg_montage`

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

When you call the function, you will first be asked whether you want to use a GUI or read information from a file to specifying the montage. If you choose GUI, you will see, on the left hand side, the montage-matrix, where each row stands for a new channel. This means the labels in the left column describe the new labels. The old labels are on top, that means, each row contains weights for how the old channels must be weighted to produce new channels in the montage. On the right hand side, you see a graphical representation of the current matrix. The default is the identity matrix, i.e., the montage will not change anything. The concept is very general. For example, if you want to remove channels from the data, just delete the corresponding row from the montage matrix. To re-reference to a particular channel the column for this channel should be -1 for all rows, except the row corresponding to itself which should be 0, whereas the other channels should have 1 in the intersection of their column and row (the diagonal of the matrix) and 0 elsewhere. For average reference the matrix should have $(N-1)/N$ (where N is number of channels) at the diagonal and $-1/N$ elsewhere. In principle, any montage can be represented this way. If you are not sure about how to represent a montage you need, ask an expert or write

to the SPM mailing list. The specification will only need to be done once for your setup and then you can save the montage and use it routinely. After changing the weights of the matrix, you can check the montage by pressing the button in the lower right below the figure.

If you choose to specify the montage by "file", you have to enter the filename of a `mat`-file, which contains a struct with 3 fields: `labelnew` (labels of new channels), `labelorg` (labels of original channels), and the montage-matrix `tra` ("tra" as in transform).

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

4.8.7 Grand mean: `spm_eeg_grandmean`

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

The function will ask you for the name of the output file. Note that in a script, by default, when you don't specify an output filename, SPM will generate a new file with the filename of the first selected file, prepended with a *'g'*.

4.8.8 Merge: `spm_eeg_merge`

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

The function will first check whether there are at least two files, and whether all data are consistent with each other, i.e., have the same number of channels, time points, and sampling rate. The function will also ask what to do with condition labels. The simplest option is to keep them the same. This might be useful for instance when you have several sessions for one subject with the same conditions in all files. In other cases, however, it might be helpful to rename the conditions like "condition A" to something like "condition A, session 1", etc. The simplest way to do it is to append the name of the original file to the condition labels. There is also a possibility to specify more sophisticated 'recoding rules' (see the documentation in the function header). This is mostly useful for writing scripts.

4.8.9 Multimodal fusion: `spm_eeg_fuse`

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

4.8.10 Time-frequency decomposition: `spm_eeg_tf`

The time-frequency decomposition is extendable and can automatically detect and use plugin functions that implement particular spectral estimation algorithms. Algorithms presently implemented include continuous Morlet wavelet transform, Hilbert transorm and multitaper spectral

estimation. The result is written to one or two result files, one containing the instantaneous power and the other, optionally written, the phase estimates (phase estimation is not possible for all algorithms). One can select the channels and frequencies for which power and phase should be estimated. For power, the prepended output letters are *tf_*, for phase *tph_*.

The function can be configured using the SPM batch interface.

4.8.11   Rescaling and baseline correction of time-frequency: `spm_eeg_tf_rescale`

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

4.8.12   Averaging: `spm_eeg_average`

Averaging of single trial data is the crucial step to obtain the evoked response. When averaging single trial data, single trials are averaged within trial type. Power data of single trials (see sec. 4.8.10) can also be averaged by using the function `spm_eeg_average_TF`. The prepended output letter is *'m'*.

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

4.8.13   Contrast of trials: `spm_eeg_weight_epochs`

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

The function will first ask you to "Enter contrasts". This is a matrix where each contrast is given by a row of this matrix. For example, if you compute just one contrast, you have to enter a vector of the same length as the number of trial types in the file. Note that SPM will zero-pad this vector (or matrix) if you specify fewer contrast weights than you have trials. The next question is whether you want to "Weight by num replications". This is important when you use this function on single trials, where, typically, you have a different number of trials for each

trial type. If you then choose to average over multiple trials, this option allows you to choose whether you want to form an average that is weighted by the number of measurements within each trial type. As compared to an average, where you implicitly first form the averages within trial type, and then average with equal weighting.

4.8.14   Copy: `spm_eeg_copy`

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

4.8.15   Sort conditions: `spm_eeg_sort_conditions`

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

4.8.16   Remove bad trials: `spm_eeg_remove_bad_trials`

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

4.9   Displaying data with SPM M/EEG Review

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

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

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

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

- **EEG** displays EEG type data (if any). These are the data associated with "EEG" sensors. The content of this tab is described below, as well as the "MEG" and "OTHER" tabs.

- **MEG** displays MEG type data (if any).

- **OTHER** displays any other type of data (e.g. HEOG, VEOG, etc).

- **info** (active tab by default): displays basic information about the data file. This tab contains three further sub-tabs[5]: "channels", "trials" and "inv" (the latter shows source reconstructions parameters, if any). Some of this info can be changed by the user (e.g. sensor/trial[6] type, label and status, etc) by editing the table. The changes become effective when clicking on "update". They are actually saved in the data file when clicking on "SAVE".

---

[5]Users can also check sensor coregistration when clicking on "sensor positions".

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

- **source** displays source reconstructions (if any). See below (2- source reconstructions visualization).

In addition, the user can call the SPM PREPARE routine [7] or save any modification in the data file using the top-right buttons "Prepare SPM file" and "SAVE".

4.9.1   Data visualization

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

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

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

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

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

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

4.9.2   Source reconstructions visualization

SPM M/EEG REVIEW makes use of sub tabs for any source reconstruction that has been stored in the data file[8]. Since these reconstructions are associated with epoched data, the user can choose the trial he/she wants to display using the list of accessible events (top of the main tab). Each sub tab has a label given by the corresponding source reconstruction comment which is specified by the user when source reconstructing the data (see relevant section in the SPM manual).

---

[7]This is part of the SPM EEG preprocessing tools. It mainly concerns the coregistration of the sensors onto the normalized SPM space. See relevant section in the SPM manual.

[8]This concerns any distributed source reconstruction, i.e. also includes imaging DCM analyses, but not ECD reconstructions (so far).
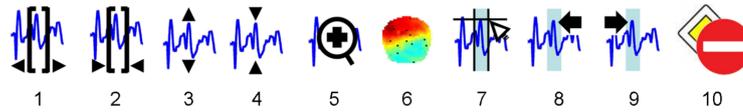
Figure 4.1: *SPM M/EEG* Review *buttons legend 1-2: increase/decrease width of plotted time window, 3-4: increase/decrease global scaling display factor, 5: zoom in, 7: add event, 8-9: scroll backward/forward data from marker to marker, 10: declare event as good/bad*

The bottom-left part of each sub tab displays basic infos about the source reconstruction (date, number of included dipoles, number of temporal modes, etc). The top part of the window displays a rendering of the reconstruction on the cortical surface that has been used. User can scroll through peri-stimulus time by using the temporal slider below the rendered surface. Other sliders allow the user to (i) change the transparency of the surface (left slider) and (ii) threshold the colormap (right sliders). In the center, a butterfly plot of the reconstructed intensity of cortical source activity over peri-stimulus time is displayed. If the data file contains more than one source reconstruction, the bottom-right part of the window displays a bar graph of the model evidences of each source reconstruction. This provides the user with a *visual* Bayesian model comparison tool[9]. SPM M/EEG Review allows quick and easy switching between different models and trials, for a visual comparison of cortical source activities.

4.10 Batching and scripts

Although all functions can be called by the GUI, using only the GUI for preprocessing multi-subject data is a cumbersome and error-prone affair. For this reason, we included functions in SPM8 which should make batching jobs feasible, with only a small amount of Matlab knowledge required. How would one batch a series of preprocessing steps in SPM8? There are two ways of doing this, the new batch system of SPM8, and generating scripts.

4.10.1 The new SPM8 batch system

This method uses the new SPM8 batch system described in chapter VIII. Using this batch system, you can put together a series of processing steps by choosing options from menus, in any order you like. When the full "job" is specified which may be a sequence of several processing step, one executes the job.

4.10.2 Script generation

Another way of batching jobs is by using scripts, written in Matlab . In SPM8, you can generate these scripts automatically. To do this, you first have to analyze one data set using the GUI or batch system. In SPM8, whenever a preprocessing function is called, all the input arguments, once they have been assembled by the GUI, are stored in a "history". This history can then be used to not only see in detail which functions have been used on a data set, but also to generate a script that repeats the same analysis steps. The big difference is that, this time, no more GUI interactions are necessary because the script already has all the input arguments which you gave during the first run. The history of an `meeg` object can be accessed by `D.history`.

To generate a script from the history of an SPM8 MEEG file, open the file in the M/EEG Review facility and select the `info` tab: a `history` tab is then available that will display all the history of the file. Clicking the Save as script button will ask for the filename of the Matlab

---

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

script to save and the list of processing steps to save (default is all but it is possible to select only a subset of them). This will generate a script, which, when run, repeats the analysis. The script can also be obtained by directly calling the function `spm_eeg_history`.

Of course, this script can not only be used to repeat an analysis, but the script can also be seen as a template that can be re-used for other analyses. One needs minimal MATLAB knowledge for these changes. For example, you can replace the filenames to preprocess a different subject. Or you can change parameters and then re-run the analysis. We have prepared an example, using the same example data set, as in the previous subsection to demonstrate this (see the file `man\example_scripts\history_subject1.m`). Note that these scripts can currently be used to do things that one couldn't do with the batch system. For example, if you want to exclude a channel from the analysis, there is no way to do this via the batch system. In the GUI, you would have to call display and switch the channel to 'bad'. With a script, you simply add a line like `D=badchannels(D, 23, 1)`, which flags channel 23 as bad (see also our example script after the filtering step). In summary, the idea is to preprocess a file through the GUI or batch system, then use the history-function to generate a template, and finally adapt this template to modify the analysis in some way. To run the example script on your computer, you need the data set that you can download from the SPM webpage ([10]).

---

[10]http://www.fil.ion.ucl.ac.uk/spm/data/eeg_mmn/

Chapter 5

Analysis in sensor space

Second-level analyses for M/EEG are used to make inferences about the population from which subjects are drawn. This chapter describes how to perform second-level analyses of EEG/MEG or time-frequency (TF) data. This simply requires transforming data from `filename.mat` and `filename.dat` format to image files (NIfTI format). Once the data are in image format second-level analyses for M/EEG are procedurally identical to 2nd level analyses for fMRI. We therefore refer the reader to the fMRI section for further details of this last step. Also the "Multimodal face-evoked responses" tutorial chapter 19 contains detailed examples of sensor-level analysis.
In the drop down "Other" menu, select the function `Convert to images`. This will ask you to select the `filename.mat` of the data you would like to convert to images.

5.0.3 Data in time domain

If when calling `Convert to images` you choose an epoched or averaged dataset, you will then be prompted to enter the output dimensions of the interpolated scalp image that will be produced. Typically, we suggest using 64. You will then be asked whether you want to interpolate or remove bad channels from your images. If you choose interpolate then the image will interpolate missing channels. This is the preferred option. If you choose to remove bad channels there will be 'holes' in the resulting images. A directory will be created with the same name as the input dataset. In this directory there will be a subdirectory for each trial type. These directories will contain 3D image files where the dimensions are space (X,Y) and time. In the case of averaged dataset a single image is put in each directory. In the case of epoched dataset there will be an image for each trial.

Averaging over time

If you know in advance the time window you are interested in (for instance if you are interested in a well characterized ERP/ERF peak) you can average over this time window to create a 2D image with just the spatial dimensions. This is done by pressing the `Specify 1-st level` button (the name is historical as this operation is remotely related to 1-st level analysis of fMRI data). You will be asked to specify the time window (in ms) and select one or more space-time images and the output directory.

Masking

When you set up your statistical analysis, it might be useful to use an explicit mask for two reasons. Firstly, if you smooth the images, the areas outside the scalp will also have some values and they will be included in the analysis by default if you don't use a mask. Secondly, you might want to limit your time window. This can also be done by using small volume correction afterwards, but using a mask might be more convenient if you have a fixed time window of interest. Such a mask can be created by selecting `Mask images` from "Other" dropdown menu. You will be asked to provide one unsmoothed image to be used as a template for the mask. This can be

any of the images you exported. Then you will be asked to specify the time window of interest and the name for the output mask file.

5.0.4   Time-frequency data

If you supply a time-frequency dataset as the input, you will first be asked to reduce your data from a 4D data (space(X,Y), time, frequency) to either a 3D image (space(X,Y), time) or a 2D time-frequency image (time, frequency). The prompt asks you over which dimension you wish to average your data.

If you select "channels" you will be asked to select the channels you wish to average over. Next you will be prompted for a "region number". This is just a tag that allows you to average over different channels to create different analyses for more than one region. The image will be created in a new directory with the name `<region number>ROI_TF_trialtype_<condition number>`. The image that will be created will be a 2D image where the dimensions are time and frequency.

If you select "frequency" you will be asked to specify the frequency range over which you wish to average. The power is then averaged over the specified frequency band to produce channel waveforms. These waveforms are saved in a new time domain M/EEG dataset whose name starts with `F<frequency range>`. This dataset can be reviewed and further processed in the same way as ordinary datasets with waveforms (of course source reconstruction or DCM are will not work for it). Once this dataset is generated it is automatically exported to images the same way as data in time domain (see above).

5.0.5   Smoothing

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

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

Chapter 6

3D source reconstruction: Imaging approach

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

6.1   Introduction

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

Given epoched and preprocessed data (see chapter 4), the evoked and/or induced activity for each dipolar source can be estimated, for a single time-sample or a wider peristimulus time window.

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

Contrary to PET/fMRI data reconstruction, EEG/MEG source reconstruction is a non trivial operation. Often compared to estimating a body shape from its shadow, inferring brain activity from scalp data is mathematically ill-posed and requires prior information such as anatomical, functional or mathematical constraints to isolate a unique and most probable solution [3].

Distributed linear models have been around for more than a decade now [11] and the proposed pipeline in SPM for an imaging solution is classical and very similar to common approaches in the field. However, at least two aspects are quite original and should be emphasized here:

- Based on an empirical Bayesian formalism, the inversion is meant to be generic in the sense it can incorporate and estimate the relevance of multiple constraints of varied nature; data-driven relevance estimation being made possible through Bayesian model comparison [25, 64, 50, 21].

- The subject's specific anatomy is incorporated in the generative model of the data, in a fashion that eschews individual cortical surface extraction. The individual cortical mesh is obtained automatically from a canonical mesh in MNI space, providing a simple and efficient way of reporting results in stereotactic coordinates.

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

1. Source space modeling,

2. Data co-registration,

3. Forward computation,

4. Inverse reconstruction.

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

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

6.2   Getting started

Everything which is described hereafter is accessible from the SPM user-interface by choosing the "EEG" application, `3D Source Reconstruction` button. When you press this button a new window will appear with a GUI that will guide you through the necessary steps to obtain an imaging reconstruction of your data. At each step, the buttons that are not yet relevant for this step will be disabled. When you open the window the only two buttons you can press are `Load` which enables you to load a pre-processed SPM MEEG dataset and the `Group inversion` button that will be described below. You can load a dataset which is either epoched with single trials for different conditions, averaged with one event related potential (ERP) per condition, or grand-averaged. An important pre-condition for loading a dataset is that it should contain sensors and fiducials. This will be checked when you load a file and loading will fail in case of a problem. You should make sure that for each modality present in the dataset as indicated by channel types (either EEG or MEG) there is a sensor description. If, for instance, you have an MEG dataset with some EEG channels that you don't actually want to use for source reconstruction, change their type to "*LFP*" or "*Other*" before trying to load the dataset (the difference is that *LFP* channels will stil be filtered and available for artefact detection whereas *Other* channels won't). MEG datasets converted by SPM from their raw formats will always contain sensor and fiducial descriptions. In the case of EEG for some supported channel setups (such as extended 10-20 or BioSemi) SPM will provide default channel locations and fiducials that you can use for your reconstruction. Sensor and fiducial descriptions can be modified using the `Prepare` interface and in this interface you can also verify that these descriptions are sensible by performing a coregistration (see chapter 4 and also below for more details about coregistration).

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

6.3   Source space modeling

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

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

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

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

6.4 Coregistration

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

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

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

2. Surface matching (between some headshape in MEG/EEG space and some sMRI derived scalp tesselation).
   For EEG, the sensor locations can be used instead of the headshape. For MEG, the headshape is first coregistrated into sMRI space; the inverse transformation is then applied to the head model and the mesh.
   Surface matching is performed using an Iterative Closest Point algorithm (ICP). The ICP algorithm [4] is an iterative alignment algorithm that works in three phases:

   - Establish correspondence between pairs of features in the two structures that are to be aligned based on proximity;

   - Estimate the rigid transformation that best maps the first member of the pair onto the second;

   - Apply that transformation to all features in the first structure. These three steps are then reapplied until convergence is concluded. Although simple, the algorithm works quite effectively when given a good initial estimate.

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

- `select` - locations of some points such as the commonly used nasion and preauricular points and also CTF recommended fiducials for MEG (as used at the FIL) are hard-coded in SPM. If your fiducial corresponds to one of these points you can select this option and then select the correct point from a list.

- `type` - here you can enter the MNI coordinates for your fiducial (1×3 vector). If your fiducial is not on SPM's hard-coded list, it is advised to carefully find the right point on either the template image or on your subject's own image normalized to the template. You can do it by just opening the image using SPM's Display/images functionality. You can then record the MNI coordinates and use them in all coregistrations you need to do using the "type" option.

- `click` - here you will be presented with a structural image where you can click on the right point. This option is good for "quick and dirty" coregistration or to try out different options.

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

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

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

6.5　Forward computation (*forward*)

This refers to computing for each of the dipoles on the cortical mesh the effect it would have on the sensors. The result is a $N \times M$ matrix where N is the number of sensors and M is the number of mesh vertices (that you chose from several options at a previous step). This matrix can be quite big and it is, therefore, not stored in the header, but in a separate `*.mat` file which has `SPMgainmatrix` in its name and is written in the same directory as the dataset. Each column in this matrix is a so called "lead field" corresponding to one mesh vertex.

The lead fields are computed using the "forwinv" toolbox[1] developed by Robert Oostenveld, which SPM shares with FieldTrip. This computation is based on Maxwell's equations and makes assumptions about the physical properties of the head. There are different ways to specify these assumptions which are known as "forward models".

The "forwinv" toolbox can support different kinds of forward models. When you press `Forward Model` button (which should be enabled after successful coregistration), you will have a choice of several head models depending on the modality of your dataset. In SPM8 we recommend useing a single shell model for MEG and "EEG BEM" for EEG. You can also try other options and compare them using model evidence (see below). The first time you use the EEG BEM option with a new structural image (and also the first time you use the `Template` option) a lengthy computation will take place that prepares the BEM model based on the head meshes. The BEM will then be saved in a quite large `*.mat` file with ending `_EEG_BEM.mat` in the same directory with the structural image ("canonical" subdirectory of SPM for the template). When the head model is ready, it will be displayed in the graphics window with the cortical mesh and sensor locations you should verify for the final time that everything fits well together.

---

[1]forwinv: http://fieldtrip.fcdonders.nl/development/forwinv

The actual lead field matrix will be computed at the beginning of the next step and saved. This is a time-consuming step and it takes longer for high-resolution meshes. The lead field file will be used for all subsequent inversions if you do not change the coregistration and the forward model.

6.6 Inverse reconstruction

To get started press the `Invert` button. The first choice you will see is between `Imaging`, `VB-ECD` and `Beamforming`. For reconstruction based on an empirical Bayesian approach to localize either the evoked response, the evoked power or the induced power, as measured by EEG or MEG press the `Imaging` button. The other options are explained in greater detail elsewhere.

If you have trials belonging to more than one condition in your dataset then the next choice you will have is whether to invert all the conditions together or to choose a subset. It is recommended to invert the conditions together if you are planning to later do a statistical comparison between them. If you have only one condition, or after choosing the conditions, you will get a choice between "Standard" and "Custom" inversion. If you choose "Standard" inversion, SPM will start the computation with default settings. These correspond to the multiple sparse priors (MSP) algorithm [19] which is then applied to the whole input data segment.

If you want to fine-tune the parameters of the inversion, choose the "Custom" option. You will then have the possibility to choose between several types of inversion differing by their hyperprior models (IID - equivalent to classical minimum norm, COH - smoothness prior similar to methods such as LORETA) or the MSP method .

You can then choose the time window that will be available for inversion. Based on our experience, it is recommended to limit the time window to the activity of interest in cases when the amplitude of this activity is low compared to activity at other times. The reason is that if the irrelevant high-amplitude activity is included, the source reconstruction scheme will focus on reducing the error for reconstructing this activity and might ignore the activity of interest. In other cases, when the peak of interest is the strongest peak or is comparable to other peaks in its amplitude, it might be better not to limit the time window to let the algorithm model all the brain sources generating the response and then to focus on the sources of interest using the appropriate contrast (see below). There is also an option to apply a hanning taper to the channel time series in order to downweight the possible baseline noise at the beginning and end of the trial. There is also an option to pre-filter the data. Finally, you can restrict solutions to particular brain areas by loading a `*.mat` file with a $K \times 3$ matrix containing MNI coordinates of the areas of interest. This option may initially seem strange, as it may seem to overly bias the source reconstructions returned. However, in the Bayesian inversion framework you can compare different inversions of the same data using Bayesian model comparison. By limiting the solutions to particular brain areas you greatly simplify your model and if that simplification really captures the sources generating the response, then the restricted model will have much higher model evidence than the unrestricted one. If, however, the sources you suggested cannot account for the data, the restriction will result in a worse model fit and depending on how much worse it is, the unrestricted model might be better in the comparison. So using this option with subsequent model comparison is a way, for instance, to integrate prior knowledge from the literature or from fMRI/PET/DTI into your inversion. It also allows for comparison of alternative prior models.

Note that for model comparison to be valid all the settings that affect the input data, like the time window, conditions used and filtering should be identical.

SPM8 imaging source reconstruction also supports multi-modal datasets. These are datasets that have both EEG and MEG data from a simultaneous recording. Datasets from the "Neuromag" MEG system which has two kinds of MEG sensors are also treated as multimodal. If your dataset is multimodal a dialogue box will appear asking to select the modalities for source reconstruction from a list. If you select more than one modality, multimodal fusion will be performed. This option based on the paper by Henson et al. [32] uses a heuristic to rescale the data from different modalities so that they can be used together.

Once the inversion is completed you will see the time course of the region with maximal activity in the top plot of the graphics window. The bottom plot will show the maximal intensity projection (MIP) at the time of the maximal activation. You will also see the log-evidence value

that can be used for model comparison, as explained above. Note that not all the output of the inversion is displayed. The full output consists of time courses for all the sources and conditions for the entire time window. You can view more of the results using the controls in the bottom right corner of the 3D GUI. These allow focusing on a particular time, brain area and condition. One can also display a movie of the evolution of neuronal activity.

6.7 Summarizing the results of inverse reconstruction as an image

SPM offers the possibility of writing the results as 3D NIfTI images, so that you can then proceed with GLM-based statistical analysis using Random Field theory. This is similar to the 2nd level analysis in fMRI for making inferences about region and trial-specific effects (at the between subject level).

This entails summarizing the trial- and subject-specific responses with a single 3-D image in source space. Critically this involves prompting for a time-frequency contrast window to create each contrast image. This is a flexible and generic way of specifying the data feature you want to make an inference about (e.g., gamma activity around 300 ms or average response between 80 and 120 ms). This kind of contrast is specified by pressing the `Window` button. You will then be asked about the time window of interest (in ms, peri-stimulus time). It is possible to specify one or more time segments (separated by a semicolon). To specify a single time point repeat the same value twice. The next question is about the frequency band. If you just want to average the source time course leave that at the default, zero. In this case the window will be weighted by a Gaussian. In the case of a single time point this will be a Gaussian with 8 ms full width half maximum (FWHM). If you specify a particular frequency or a frequency band, then a series of Morlet wavelet projectors will be generated summarizing the energy in the time window and band of interest.

There is a difference between specifying a frequency band of interest as zero, as opposed to specifying a wide band that covers the whole frequency range of your data. In the former case the time course of each dipole will be averaged, weighted by a gaussian. Therefore, if within your time window this time course changes polarity, the activity can average out and in an ideal case even a strong response can produce a value of zero. In the latter case the power is integrated over the whole spectrum ignoring phase, and this would be equivalent to computing the sum of squared amplitudes in the time domain.

Finally, if the data file is epoched rather than averaged, you will have a choice between "evoked", "induced" and "trials". If you have multiple trials for certain conditions, the projectors generated at the previous step can either be applied to each trial and the results averaged (induced) or applied to the averaged trials (evoked). Thus it is possible to perform localization of induced activity that has no phase-locking to the stimulus. It is also possible to focus on frequency content of the ERP using the "evoked" option. Clearly the results will not be the same. The projectors you specified (bottom plot) and the resulting MIP (top plot) will be displayed when the operation is completed. "trials" option makes it possible to export an image per trial which might be useful fot doing within-subject statistics.

The values of the exported images are normalized to reduce between-subject variance. Therefore, for best results it is recommended to export images for all the time windows and conditions that will be included in the same statistical analysis in one step. Note that the images exported from the source reconstruction are a little peculiar because of smoothing from a 2D cortical sheet into 3D volume. SPM statistical machinery has been optimized to deal with these peculiarities and get sensible results. This presently only works for T-contrasts but not for F. Also if you try to analyze the images with older versions of SPM or with a different software package you might get different (less focal) results.

6.8 Rendering interface

By pressing the `Render` button you can open a new GUI window which will show you a rendering of the inversion results on the brain surface. You can rotate the brain, focus on different time

points, run a movie and compare the predicted and observed scalp topographies and time series. A useful option is "virtual electrode" which allows you to extract the time course from any point on the mesh and the MIP at the time of maximal activation at this point. Just press the button and click anywhere in the brain.

An additional tool for reviewing the results is available in the SPM M/EEG Review function.

6.9   Group inversion

A problem encountered with MSP inversion is that sometimes it is "too good", producing solutions that were so focal in each subject that the spatial overlap between the activated areas across subjects was not sufficient to yield a significant result in a between-subjects contrast. This could be improved by smoothing, but smoothing compromises the spatial resolution and thus subverts the main advantage of using an inversion method that can produce focal solutions.

To circumvent this problem we proposed a modification of the MSP method [48] that effectively restricts the activated sources to be the same in all subjects with only the degree of activation allowed to vary. We showed that this modification makes it possible to obtain significance levels close to those of non-focal methods such as minimum norm while preserving accurate spatial localization.

The group inversion can yield much better results than individual inversions because it introduces an additional constraint for the ill-posed inverse problem, namely that the responses in all subjects should be explained by the same set of sources. Thus it should be your method of choice when analyzing an entire study with subsequent GLM analysis of the images.

Group inversion works very similarly to what was described above. You can start it by pressing the "Group inversion" button right after opening the 3D GUI. You will be asked to specify a list of M/EEG data sets to invert together. Then the routine will ask you to perform coregistration for each of the files and specify all the inversion parameters in advance. It is also possible to specify the contrast parameters in advance. Then the inversion will proceed by computing the inverse solution for all the files and will write out the output images. The results for each subject will also be saved in the header of the corresponding input file. It is possible to load this file into the 3D GUI after the inversion and explore the results as described above.

6.10   Batching source reconstruction

There is a possibility to run imaging source reconstruction using the SPM8 batch tool. It can be accessed by pressing the "Batch" button in the main SPM window and then going to "M/EEG source reconstruction" in the "SPM" under "M/EEG". There are three separate tools there: for building head models, computing the inverse solution and computing contrasts and generating images. This makes it possible for instance to generate images for several different contrasts from the same inversion. All the three tools support multiple datasets as inputs. In the case of the inversion tool group inversion will be done for multiple datasets.

6.11   Appendix: Data structure

The Matlab object describing a given EEG/MEG dataset in SPM is denoted as *D*. Within that structure, each new inverse analysis will be described by a new cell of sub-structure field *D.inv* and will be made of the following fields:

- `method`: character string indicating the method, either "ECD" or "Imaging" in present case;

- `mesh`: sub-structure with relevant variables and filenames for source space and head modeling;

- `datareg`: sub-structure with relevant variables and filenames for EEG/MEG data registration into MRI space;

- `forward`: sub-structure with relevant variables and filenames for forward computation;

- `inverse`: sub-structure with relevant variable, filenames as well as results files;

- `comment`: character string provided by the user to characterize the present analysis;

- `date`: date of the last modification made to this analysis.

- `gainmat`: name of the gain matrix file.

Chapter 7

Localization of Equivalent Current Dipoles

This chapter describes source reconstruction based on "Variational Bayes Equivalent Current Dipoles" (VB-ECDs). For more details about the implementation, please refer to the help and comments in the routines themselves, as well as the original paper by [42].

7.1 Introduction

3D imaging (or distributed) reconstruction methods consider all possible source location simultaneously, allowing for large and widely spread clusters of activity. This is to be contrasted with "Equivalent Current Dipole (ECD) approaches which rely on two different hypotheses:

- only a few (say less than 5) sources are active simultaneously, and

- those sources are very focal.

This leads to the ECD model where the observed scalp potential will be explained by a handful of discrete current sources, i.e. dipoles, located inside the brain volume.

In contrast to the 3D imaging reconstruction, the number of ECDs considered in the model, i.e. the number of "active locations, should be defined a priori. This is a crucial step, as the number of sources considered defines the ECD model. This choice should be based on empirical knowledge of the brain activity observed or any other source of information (for example by looking at the scalp potential distribution). In general, each dipole is described by 6 parameters: 3 for its location, 2 for its orientation and 1 for its amplitude. Once the number of ECDs is fixed, a non-linear optimisation algorithm is used to adjust the dipoles parameters (6 times the number of dipoles) to the observed potential.

Classical ECD approaches use a simple best fitting optimisation using least square error criteria. This leads to relatively simple algorithms but presents a few drawbacks:

- constraints on the dipoles are difficult to include in the framework;

- the noise cannot be properly taken into account, as its variance should be estimated alongside the dipole parameters;

- it is difficult to define confidence intervals on the estimated parameters, which could lead to over-confident interpretation of the results;

- models with different numbers of dipoles cannot be compared except through their goodness-of-fit, which can be misleading.

As adding dipoles to a model will necessarily improve the overall goodness of fit, one could erroneously be tempted to use as many ECDs as possible and to perfectly fit the observed signal. Through using Bayesian techniques, however, it is possible to circumvent all of the above limitations of classical approaches.

Briefly, a probabilistic generative model is built providing a likelihood model for the data[1]. The model is completed by a set of priors on the various parameters, leading to a Bayesian model, allowing the inclusion of user-specified prior constraints.

A "variational Bayes (VB) scheme is then employed to estimate the posterior distribution of the parameters through an iterative procedure. The confidence interval of the estimated parameters is therefore directly available through the estimated posterior variance of the parameters. Critically, in a Bayesian context, different models can be compared using their evidence or marginal likelihood. This model comparison is superior to classical goodness-of-fit measures, because it takes into account the complexity of the models (e.g., the number of dipoles) and, implicitly, uncertainty about the model parameters. VB-ECD can therefore provide an objective and accurate answer to the question: Would this data set be better modelled by 2 or 3 ECDs?

7.2 Procedure in SPM8

This section aims at describing how to use the VB-ECD approach in SPM8.

7.2.1 Head and forward model

The engine calculating the projection of the dipolar sources on the scalp electrode comes from Fieldtrip and is the same for the 3D imaging or DCM. The head model should thus be prepared the same way, as described in the chapter 6. For the same data set, differences between the VB-ECD and imaging reconstructions would therefore be due to the reconstruction approach only.

7.2.2 VB-ECD reconstruction

To get started, after loading and preparing the head model, press the 'Invert' button[2]. The first choice you will see is between 'Imaging', 'VB-ECD' and 'DCM'. The 'Imaging' reconstruction corresponds to the imaging solution, as described in chapter 6, and 'DCM' is described in chapter 8. Then you are invited to fill in information about the ECD model and click on buttons in the following order:

1. indicate the time bin or time window for the reconstruction, within the epoch length. Note that the data will be averaged over the selected time window! VB-ECD will thus always be calculated for a single time bin.

2. enter the trial type(s) to be reconstructed. Each trial type will be reconstructed separately.

3. add a single (i.e. individual) dipole or a pair of symmetric dipoles to the model. Each "element (single or pair) is added individually to the model.

4. use "Informative or 'Non-informative location priors. "Non-informative means flat priors over the brain volume. With "Informative, you can enter the a priori location of the source[3].

5. use "Informative or 'Non-informative moment priors. "Non-informative means flat priors over all possible directions and amplitude. With "Informative, you can enter the a priori moment of the source[4].

6. go back to step 3 and add some more dipole(s) to the model, or stop adding dipoles.

7. specify the number of iterations. These are repetitions of the fitting procedure with different initial conditions. Since there are multiple local maxima in the objective function, multiple iterations are necessary to get good results especially when non-informative location priors are chosen.

---

[1]This includes an independent and identically distributed (IID) Normal distribution for the errors, but other distributions could be specified.

[2]The GUI for VB-ECD can also be launched directly from Matlab command line with the instruction: $D = spm\_eeg\_inv\_vbecd\_gui$.

[3]For a pair of dipoles, only the right dipole coordinates are required.

[4]For a pair of dipoles, only the right dipole moment is required.

The routine then proceeds with the VB optimization scheme to estimate the model parameters. There is graphical display of the intermediate results. When the best solution is selected the model evidence will be shown at the top of the SPM Graphics window. This number can be used to compare solutions with different priors.

Results are finally saved into the data structure D in the field *.inv{D.val}.inverse* and displayed in the graphic window.

7.2.3 Result display

The latest VB-ECD results can be displayed again through the function $D = spm\_eeg\_inv\_vbecd\_disp$. If a specific reconstruction should be displayed, then use: *spm_eeg_inv_vbecd_disp('Init',D, ind)*. In the GUI you can use the `'dip'` button (located under the 'Invert' button) to display the dipole locations.

In the upper part, the 3 main figures display the 3 orthogonal views of the brain with the dipole location and orientation superimposed. The location confidence interval is described by the dotted ellipse around the dipole location on the 3 views. It is not possible to click through the image, as the display is automatically centred on the dipole displayed. It is possible though to zoom into the image, using the right-click context menu.

The lower left table displays the current dipole location, orientation (Cartesian or polar coordinates) and amplitude in various formats.

The lower right table allows for the selection of trial types and dipoles. Display of multiple trial types and multiple dipoles is also possible. The display will center itself on the average location of the dipoles.

Chapter 8

Dynamic Causal Modelling for M/EEG

8.1 Introduction

Dynamic Causal Modelling (DCM) is based on an idea initially developed for fMRI data: The measured data are explained by a network model consisting of a few sources, which are interacting dynamically. This network model is inverted using a Bayesian approach, and one can make inferences about connections between sources, or the modulation of connections by task.

For M/EEG data, DCM is a powerful technique for inferring about parameters that one doesn't observe with M/EEG directly. Instead of asking 'How does the strength of the source in left superior temporal gyrus (STG) change between condition A and B?', one can ask questions like 'How does the backward connection from this left STG source to left primary auditory cortex change between condition A and B?'. In other words, one isn't limited to questions about source strength as estimated using a source reconstruction approach, but can test hypotheses about what is happening between sources, in a network.

As M/EEG data is highly resolved in time, as compared to fMRI, the inferences are about more neurobiologically plausible parameters. These relate more directly to the causes of the underlying neuronal dynamics.

The key DCM for M/EEG methods paper appeared in 2006, and the first DCM studies about mismatch negativity came out in 2007/2008. At its heart DCM for M/EEG is a source reconstruction technique, and for the spatial domain we use exactly the same leadfields as other approaches. However, what makes DCM unique, is that is combines the spatial forward model with a biologically informed temporal forward model, describing e.g. the connectivity between sources. This critical ingredient not only makes the source reconstruction more robust by implicitly constraining the spatial parameters, but also allows one to infer about connectivity.

Our methods group is continuing to work on further improvements and extensions to DCM. In the following, we will describe the usage of DCM for evoked responses (both MEG and EEG), DCM for induced responses (i.e., based on power data in the time-frequency domain), and DCM for local field potentials (measured as steady-state responses). All three DCMs share the same interface, as many of the parameters that need to be specified are the same for all three approaches. Therefore, we will first describe DCM for evoked responses, and then point out where the differences to the other two DCMs lie.

This manual provides only a procedural guide for the practical use of DCM for M/EEG. If you want to read more about the scientific background, the algorithms used, or how one would typically use DCM in applications, we recommend the following reading. The two key methods contributions can be found in [13] and [43]. Two other contributions using the model for testing interesting hypotheses about neuronal dynamics are described in [44] and [14]. At the time of writing, there were also three application papers published which demonstrate what kind of hypotheses can be tested with DCM [28, 27, 26]. Another good source of background information

is the recent SPM book [16], where Parts 6 and 7 cover not only DCM for M/EEG but also related research from our group. The DCMs for induced responses and steady-state responses are covered in [10, 9] and [54, 51, 52]. Also note that there is a DCM example file, which we put onto the webpage http://www.fil.ion.ucl.ac.uk/spm/data/eeg_mmn/. After downloading DCMexample.mat, you can load (see below) this file using the DCM GUI, and have a look at the various options, or change some, after reading the description below.

8.2   Overview

In summary, the goal of DCM is to explain measured data (such as evoked responses) as the output of an interacting network consisting of a several areas, some of which receive input (i.e., the stimulus). The differences between evoked responses, measured under different conditions, are modelled as a modulation of selected DCM parameters, e.g. cortico-cortical connections [13]. This interpretation of the evoked response makes hypotheses about connectivity directly testable. For example, one can ask, whether the difference between two evoked responses can be explained by top-down modulation of early areas [28]. Importantly, because model inversion is implemented using a Bayesian approach, one can also compute Bayesian model evidences. These can be used to compare alternative, equally plausible, models and decide which is the best [45].

DCM for evoked responses takes the spatial forward model into account. This makes DCM a spatiotemporal model of the full data set (over channels and peri-stimulus time). Alternatively, one can describe DCM also as a spatiotemporal source reconstruction algorithm which uses additional temporal constraints given by neural mass dynamics and long-range effective connectivity. This is achieved by parameterising the lead-field, i.e., the spatial projection of source activity to the sensors. In the current version, this can be done using two different approaches. The first assumes that the leadfield of each source is modelled by a single equivalent current dipole (ECD) [43]. The second approach posits that each source can be presented as a 'patch' of dipoles on the grey matter sheet [12]. This spatial model is complemented by a model of the temporal dynamics of each source. Importantly, these dynamics not only describe how the intrinsic source dynamics evolve over time, but also how a source reacts to external input, coming either from subcortical areas (stimulus), or from other cortical sources.

The GUI allows one to enter all the information necessary for specifying a spatiotemporal model for a given data set. If you want to fit multiple models, we recommend using a batch script. An example of such a script (*DCM_ERP_example*), which can be adapted to your own data, can be found in the *man/example_scripts/* folder of the distribution. You can run this script on example data provided by via the SPM webpage (http://www.fil.ion.ucl.ac.uk/spm/data/eeg_mmn/). However, you first have to preprocess these data to produce an evoked response by going through the preprocessing tutorial (chapter 18) or by running the `history_subject1.m` script in the `example_scripts` folder.

8.3   Calling DCM for ERP/ERF

After calling *spm eeg*, you see SPM's graphical user interface, the top-left window. The button for calling the DCM-GUI is found in the second partition from the top, on the right hand side. When pressing the button, the GUI pops up. The GUI is partitioned into five parts, going from the top to the bottom. The first part is about loading and saving existing DCMs, and selecting the type of model. The second part is about selecting data, the third is for specification of the spatial forward model, the fourth is for specifying connectivity, and the last row of buttons allows you to estimate parameters and view results.

You have to select the data first and specify the model in a fixed order (data selection > spatial model > connectivity model). This order is necessary, because there are dependencies among the three parts that would be hard to resolve if the input could be entered in any order. At any time, you can switch back and forth from one part to the next. Also, within each part, you can specify information in any order you like.

8.4 load, save, select model type

At the top of the GUI, you can load an existing DCM or save the one you are currently working on. In general, you can *save* and *load* during model specification at any time. You can also switch between different DCM analyses (the left menu). The default is 'ERP' which is DCM for evoked responses described here. Currently, the other two types are steady-state responses (SSR) and induced responses (IND); both are also described in this manual. The menu on the right-hand side lets you choose the neuronal model. Currently, there are four model types. The first is 'ERP', which is the standard model described in most of our papers, e.g. [13]. The second is 'SEP', which uses a variant of this model, however, the dynamics tend to be faster [49]. The third is 'NMM', which is a nonlinear neural mass model based on a first-order approximation, and the fourth is 'MFM', which is also nonlinear and is based on a second-order approximation. The latter two options haven't been described in the literature yet, but are under review.

8.5 Data and design

In this part, you select the data and model between-trial effects. The data can be either event-related potentials or fields. These data must be in the SPM-format. On the right-hand side you can enter trial indices of the evoked responses in this SPM-file. For example, if you want to model the second and third evoked response contained within an SPM-file, specify indices 2 and 3. The indices correspond to the order specified by the *condlist* method (see 4). If the two evoked responses, for some reason, are in different files, you have to merge these files first. You can do this with the SPM preprocessing function *merge* (*spm_eeg_merge*), see 4. You can also choose how you want to model the experimental effects (i.e. the differences between conditions). For example, if trial 1 is the standard and trial 2 is the deviant response in an oddball paradigm, you can use the standard as the baseline and model the differences in the connections that are necessary to fit the deviant. To do that type 0 1 in the text box below trial indices. Alternatively, if you type −1 1 then the baseline will be the average of the two conditions and the same factor will be subtracted from the baseline connection values to model the standard and added to model the deviant. The latter option is perhaps not optimal for an oddball paradigm but might be suitable for other paradigms where there is no clear 'baseline condition'. When you want to model three or more evoked responses, you can model the modulations of a connection strength of the second and third evoked responses as two separate experimental effects relative to the first evoked response. However, you can also choose to couple the connection strength of the first evoked response with the two gains by imposing a linear relationship on how this connection changes over trials. Then you can specify a single effect (e.g. −1 0 1). This can be useful when one wants to add constraints on how connections (or other DCM parameters) change over trials. A compelling example of this can be found in [26]. For each experimental effect you specify, you will be able to select the connections in the model that are affected by it (see below).

Press the button 'data file' to load the M/EEG dataset. Under 'time window (ms)' you have to enter the peri-stimulus times which you want to model, e.g. 1 to 200 ms.

You can choose whether you want to model the mean or drifts of the data at sensor level. Select 1 for 'detrend' to just model the mean. Otherwise select the number of discrete cosine transform terms you want to use to model low-frequency drifts (> 1). In DCM, we use a projection of the data to a subspace to reduce the amount of data. The type of spatial projection is described in [14]. You can select the number of modes you wish to keep. The default is 8.

You can also choose to window your data, along peri-stimulus time, with a hanning window (radio button). This windowing will reduce the influence of the beginning and end of the time-series.

If you are happy with your data selection, the projection and the detrending terms, you can click on the > (forward) button, which will bring you to the next stage *electromagnetic model*. From this part, you can press the red < button to get back to the data and design part.

8.6 Electromagnetic model

With the present version of DCM, you have three options for how to spatially model your evoked responses. Either you use a single equivalent current dipole (ECD) for each source, or you use a patch on the cortical surface (IMG), or you don't use a spatial model at all (local field potentials (LFP)). In all three cases, you have to enter the source names (one name in one row). For ECD and IMG, you have to specify the prior source locations (in mm in MNI coordinates). Note that by default DCM uses uninformative priors on dipole orientations, but tight priors on locations. This is because tight priors on locations ensure that the posterior location will not deviate to much from its prior location. This means each dipole stays in its designated area and retains its meaning. The prior location for each dipole can be found either by using available anatomical knowledge or by relying on source reconstructions of comparable studies. Also note that the prior location doesn't need to be overly exact, because the spatial resolution of M/EEG is on a scale of several millimeters. You can also load the prior locations from a file ('load'). You can visualize the locations of all sources when you press 'dipoles'.

The onset-parameter determines when the stimulus, presented at 0 ms peri-stimulus time, is assumed to activate the cortical area to which it is connected. In DCM, we usually do not model the rather small early responses, but start modelling at the first large deflection. Because the propagation of the stimulus impulse through the input nodes causes a delay, we found that the default value of 60 ms onset time is a good value for many evoked responses where the first large deflection is seen around 100 ms. However, this value is a prior, i.e., the inversion routine can adjust it. The prior mean should be chosen according to the specific responses of interest. This is because the time until the first large deflection is dependent on the paradigm or the modality you are working in, e.g. audition or vision. You may also find that changing the onset prior has an effect on how your data are fitted. This is because the onset time has strongly nonlinear effects (a delay) on the data, which might cause differences in which maximum was found at convergence, for different prior values. It is also possible to type several numbers in this box (identical or not) and then there will be several inputs whose timing can be optimized separately. These inputs can be connected to different model sources. This can be useful, for instance, for modelling a paradigm with combined auditory and visual stimulation.

When you want to proceed to the next model specification stage, hit the > (forward) button and proceed to the *neuronal model*.

8.7 Neuronal model

There are five (or more) matrices which you need to specify by button presses. The first three are the connection strength parameters for the first evoked response. There are three types of connections, *forward*, *backward* and *lateral*. In each of these matrices you specify a connection *from* a source area *to* a target area. For example, switching on the element (2, 1) in the intrinsic forward connectivity matrix means that you specify a forward connection from area 1 to 2. Some people find the meaning of each element slightly counter-intuitive, because the column index corresponds to the source area, and the row index to the target area. This convention is motivated by direct correspondence between the matrices of buttons in the GUI and connectivity matrices in DCM equations and should be clear to anyone familiar with matrix multiplication.

The one or more inputs that you specified previously can go to any area and to multiple areas. You can select the receiving areas by selecting area indices in the *C input* vector.

The B matrix contains all gain modulations of connection strengths as set in the A-matrices. These modulations model the difference between the first and the other modelled evoked responses. For example, for two evoked responses, DCM explains the first response by using the A-matrix only. The 2nd response is modelled by modulating these connections by the weights in the B-matrix.

8.8 Estimation

When you are finished with model specification, you can hit the *estimate* button in the lower left corner. If this is the first estimation and you have not tried any other source reconstructions with

this file, DCM will build a spatial forward model. You can use the template head model for quick results. For this, you answer 'no' to the question 'Redefine MRI fiducials?', and 'yes' to 'Use headshape points?' DCM will now estimate model parameters. You can follow the estimation process by observing the model fit in the output window. In the matlab command window, you will see each iteration printed out with expected-maximization iteration number, free energy F, and the predicted and actual change of F following each iteration step. At convergence, DCM saves the results in a DCM file, by default named 'DCM_ERP.mat'. You can save to a different name, eg. if you are estimating multiple models, by pressing 'save' at the top of the GUI and writing to a different name.

8.9 Results

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

With *ERPs (mode)* you can plot, for each mode, the data for both evoked responses, and the model fit.

When you select *ERPs (sources)*, the dynamics of each area are plotted. The activity of the pyramidal cells (which is the reconstructed source activity) are plotted in solid lines, and the activity of the two interneuron populations are plotted as dotted lines.

The option *coupling (A)* will take you to a summary about the posterior distributions of the connections in the A-matrix. In the upper row, you see the posterior means for all intrinsic connectivities. As above, element (i, j) corresponds to a connection from area j to i. In the lower row, you'll find, for each connection, the probability that its posterior mean is different from the prior mean, taking into account the posterior variance.

With the option *coupling(B)* you can access the posterior means for the gain modulations of the intrinsic connectivities and the probability that they are unequal to the prior means. If you specified several experimental effects, you will be asked which of them you want to look at.

With *coupling(C)* you see a summary of the posterior distribution for the strength of the input into the input receiving area. On the left hand side, DCM plots the posterior means for each area. On the right hand side, you can see the corresponding probabilities.

The option *Input* shows you the estimated input function. As described by [13], this is a gamma function with the addition of low-frequency terms.

With *Response*, you can plot the selected data, i.e. the data, selected by the spatial modes, but back-projected into sensor space.

With *Response (image)*, you see the same as under Results but plotted as an image in grey-scale.

And finally, with the option *Dipoles*, DCM displays an overlay of each dipole on an MRI template using the posterior means of its 3 orientation and 3 location parameters. This makes sense only if you have selected an ECD model under *electromagnetic model*.

Before estimation, when you press the button 'Initialise' you can assign parameter values as initial starting points for the free-energy gradient ascent scheme. These values are taken from another already estimated DCM, which you have to select.

The button *BMS* allows you do Bayesian model comparison of multiple models. It will open the SPM batch tool for model selection. Specify a directory to write the output file to. For the "Inference method" you can choose between "Fixed effects" and "Random effects" (see [65] for additional explanations). Choose "Fixed effects" if you are not sure. Then click on "Data" and in the box below click on "New: Subject". Click on "Subject" and in the box below on "New: Session". Click on models and in the selection window that comes up select the DCM mat files for all the models (remember the order in which you select the files as this is necessary for interpretating the results). Then run the model comparison by pressing the green "Run" button. You will see, at the top, a bar plot of the log-model evidences for all models. At the bottom, you will see the probability, for each model, that it produced the data. By convention, a model can be said to be the best among a selection of other models, with strong evidence, if its log-model evidence exceeds all other log-model evidences by at least 3.

8.10   Steady-State Responses

8.10.1   Model specification

DCM for steady state responses can be applied to M/EEG or intracranial data.

The data must first be prepared using the temporal pre-processing option; 'other', then 'prep'. Here you specify the data type (MEG, EEG or LFP), and select the channels for a DCM analysis. In general you select all MEG/EEG channels and all or a subgroup of LFP channels from the areas you are interested in. A sample script to convert raw txt data, called textitspm_lfp_txt2mat can be found in the folder toolbox\Neural_Models.

The top panel of the DCM for ERP window allows you to toggle through available analysis methods. On the top left drop-down menu, select 'SSR'. The second drop-down menu in the right of the top-panel allows you to specify whether the analysis should be performed using a model which is linear in the states, for this you can choose ERP. Alternatively you may use a conductance based model, which is non-linear in the states by choosing, 'NMM' or 'MFM'. (see [49] for a description of the differences).

The steady state (frequency) response is generated automatically from the time domain recordings. The time duration of the frequency response is entered in the second panel in the time-window. The options for detrending allow you to remove either 1st, 2nd, 3rd or 4th order polynomial drifts from channel data. In the subsampling option you may choose to downsample the data before constructing the frequency response. The number of modes specifies how many components from the leadfield are present in channel data. The specification of between trial effects and design matrix entry is the same as for the case of ERPs, described above.

8.10.2   The Lead-Field

The cross-spectral density is a description of the dependencies among the observed outputs of these neuronal sources. To achieve this frequency domain description we must first specify the likely sources and their location. If LFP data are used then only source names are required. This information is added in the third panel by selecting 'LFP'. Alternatively, x,y,z coordinates are specified for ECD or IMG solutions.

8.10.3   Connections

The bottom panel then allows you to specify the connections between sources and whether these sources can change from trial type to trial type.

On the first row, three connection types may be specified between the areas. For NMM and MFM options these are Excitatory, Inhibitory or Mixed excitatory and inhibitory connections. When using the ERP option the user will specify if connections are 'Forward', 'Backward' or 'Lateral'. To specify a connection, switch on the particular connection matrix entry. For example to specify an Inhibitory connection from source 3 to source 1, turn on the 'Inhib' entry at position (3,1).

On this row the inputs are also specified. These are where external experimental inputs enter the network.

The matrix on the next row allows the user to select which of the connections specified above can change across trial types. For example in a network of two sources with two mixed connections (1,2) and (2,1), you may wish to allow only one of these to change depending on experimental context. In this case, if you wanted the mixed connection from source 2 to source 1 to change depending on trial type, then select entry (2,1) in this final connection matrix.

8.10.4   Cross Spectral Densities

The final selection concerns what frequencies you wish to model. These could be part of a broad frequency range e.g. like the default 4 - 48 Hz, or you could enter a narrow band e.g. 8 to 12 Hz, will model the alpha band in 1Hz increments.

Once you hit the 'invert DCM' option the cross spectral densities are computed automatically (using the spectral-toolbox). The data for inversion includes the auto-spectra and cross-spectra between channels or between channel modes. This is computed using a multivariate autoregressive

model, which can accurately measure periodicities in the time-domain data. Overall the spectra are then presented as an upper-triangular, s x s matrix, with auto-spectra on the main diagonal and cross-spectra in the off-diagonal terms.

8.10.5   Output and Results

The results menu provides several data estimates. By examining the 'spectral data', you will be able to see observed spectra in the matrix format described above. Selecting 'Cross-spectral density' gives both observed and predicted responses. To examine the connectivity estimates you can select the 'coupling (A)' results option, or for the modulatory parameters, the 'coupling (B)' option. Also you can examine the input strength at each source by selecting the 'coupling (C)' option, as in DCM for ERPs. The option 'trial-specific effects' shows the change in connectivity parameter estimates (from B) from trial to trial relative to the baseline connection (from A). To examine the spectral input to these sources choose the 'Input' option; this should look like a mixture of white and pink noise. Finally the 'dipoles' option allows visualisation of the a posteriori position and orientation of all dipoles in your model.

8.11   Induced responses

DCM for induced responses aims to model coupling within and between frequencies that are associated with linear and non-linear mechanisms respectively. The procedure to do this is similar to that for DCM for ERP/ERF. In the following, we will just point out the differences in how to specify models in the GUI. Before using the technique, we recommend reading about the principles behind DCM for induced responses [10].

8.11.1   Data

The data to be modelled must be single trial, epoched data. We will model the entire spectra, including both the evoked (phase-locked to the stimulus) and induced (non-phase-locked to the stimulus) components.

8.11.2   Electromagnetic model

Currently, DCM for induced responses uses only the ECD method to capture the data features. Note that a difference to DCM for evoked responses is that the parameters of the spatial model are not optimized. This means that DCM for induced responses will project the data into source space using the spatial locations provided by you.

8.11.3   Neuronal model

This is where you specify the connection architecture. Note that in DCM for induced responses, the A-matrix encodes the linear and nonlinear coupling strength between sources.

8.11.4   Wavelet transform

This function can be called below the connectivity buttons and allows one to transfer data into the time-frequency domain using a Morlet Wavelet transform as part of the feature extraction. There are two parameters: The frequency window defines the desired frequency band and the wavelet number specifies the temporal-frequency resolution. We recommend values greater than 5 to obtain a stable estimation.

8.11.5   Results

Frequency modes

This will display the frequency modes, identified using singular value decomposition of spectral dynamics in source space (over time and sources).

Time-Frequency

This will display the observed time-frequency power data for all pre-specified sources (upper panel) and the fitted data (lower panel).

Coupling (A-Hz)

This will display the coupling matrices representing the coupling strength from source to target frequencies.

8.12   Phase-coupled responses

DCM for phase-coupled responses is based on a weakly coupled oscillator model of neuronal interactions.

8.12.1   Data

The data to be modeled must be multiple trial, epoched data. Multiple trials are required so that the full state-space of phase differences can be explored. This is achieved with multiple trials as each trial is likely to contain different initial relative phase offsets. Information about different trial types is entered as it is with DCM for ERP ie. using a design matrix. DCM for phase coupling is intended to model dynamic transitions toward synchronization states. As these transitions are short it is advisable to use short time windows of data to model and the higher the frequency of the oscillations you are interested in, the shorter this time window should be. DCM for phase coupling will probably run into memory problems if using long time windows or large numbers of trials.

8.12.2   Electromagnetic model

Currently, DCM for phase-coupled responses will work with either ECD or LFP data. Note that a difference to DCM for evoked responses is that the parameters of the spatial model are not optimized. This means that DCM for phase-coupled responses will project the data into source space using the spatial locations you provide.

8.12.3   Neuronal model

This is where you specify the connection architecture for the weakly coupled oscillator model. If using the GUI, the Phase Interaction Functions are given by $a_{ij}sin(\phi_i - \phi_j)$ where a_{ij} are the connection weights that appear in the A-matrix and ϕ_i and ϕ_j are the phases in regions i and j. DCM for phase coupling can also be run from a MATLAB script. This provides greater flexibility in that the Phase Interaction Functions can be approximated using arbitrary order Fourier series. Have a look in the `example_scripts` to see how.

8.12.4   Hilbert transform

Pressing this button does two things. First, source data are bandpass filtered into the specified range. Second, a Hilbert transform is applied from which time series of phase variables are obtained.

8.12.5   Results

Sin(Data) - Region i

This plots the sin of the data (ie. sin of phase variable) and the corresponding model fit for the ith region.

Coupling (A),(B)

This will display the intrinsic and modulatory coupling matrices. The i, jth entry in A specifies how quickly region i changes its phase to align with region j. The corresponding entry in B shows how these values are changed by experimental manipulation.

Part V

Utilities

Part VI

Tools

Chapter 9

FieldMap Toolbox

9.1 Introduction

This chapter describes how to use the FieldMap toolbox version 2.1[1] for creating unwrapped field maps that can be used to do geometric distortion correction of EPI images [37, 38, 2]. The methods are based on earlier work by Jezzard et al.,[40] and a phase-unwrapping algorithm by Jenkinson [39]. The toolbox can be used via the SPM batch editor or in an interactive mode so that the user can see the effect of applying different field maps and unwarping parameters to EPI images. A voxel displacement map (VDM) is created that can be used with Realign & Unwarp for doing a combined static and dynamic distortion correction or with an Apply VDM function for doing a static distortion correction on a set of realigned images. Realign & Unwarp is designed to work only with images acquired with the phase-encode direction aligned with the anterior-posterior axis. Images acquired with phase-encode directions aligned with other axes can be distortion corrected using the FieldMap toolbox and Apply VDM utility.

9.2 Presubtracted Phase and Magnitude Data

Calculate a voxel displacement map (VDM) from presubtracted phase and magnitude field map data (Figure 9.1). This option expects a single magnitude image and a single phase image resulting from the subtraction of two phase images (where the subtraction is usually done automatically by the scanner software). The phase image will be scaled between +/- PI.

9.2.1 Data

Subjects or sessions for which individual field map data has been acquired.

Subject

Data for this subject or field map session.

Phase Image   Select a single phase image. This should be the result from the subtraction of two phase images (where the subtraction is usually done automatically by the scanner software). The phase image will be scaled between +/- PI.

Magnitude Image   Select a single magnitude image. This is used for masking the phase information and coregistration with the EPI data. If two magnitude images are available, select the one acquired at the shorter echo time because it will have greater signal

FieldMap defaults   FieldMap default values can be entered as a file or set of values.

---

[1] FieldMap Version 2.0 can be downloaded as part SPM5: http://www.fil.ion.ucl.ac.uk/spm/software/spm5/
FieldMap Version 1.1 for SPM2 can be downloaded from http://www.fil.ion.ucl.ac.uk/spm/toolbox/fieldmap/
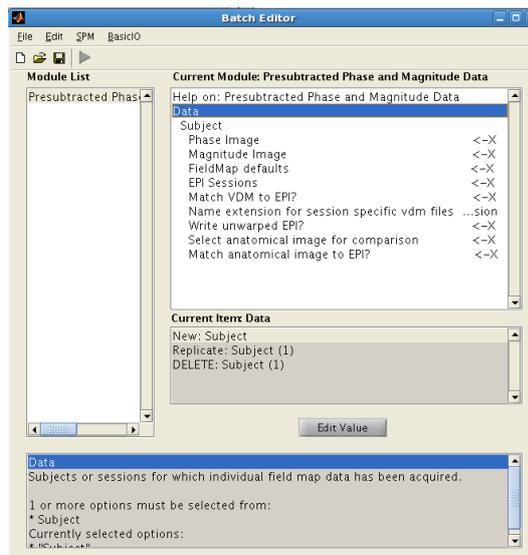
Figure 9.1: FieldMap using the SPM8 User Interface.

Defaults values   Defaults values

Echo times [short TE long TE] Enter the short and long echo times (in ms) of the data used to acquire the field map.

Mask brain Select masking or no masking of the brain. If masking is selected, the magnitude image is used to generate a mask of the brain.

Blip direction Enter the blip direction. This is the polarity of the phase-encode blips describing the direction in which k-space is traversed along the y-axis during EPI acquisition with respect to the coordinate system used in SPM. In this coordinate system, the phase encode direction corresponds with the y-direction and is defined as positive from the posterior to the anterior of the head.

The convention used to describe the direction of the k-space traversal is based on the coordinate system used by SPM. In this coordinate system, the phase encode direction corresponds with the y-direction and is defined as positive from the posterior to the anterior of the head. The x-direction is defined as positive from left to right and the z-direction is defined as positive from foot to head. The polarity of the phase-encode blips describes in which direction k-space is traversed along the y-axis with respect to the coordinate system described here.

Total EPI readout time Enter the total EPI readout time (in ms). This is the time taken to acquire all of the phase encode steps required to cover k-space (ie one image slice). For example, if the EPI sequence has 64 phase encode steps, the total readout time is the time taken to acquire 64 echoes, e.g. total readout time = number of echos x echo spacing. This time does not include i) the duration of the excitation, ii) the delay between, the excitation and the start of the acquisition or iii) time for fat saturation etc.

EPI-based field map? Select non-EPI or EPI based field map. The field map data may be acquired using a non-EPI sequence (typically a gradient echo sequence) or an EPI sequence. The processing will be slightly different for the two cases. If using an EPI-based field map, the resulting Voxel Displacement Map will be inverted since the field map was acquired in distorted space.

Jacobian modulation? Select whether or not to use Jacobian modulation. This will adjust the intensities of voxels that have been stretched or compressed but in general is not recommended for EPI distortion correction

uflags Different options for phase unwrapping and field map processing

UNWRAPPING METHOD Select method for phase unwrapping

FWHM FWHM of Gaussian filter used to implement weighted smoothing of unwrapped maps.

PAD Size of padding kernel if required.

WEIGHTED SMOOTHING Select normal or weighted smoothing.

mflags Different options used for the segmentation and creation of the brain mask.

TEMPLATE IMAGE FOR BRAIN MASKING Select template file for segmentation to create brain mask

FWHM FWHM of Gaussian filter for smoothing brain mask.

NUMBER OF EROSIONS Number of erosions used to create brain mask.

NUMBER OF DILATIONS Number of dilations used to create brain mask.

THRESHOLD Threshold used to create brain mask from segmented data.

REGULARIZATION Regularization value used in the segmentation. A larger value helps the segmentation to converge.

Defaults File Select the 'pm_defaults*.m' file containing the parameters for the field map data. Please make sure that the parameters defined in the defaults file are correct for your field map and EPI sequence. To create your own customised defaults file, either edit the distributed version and/or save it with the name 'pm_defaults_yourname.m'.

EPI Sessions If a single set of field map data will be used for multiple EPI runs/sessions, select the first EPI in each run/session. A VDM file will created for each run/session, matched to the first EPI in each run/session and saved with a unique name extension.

Session Data for this session.

Select EPI to Unwarp Select a single image to distortion correct. The corrected image will be saved with the prefix u. Note that this option is mainly for quality control of correction so that the original and distortion corrected images can be displayed for comparison. To unwarp multiple images please use either Realign & Unwarp or Apply VDM.

Match VDM to EPI? Match VDM file to EPI image. This will coregister the field map data to the selected EPI for each run/session.

In general, the field map data should be acquired so that it is as closely registered with the EPI data as possible but matching can be selected if required. If a precalculated field map was loaded then the user is prompted to select a magnitude image in the same space as the field map. If real and imaginary images were selected, the toolbox automatically creates a magnitude image from these images and saves it with the name mag_NAME-OF-FIRST-INPUT-IMAGE.img.

Name extension for run/session specific VDM file This will be the name extension followed by an incremented integer for run/session specific VDM files.

Write unwarped EPI? Write out distortion corrected EPI image. The image is saved with the prefix u. Note that this option is mainly for quality control of correction so that the original and distortion corrected images can be displayed for comparison. To unwarp multiple images please use either Realign & Unwarp or Apply VDM.

Select anatomical image for comparison Select an anatomical image for comparison with the distortion corrected EPI or leave empty. Note that this option is mainly for quality control of correction.

Match anatomical image to EPI? Match the anatomical image to the distortion corrected EPI. Note that this option is mainly for quality control of correction allowing for visual inspection and comparison of the distortion corrected EPI.

9.3 Real and Imaginary Data

Calculate a voxel displacement map (VDM) from real and imaginary field map data. This option expects two real and imaginary pairs of data of two different echo times. The phase images will be scaled between +/- PI.

9.3.1   Data

Subjects or sessions for which individual field map data has been acquired.

Subject

Data for this subject or field map session.

Short Echo Real Image   Select short echo real image

Short Echo Imaginary Image   Select short echo imaginary image

Long Echo Real Image   Select long echo real image

Long Echo Imaginary Image   Select long echo imaginary image

Other inputs   As for Presubtracted Phase and Magnitude Data.

9.4   Phase and Magnitude Data

Calculate a voxel displacement map (VDM) from double phase and magnitude field map data.
This option expects two phase and magnitude pairs of data of two different echo times.

9.4.1   Data

Subjects or sessions for which individual field map data has been acquired.

Subject

Data for this subject or field map session.

Short Echo Phase Image   Select short echo phase image

Short Echo Magnitude Image   Select short echo magnitude image

Long Echo Phase Image   Select long echo phase image

Long Echo Magnitude Image   Select long echo magnitude image

Other inputs   As for Presubtracted Phase and Magnitude Data.

9.5   Precalculated FieldMap (in Hz)

Calculate a voxel displacement map (VDM) from a precalculated field map. This option expects
a processed field map (ie phase unwrapped, masked if necessary and scaled to Hz). Precalculated
field maps can be generated by the FieldMap toolbox and stored as fpm_* files.

9.5.1   Data

Subjects or sessions for which individual field map data has been acquired.

Subject

Data for this subject or field map session.

Precalculated field map Select a precalculated field map. This should be a processed field map (ie phase unwrapped, masked if necessary and scaled to Hz) , for example as generated by the FieldMap toolbox and are stored with fpm_* prefix.

Select magnitude image in same space as fieldmap Select magnitude image which is in the same space as the field map to do matching to EPI.

Other inputs As for Presubtracted Phase and Magnitude Data.

9.6 Apply VDM

Apply VDM (voxel displacement map) to resample voxel values in selected image(s). This allows a VDM to be applied to any images which are assumed to be already realigned (e.g. including EPI fMRI time series and DTI data).

The VDM can be been created from a field map acquisition using the FieldMap toolbox and comprises voxel shift values which describe geometric distortions occuring as a result of magnetic susceptbility artefacts. Distortions along any single dimension can be corrected therefore input data may have been acquired with phase encode directions in X, Y (most typical) and Z.

The selected images are assumed to be realigned to the first in the time series (e.g. using Realign: Estimate) but do not need to be resliced. The VDM is assumed to be in alignment with the images selected for resampling (note this can be achieved via the FieldMap toolbox). The resampled images are written to the input subdirectory with the same (prefixed) filename.

e.g. The typical processing steps for fMRI time series would be 1) Realign: Estimate, 2) FieldMap to create VDM, 3) Apply VDM.

Note that this routine is a general alternative to using the VDM in combination with Realign & Unwarp which estimates and corrects for the combined effects of static and movement-related susceptibility induced distortions. Apply VDM can be used when dynamic distortions are not (well) modelled by Realign & Unwarp (e.g. for fMRI data acquired with R->L phase-encoding direction, high field fMRI data or DTI data).

9.6.1 Data

Subjects or sessions for which VDM file is being applied to images.

Session

Data for this session.

Images Select scans for this session. These are assumed to be realigned to the first in the time series (e.g. using Realign: Estimate) but do not need to be resliced

Fieldmap (vdm* file) Select VDM (voxel displacement map) for this session (e.g. created via FieldMap toolbox). This is assumed to be in alignment with the images selected for resampling (note this can be achieved via the FieldMap toolbox).

9.6.2 Reslice Options

Apply VDM reslice options

Distortion direction

In which direction are the distortions? Any single dimension can be corrected therefore input data may have been acquired with phase encode directions in Y (most typical), X or Z

Reslice which images ?

All Images (1..n)
  This applies the VDM and reslices all the images.
  All Images + Mean Image
  This applies the VDM reslices all the images and creates a mean of the resliced images.

Interpolation

The method by which the images are sampled when being written in a different space. Nearest
Neighbour is fastest, but not recommended for image realignment. Bilinear Interpolation is
probably OK for PET, but not so suitable for fMRI because higher degree interpolation generally
gives better results [66, 67, 68]. Although higher degree methods provide better interpolation,
but they are slower because they use more neighbouring voxels.

Wrapping

This indicates which directions in the volumes the values should wrap around in. For example,
in MRI scans, the images wrap around in the phase encode direction, so (e.g.) the subject's nose
may poke into the back of the subject's head. These are typically:
  No wrapping - for PET or images that have already been spatially transformed. Also the
recommended option if you are not really sure.
  Wrap in Y - for (un-resliced) MRI where phase encoding is in the Y direction (voxel space)
etc.

Masking

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

Filename Prefix

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

9.7   Creating Field Maps Using the FieldMap GUI

The FieldMap Toolbox GUI is shown on the left Figure 9.2. It is divided into two parts. The top
part deals with creating the field map in Hz and the bottom part deals with creating the voxel
displacement map (VDM) and unwarping the EPI. The toolbox can be used by working through
the different inputs in the following order:

9.7.1   Create field map in Hz

Load defaults file

Select the defaults file from which to load default parameters. If necessary, the parameters
used to create the field map can be temporarily modified using the GUI. To change the default
parameters, edit `pm_defaults.m` or create a new file called `pm_defaults_NAME.m` (as described in
Section 9.2.1).
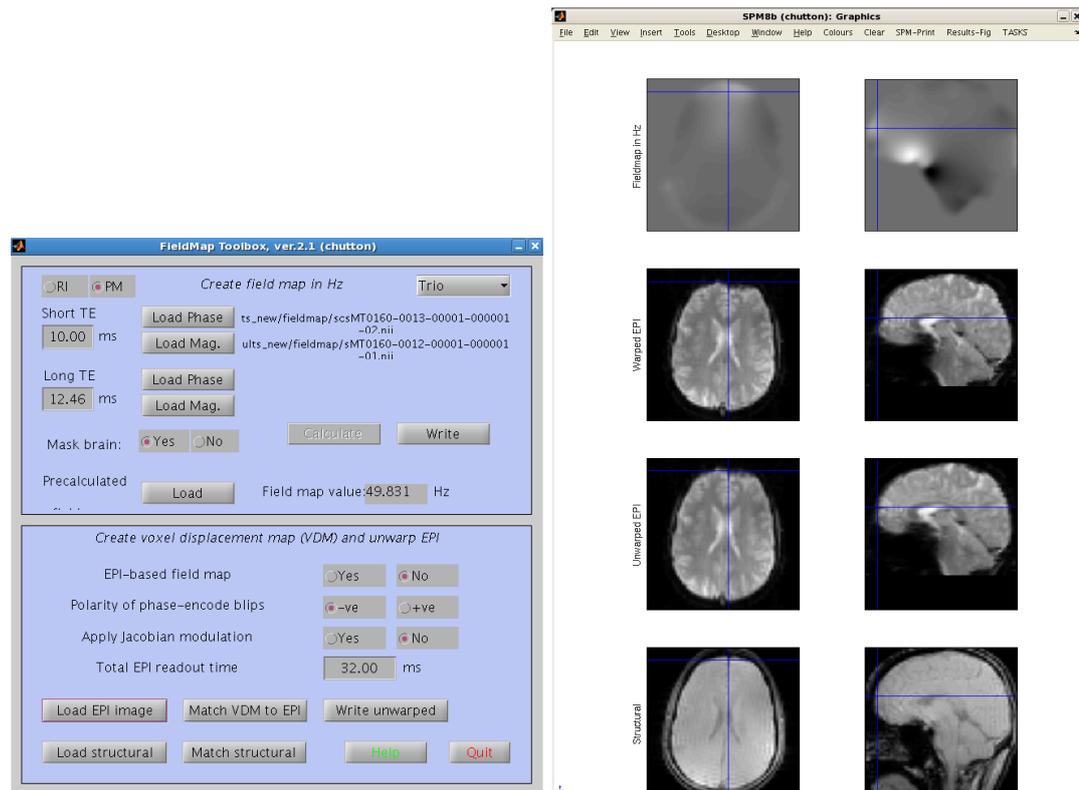
Figure 9.2: FieldMap GUI and Results.

Data Input Format

PM The acquired field map images are in phase and magnitude format. There may be a single pair of phase and magnitude images (i.e. 2 images) in which case the phase image has been created by the vendor sequence from two echo times acquisitions. Alternatively there may be two pairs of phase and magnitude images, one for each echo time(ie 4 images). The units for the phase images MUST BE RADIANS BETWEEN +pi and -pi. The user will be asked if this is required when the images are selected.

RI The acquired field map images are in real and imaginary format. Two pairs of real and imaginary image volumes, one for a shorter and one for a longer echo time (ie 4 images)[2].

File Selection

Select NIfTI format images. Generally, the acquired scanner files will be in dicom format which can be correctly converted using the DICOM converter in the corresponding version of SPM. DICOM and other image formats can also be converted to using MRIcro[3].

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.

---

[2] NB If using SPM2, the data input format can only be changed by editing the spm_defaults.m file. This is described in Section 9.2.1.

[3]MRIcro is freely available from http://www.cla.sc.edu/psyc/faculty/rorden/mricro.html.

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 9.10 and involves some or all of the following steps (as specified in spm_defaults.m):

1. Calculation of a Hz fieldmap from input data

2. Segmentation to exclude regions outside of the brain

3. Phase unwrapping

4. Smoothing and dilation of the processed fieldmap

   The processed field map (in Hz) is displayed in the graphics window (top row, right Figure 9.1) and the field at different points can be explored. The field map in Hz is converted to a VDM (voxel displacement map) using the parameters shown in the FieldMap GUI and saved with the filename vdm5_NAME-OF-FIRST-INPUT-IMAGE.img in the same directory as the acquired field map images. The VDM file is overwritten whenever the field map is recalculated or when any parameters are changed. The resulting VDM file can be used for unwarping the EPI using Realign & Unwarp in SPM8 (see Section 9.9).

Write

Write out the processed field map (in Hz) as a Nifti format image. The image will be saved with the filename fpm_NAME-OF-FIRST-INPUT-IMAGE.img in the same directory as the acquired field map images.

Load Pre-calculated

Load a precalculated unwrapped field map (`fpm_.img`). This should be a single image volume with units of Hz in NIfTI format. The precalculated field map may have been created previously using the FieldMap toolbox or by other means. Once loaded, the field map is displayed in the graphics window (top row, right, Figure 9.1) 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.

9.7.2 Create voxel displacement map (VDM) and unwarp EPI

When any of the parameters below are changed, a new VDM is created and written out as vdm5_NAME-OF-FIRST-INPUT-IMAGE.img. The vdm5_NAME-OF-FIRST-INPUT-IMAGE.mat file is not updated unless 'Match VDM to EPI' is selected as described in Section 9.7.2.

EPI-based field map - Yes/No

Select Yes if the field map is based on EPI data or No otherwise. Most scanner vendor field map sequences are non-EPI.

Polarity of phase-encode blips - +ve/-ve

Select +ve or -ve blip direction.When images are acquired K-space can be traversed using positive or negative phase-encode blips. This direction will influence the geometric distortions in terms of whether the affected regions of the image are stretched or compressed.

The convention used to describe the direction of the k-space traversal is based on the coordinate system used by SPM. In this coordinate system, the phase encode direction corresponds with the y-direction and is defined as positive from the posterior to the anterior of the head. The x-direction is defined as positive from left to right and the z-direction is defined as positive from foot to head. The polarity of the phase-encode blips describes in which direction k-space is traversed along the y-axis with respect to the coordinate system described here.

Apply Jacobian modulation - Yes/No

Select Yes to do Jacobian Modulation to adjust the intensities of voxels that have been stretched or compressed. In general this is not recommended for unwarping EPI data at this stage.

Total EPI readout time (ms)

Enter the total time in ms for the readout of the EPI echo train which is typically 10s of ms. This is the time taken to acquire all of the phase encode steps required to cover k-space (ie one image slice). For example, if the EPI sequence has 64 phase encode steps, the total readout time is the time taken to acquire 64 echoes: total readout time = number of echoes × echo spacing. This time does not include i) the duration of the excitation, ii) the delay between the excitation and the start of the acquisition or iii) time for fat saturation.

Load EPI image

Select a sample EPI image in NIfTI format. This image is automatically 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 9.1).

Match VDM to EPI

Select this option to match the field map magnitude data to the EPI image before it is used to unwarp the EPI. In general, the field map data should be acquired so that it is as closely registered with the EPI data as possible but matching can be selected if required. If a precalculated field map was loaded then the user is prompted to select a magnitude image in the same space as the field map. If real and imaginary images were selected, the toolbox automatically creates a magnitude image from these images and saves it with the name mag_NAME-OF-FIRST-INPUT-IMAGE.img.

Write unwarped

Write unwarped EPI 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 header of the selected structural image.

Help

Call `spm_help` to display `FieldMap.man`.

Quit

Quit the toolbox and closes all windows associated with it.

9.8   Using the FieldMap in Batch scripts

`FieldMap_preprocess.m` which calls `FieldMap_create.m` gives an example of how to run the FieldMap toolbox without using the GUI. To run the script, make sure your MATLAB path includes the directory where the FieldMap toolbox is installed. This can be done using the Set Path option under File in the MATLAB windows manager or using the command:

`addpath /whatever/spm/toolbox/FieldMap`

To run the FieldMap batch script, in MATLAB enter the following command:

`VDM = FieldMap_preprocess(fm_dir,epi_dir, [te1, te2, epifm, tert, kdir, mask, match] );`

where
   fm_dir - name of directory containing fieldmap images.(e.g. fm_dir = '/path/study1/subj1/fieldmap')
   epi_dir - name of directory containing epi images. (e.g. epi_dir = '/path/study1/subj1/images')
   te1 - short echo time (in ms)
   te2 - long echo time (in ms)
   epifm - epi-based fieldmap - yes or no (1/0)
   tert - total echo readout time (in ms)
   kdir - blip direction (1/-1)
   mask do brain segmentation to mask field map (1/0)
   match match vdm file to first EPI in run (1/0).
   NB: FieldMap will match the field map to the first epi image in the time series (after removing the dummy scans). Therefore, epi_dir must be the directory that contains the epi run that all other images will be realigned to.
   The script will create an fpm* file, a vdm5_* file and an unwarped version of the EPI saved with the prescript "u".

9.9   Using the VDM file with Unwarp

In SPM, select the Realign & Unwarp option. For the input data called Phase map (vdm* file), select the vdm5_ or vdm5_ file for the subject and/or session. If you acquired more than one session (or run) of EPI images, you need to select a different vdm5_* file for each one. For more information about Unwarp see http://www.fil.ion.ucl.ac.uk/spm/toolbox/unwarp.

9.10 Appendices

9.10.1 Processing Hz field maps

Processing field maps involves a series of steps for which certain parameters in the spm_defaults file must be set.

1. If the acquired field map data comprises two complex images, the phase difference between them is calculated.

2. The phase map is unwrapped using the method specified by spm_def.UNWRAPPING_METHOD = 'Mark3D' or 'Mark2D' or 'Huttonish'. For a description of these different methods see spm_unwrap.m or FieldMap_principles.man. The default option is 'Mark3D'.

3. A mask is created so that unwrapping only occurs in regions where there is signal. If necessary, this mask can be expanded so that any voxel that hasn't been unwrapped and is less than spm_def.PAD/2 voxels away from an unwrapped one will be replaced by an average of the surrounding unwrapped voxels. This can be done by setting the parameter spm_def.PAD to a value greater than 0. The default value is 0 but a value ¿ 0 (eg 10) may be necessary if normal smoothing is chosen instead of weighted smoothing (as explained in the next step).

4. If required a mask can be generated to exclude regions of the fieldmap outside of the brain (in addition to the unwrapping mask described above). This step uses SPM segmentation for which the parameters in spm_def.MFLAGS can be set. For example, if the segmentation fails, (maybe because the fieldmap magnitude image doesn't have enough contrast), spm_def.MFLAGS.REG can be increased to say 0.05). The other parameters control morphological operations to generate a smooth brain mask and have been set empirically.

5. The unwrapped phase map is scaled by 1/(2*PI*difference in echo time) to convert it to Hz.

6. A weighted gaussian smoothing (weighted by the inverse of the noise) is performed on the unwrapped phase-map if the parameter spm_def.WS = 1. If spm_def.WS = 0, a normal smoothing is done. The weighted smoothing is particularly slow on large data sets ie high resolution. If field maps are acquired at high resolution then it is recommended to use spm_def.WS = 0 and do some padding of the intensity mask eg spm_def.PAD = 10. The size of the Gaussian filter used to implement either weighted or normal smoothing of the unwrapped maps is usually set to spm_def.FWHM = 10.

9.10.2 Converting Hz field map to VDM

1. The field map in Hz is multiplied by the total EPI readout time (in ms,) of the EPI image to be unwarped, resulting in a VDM. The readout time is specified by spm_def.TOTAL_EPI_READOUT_TIM (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 × echo spacing. This time does not include i) the duration of of the excitation, ii) the delay between the excitation and the start of the acquisition or iii) time for fat saturation etc.

2. The VDM is multiplied by +/-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 decresed and where voxels were stretched, the intensities are increased slightly. The modulation involves multiplying the unwarped EPI by 1 + the 1-d derivative of the VDM in the phase direction. An intensity adjustment of this nature may improve the coregistration results between an 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 coregiseterd 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.

9.10.3   Matching field map data to EPI data

1. If required, the fieldmap can be matched to the EPI. This is done slightly differently depending on whether the field map is based on EPI or non-EPI data. If using an EPI field map, the magnitude image is coregistered to the EPI. The resulting transformation matrix is used to sample the VDM file in the space of the EPI before unwarping.

2. If using a non-EPI field map, the VDM is used to forward warp the magnitude image which is then coregistered to the EPI. The forward warped image is saved with the filename wfmag_NAME-OF-FIRST-INPUT-IMAGE.img.

Part VII

Data sets and examples

Chapter 10

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 (64×64×64 3×3mm×3 mm^3 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 Analyze format and are available from the SPM site [1]. 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:\data\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 "Menu" window, (2) the bottom-left or "Interactive" window and (3) the right-hand or "Graphics" window. Analysis then takes place in three major stages (i) spatial pre-processing, (ii) model specification, review and estimation and (iii) inference. These stages organise the buttons in SPM's Menu window.

10.1   Spatial pre-processing

10.1.1   Realignment

Under the spatial pre-processing section of the SPM base window select REALIGN (EST & RES) from the REALIGN pulldown menu. This will call up a realignment job specification in the batch editor. Then

- 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`". There should be 96 files.

- Save the job file as eg. `DIR\jobs\realign.mat`.

- Press the `RUN` button in the batch editor (green arrow).

This will run the realign job which will 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 10.2. These data are also

---

[1] Auditory fMRI dataset: http://www.fil.ion.ucl.ac.uk/spm/data/auditory/
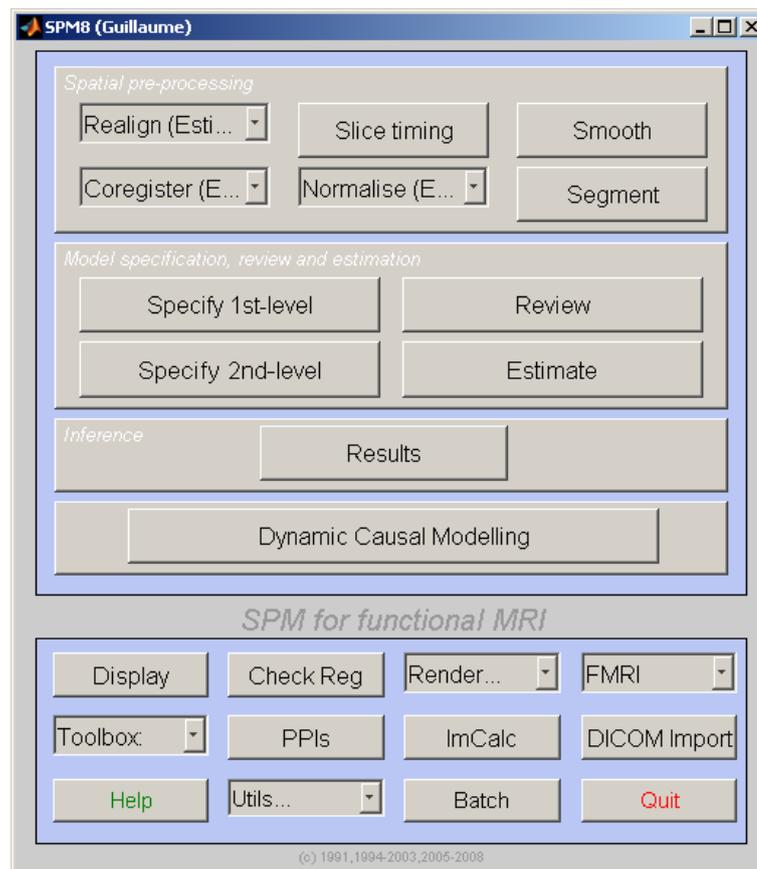
Figure 10.1: *The SPM base window comprises three sections i) spatial pre-processing, (ii) model specification, review and estimation and (iii) inference.*
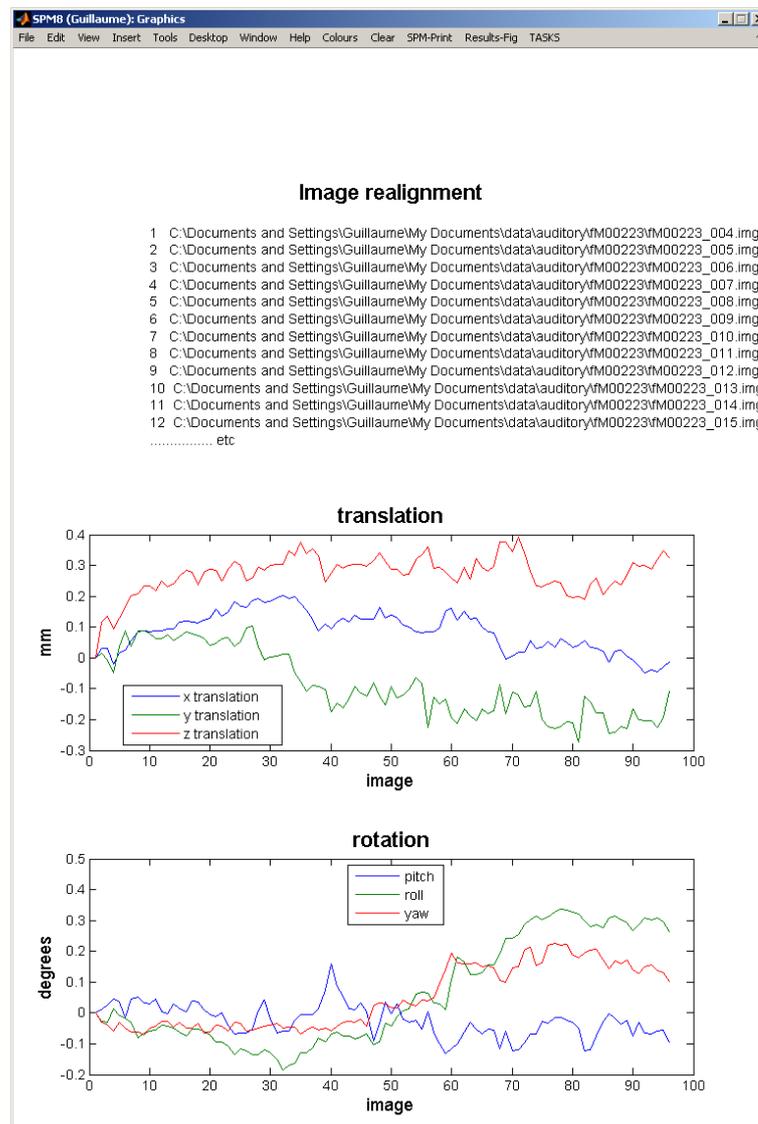
Figure 10.2: *Realignment of Auditory data.*

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.

10.1.2 Coregistration

Select COREGISTER (ESTIMATE) from the COREGISTER pulldown. This will call up the specification of a coregistration job in the batch editor.

- Highlight "Reference Image" and then select the mean fMRI scan from realignment eg. `meanfM00223_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 the `RUN` button.
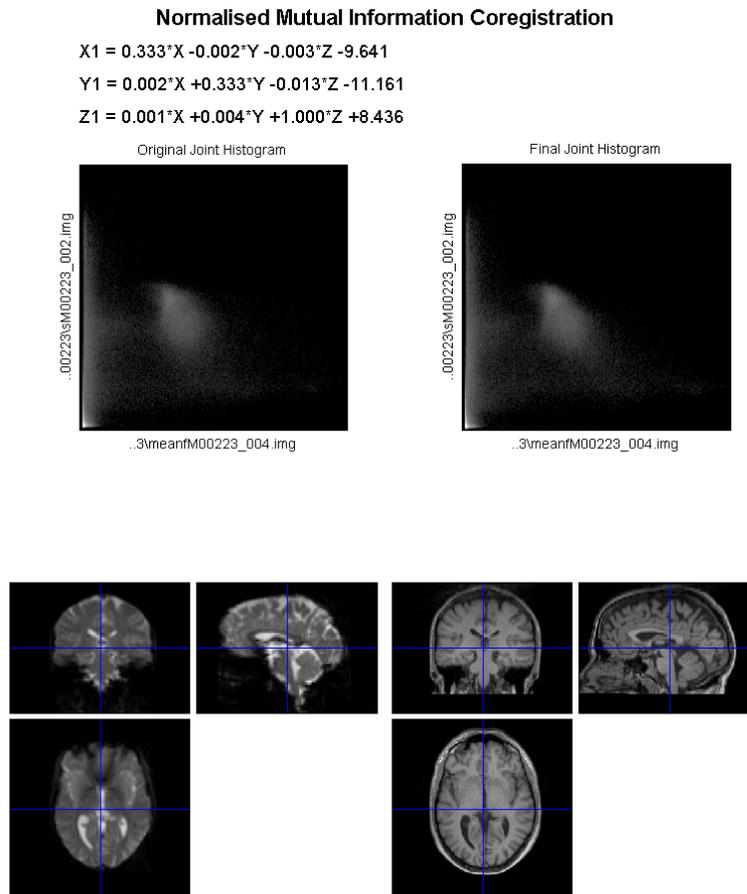
Figure 10.3: *Mutual Information Coregistration of Auditory data.*

SPM will then implement a coregistration between the structural and functional data that maximises the mutual information. The image in figure 10.3 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 10.4 in the Graphics window. You can then use your mouse to navigate these images to confirm that there is an anatomical correspondence.

10.1.3   Segmentation

Press the SEGMENT button. This will call up the specification of a segmentation job in the batch editor. 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 10.5 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. Figure 10.5 shows the gray matter image, c1sM0023_002.img along with the original structural.
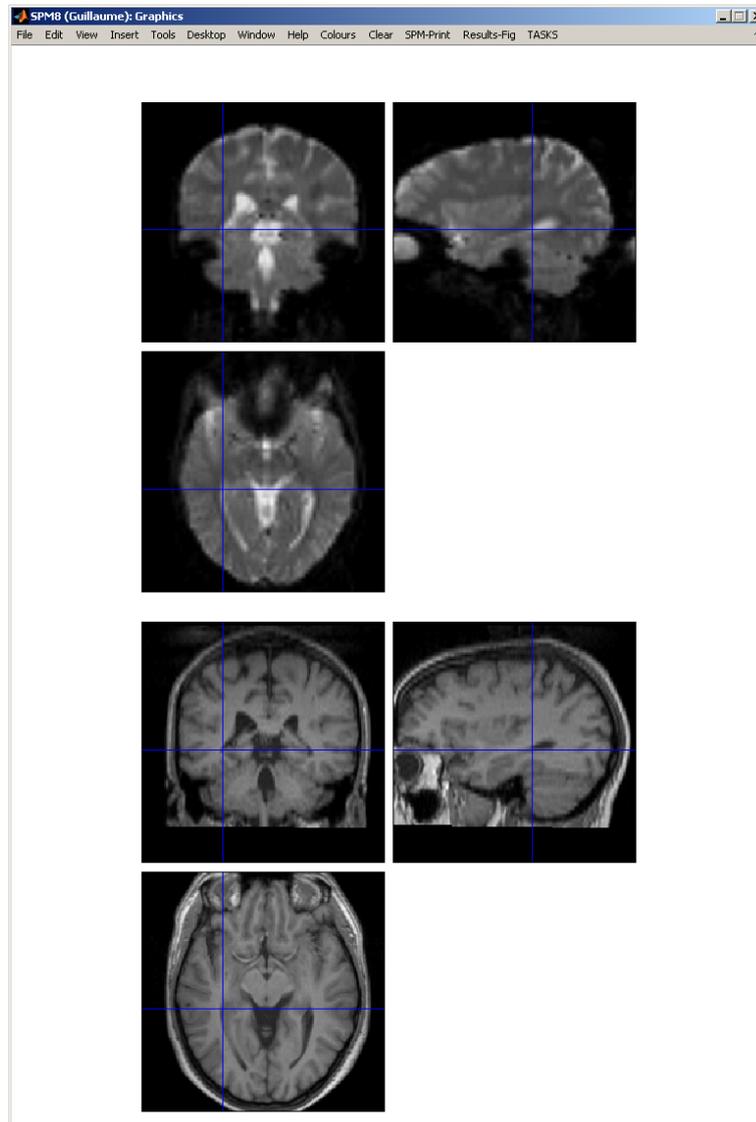
Figure 10.4: *Checking registration of functional and "registered" structural data.*
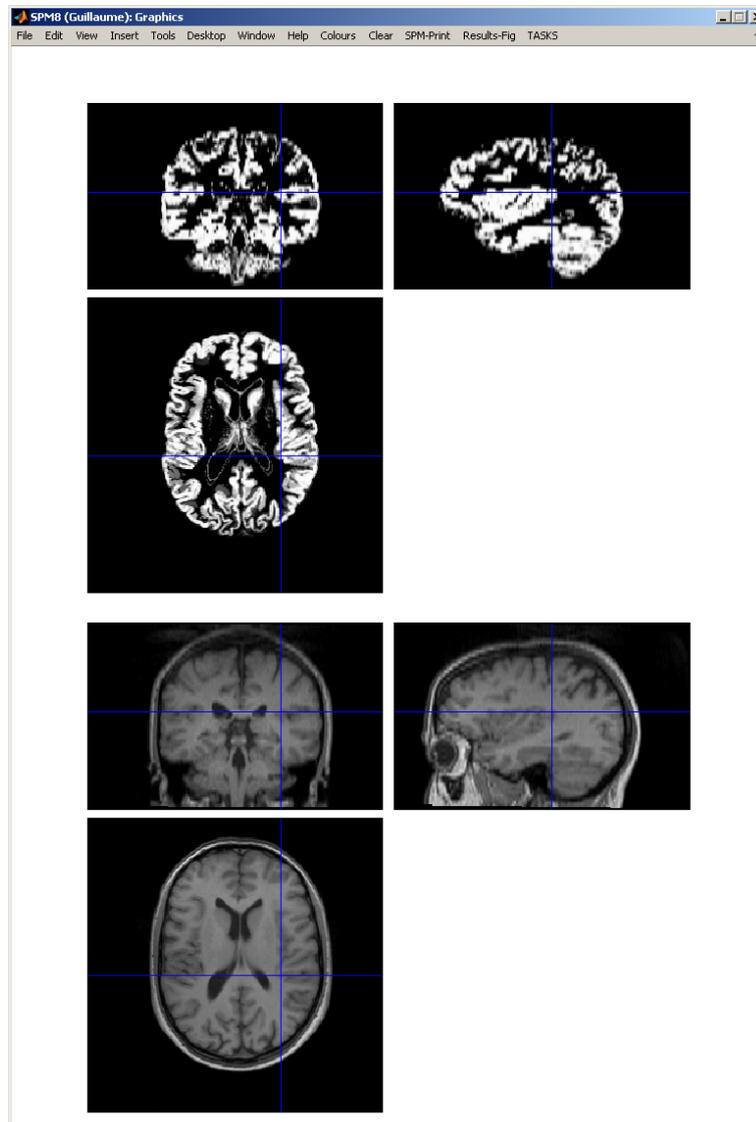
Figure 10.5: *Gray matter image and "registered" structural image.*

Figure 10.6 shows the structural and bias-corrected image, `msM0023_002.img`.

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.

10.1.4   Normalise

Select NORMALISE (WRITE) from the NORMALISE pulldown menu. This will call up the specification of a normalise job in the batch editor.

- Highlight "Data", select New "Subject",

- Highlight "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
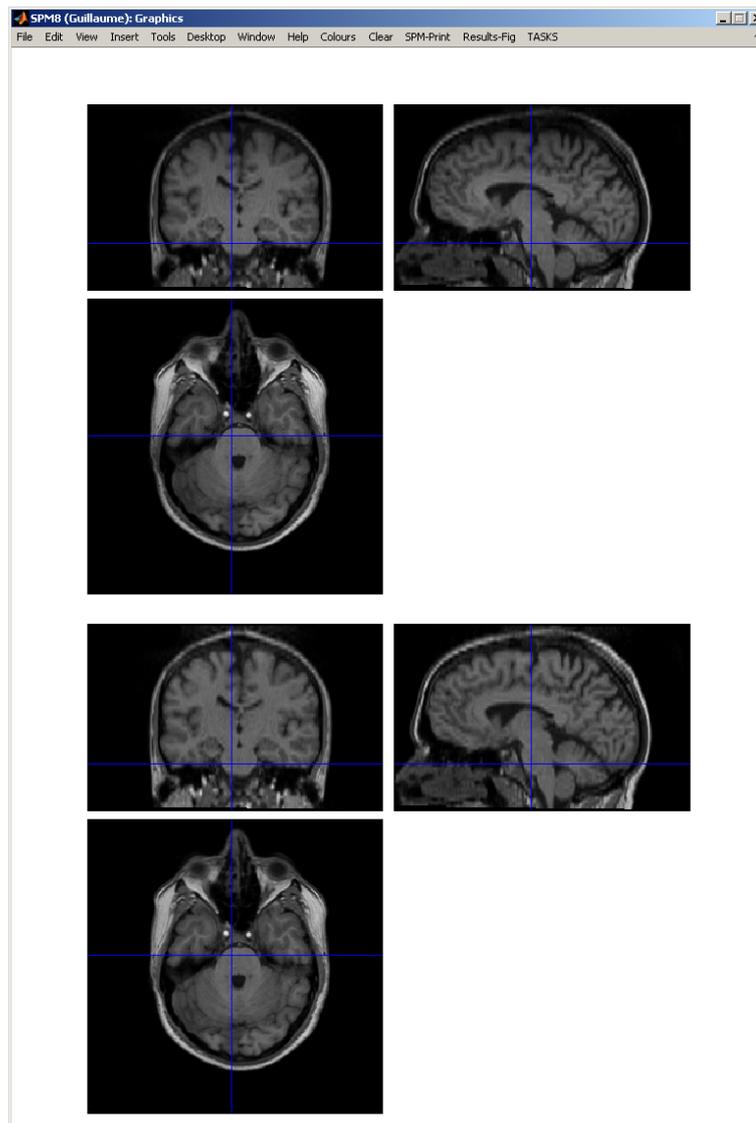
Figure 10.6: *Structural image (top) and bias-corrected structural image (bottom). Notice that the original structural is darker at the top than at the bottom. This non-uniformity has been removed in the bias-corrected image.*

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].[2]

- Press "Save", save the job as `normalise.mat` and then press the RUN button.

SPM will then write spatially normalised files to the functional data directory. These files have the prefix `w`.

If you wish to superimpose a subject's functional activations on their own anatomy[3] you will also need to apply the spatial normalisation parameters to their (bias-corrected) anatomical image. To do this

- Select NORMALISE (WRITE), highlight "Data", select "New Subject".

- Highlight "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 the Run button.

10.1.5   Smoothing

Press the SMOOTH button[4]. This will call up the specification of a smooth job in the batch editor.

- Select "Images to Smooth" and then select the spatially normalised files created in the last section eg. `^wrf.*`.

- Highlight "FWHM" and change [8 8 8] to [6 6 6]. This will smooth the data by 6mm in each direction.

- Save the job as `smooth.mat` and press the Run button.

An example of functional image and its smoothed version is displayed on Figure 10.7.

10.2   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 batch editor. Then

- 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.

---

[2]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.

[3]Beginners may wish to skip this step, and instead just superimpose functional activations on an "average structural image".

[4]The smoothing step is unnecessary if you are only interested in Bayesian analysis of your functional data.
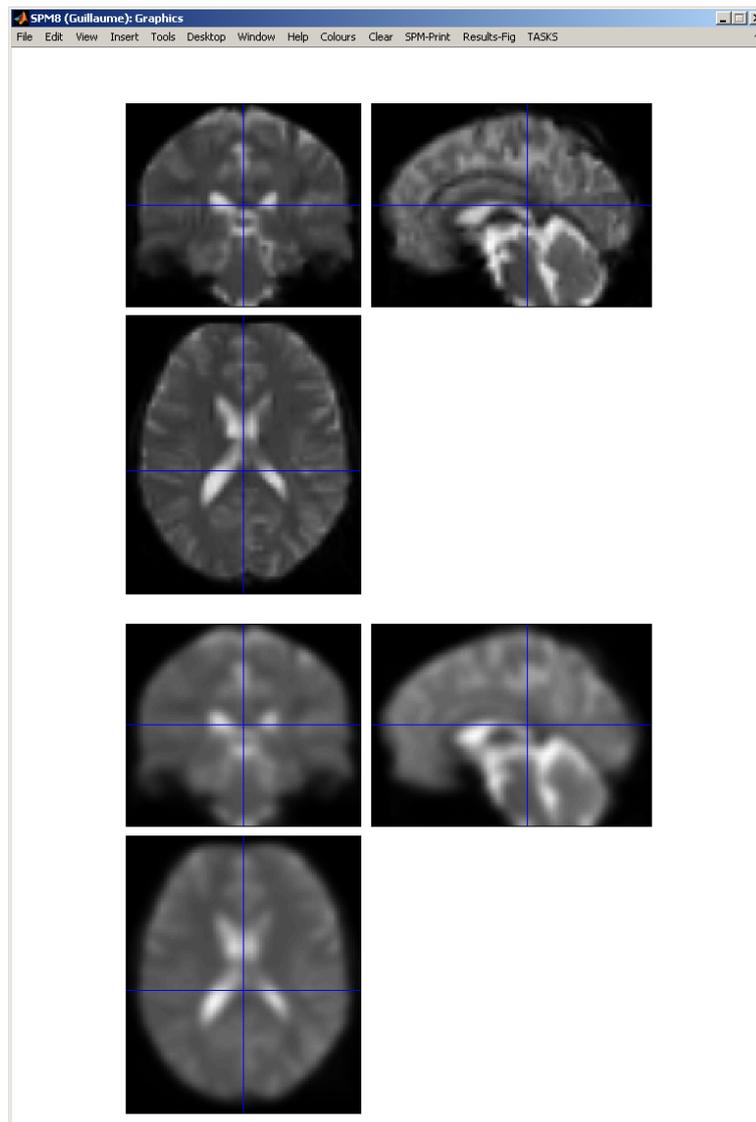
Figure 10.7: *Functional image (top) and 6mm-smoothed functional image (bottom). These images were obtained using SPM's "CheckReg" facility.*
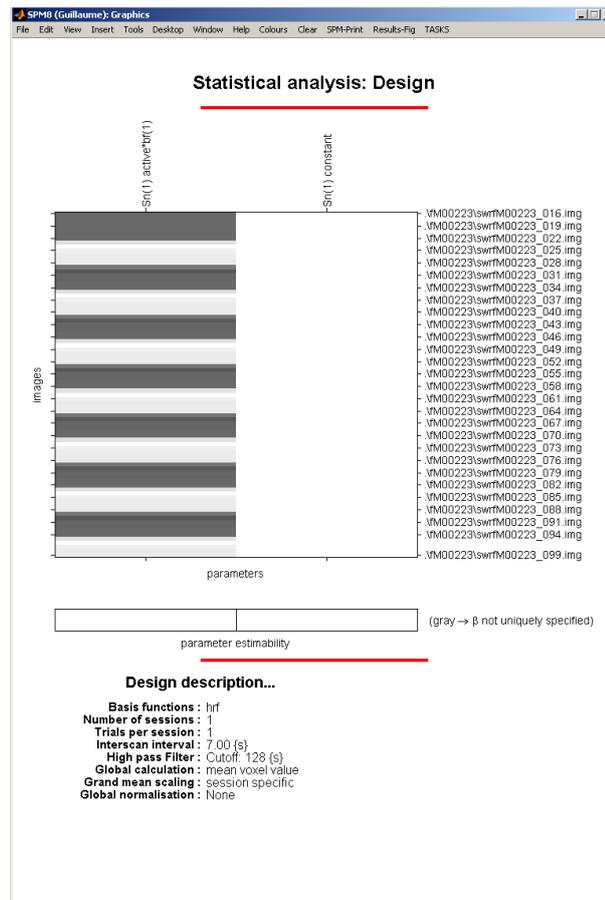
Figure 10.8: *Design matrix: The filenames on the right-hand side of the design matrix indicate the scan associated with each row.*

- Highlight "Scans" and use SPM's file selector to choose the 84 smoothed, normalised functional images ie `swrfM00223_016.img` to `swrfM00223_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 the `Run` button.

SPM will then write an `SPM.mat` file to the `DIR/classical` directory. It will also plot the design matrix, as shown in Figure 10.8.

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 10.8. If you select "Explore" then "Session 1" then "active", SPM will produce the plots shown in Figure 10.9.

If you select the second item on the "Design" tab, "Design Orthogonality", SPM will produce the plot shown in Figure 10.10. 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 θ is the angle between the two vectors. So, the vectors will be orthogonal if $cos\theta = 0$. The
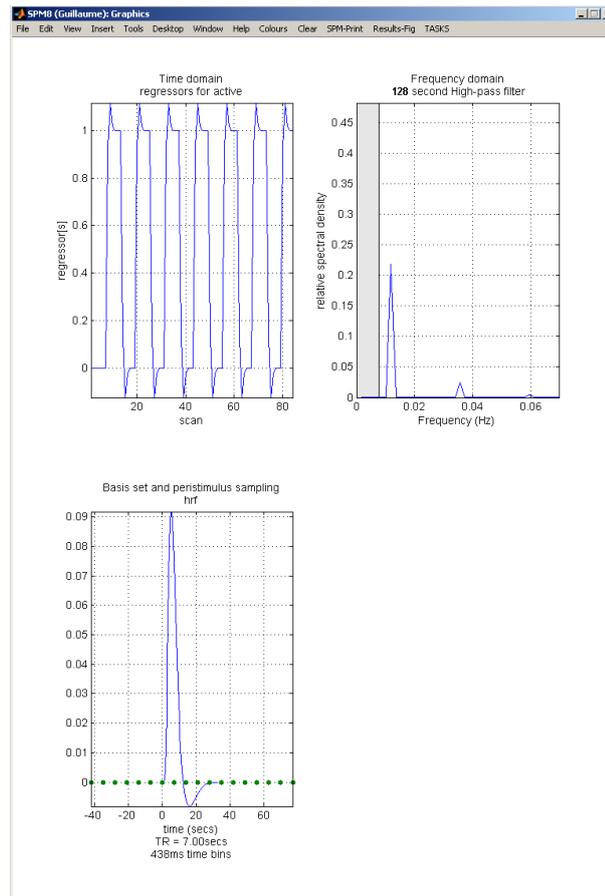
Figure 10.9: *Exploring the design matrix in Figure 10.8: This shows the time series of the "active" regressor (top left), a frequency domain plot of the active regressor (top right) and the basis function used to convert assumed neuronal activity into hemodynamic activity. In this model we used the 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).*
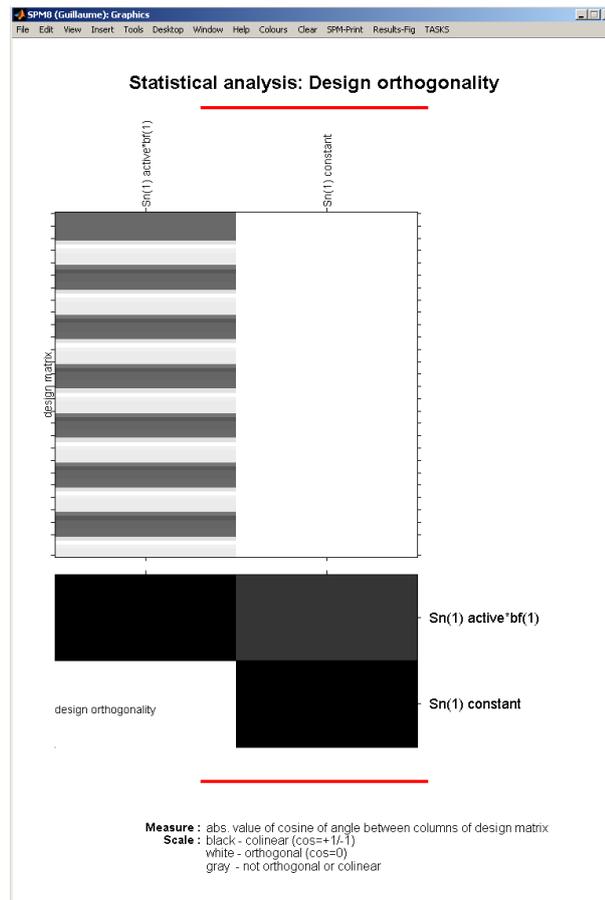
Figure 10.10: *Design Orthogonality: The description above the first column in the design matrix* Sn(1)Active*bf(1) *means that this column refers to the first session of data (in this analysis there is only 1 session), the name of this condition/trial is 'Active' and the trial information has been convolved 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.*

upper-diagonal elements in the matrix at the bottom of figure 10.10 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.

10.2.1   Estimate

Press the ESTIMATE button. This will call up the specification of an fMRI estimation job in the batch editor. Then

- Highlight the "Select SPM.mat" option and then choose the `SPM.mat` file saved in the classical subdirectory.

- Save the job as `estimate.job` and press the `Run` button.

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

10.3   Inference

After estimation:

Figure 10.11: *The contrast manager*
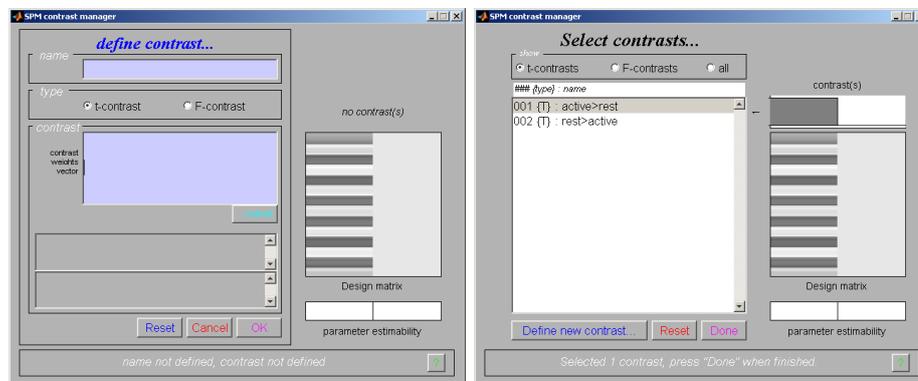
Figure 10.12: *Left: A contrast is entered by specifying the numeric values in the lower window and the name in the upper window. Right: After contrasts have been specified they can be selected.*

- Press "Results".

- Select the `SPM.mat` file created in the last section.

This will invoke the contrast manager.

10.3.1 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".

10.3.2 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.

10.3.3   Thresholds

You will then be prompted with

- *Title for comparison ?*

- Enter eg. "active > rest".

- *p value adjustment to control: [FWE/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.

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 10.13.

10.3.4   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.

10.3.5   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: right-clicking in the MIP will activate a pulldown menu, left-clicking on the red cursor will allow it to be dragged to a new position.
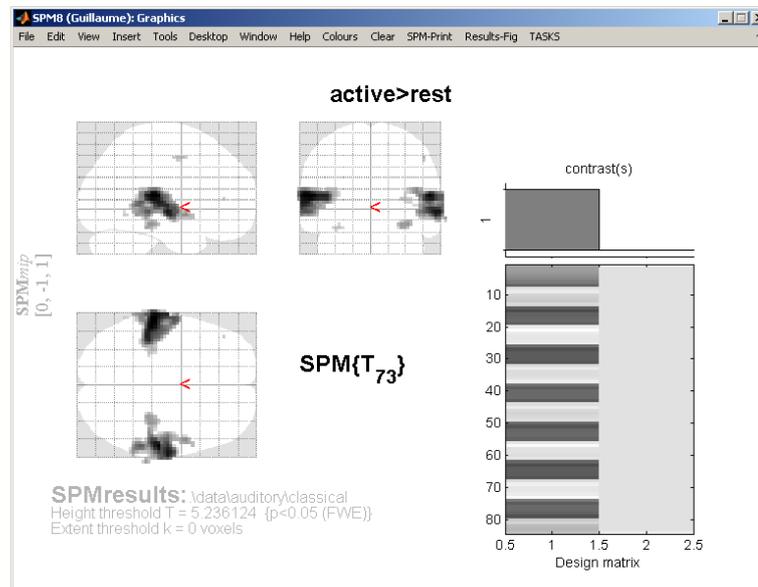
Figure 10.13: *SPM showing bilateral activation of auditory cortex.*
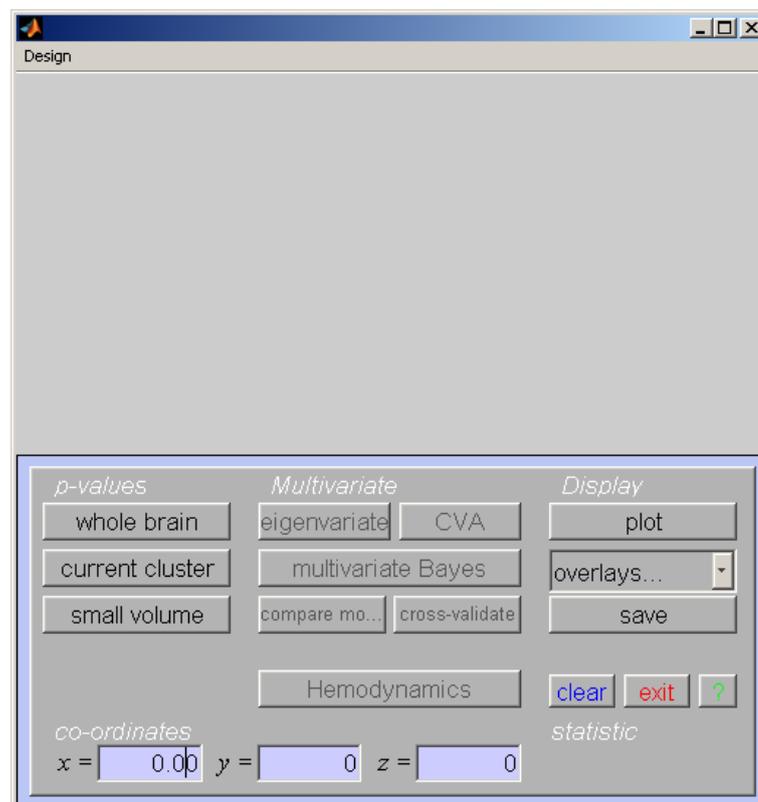
Figure 10.14: *SPM's Interactive window during results assessment. The "p-values" section is used to produce tables of statistical information. The visualisation section is used to plot responses at a voxel or to visual activations overlaid on anatomical images. The "Multivariate" section, ie. the "eigenvariate" button, is used to extract data for subsequent analyses such as assessment of PsychoPhysiological Interactions (PPIs) or Dynamic Causal Models (DCMs).*

Figure 10.15: *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 "whole brain" button. This displays the table of results in a separate window.*

10.3.6   Design matrix

SPM also displays the design matrix with the selected contrast. The design matrix is also surfable: right-clicking will show parameter names, left-clicking will show design matrix values for each scan.

In the SPM Interactive window (lower left panel) a button box appears with various options for displaying statistical results (p-values panel) and creating plots/overlays (visualisation panel). Clicking "Design" (upper left) will activate a pulldown menu as in the "Explore design" option.

10.3.7   Statistical tables

To get a summary of local maxima, press the "whole brain" button in the p-values section of the interactive window. This will list all clusters above the chosen level of significance as well as separate (>8mm apart) maxima within a cluster, with details of significance thresholds and search volume underneath, as shown in Figure 10.15

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

- **x, y, z (mm)**: coordinates in MNI space for each maximum.

- peak-level: the chance (p) of finding (under the null hypothesis) a peak with this or a greater height (T- or Z-statistic), corrected (FWE or FDR)/ uncorrected for search volume.

- **cluster-level**: the chance (p) of finding a cluster with this many (ke) or a greater number of voxels, corrected (FWE or FDR)/ uncorrected for search volume.

- **set-level**: the chance (p) of finding this (c) or a greater number of clusters in the search volume.

It is also worth noting that:

- The table is surfable: clicking a row of cluster coordinates will move the pointer in the MIP to that cluster, clicking other numbers will display the exact value in the MATLAB window (e.g. 0.000 = 6.1971e-07).
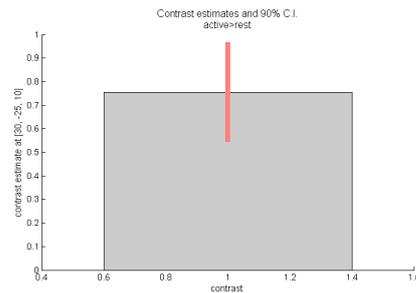
Figure 10.16: *Estimated effect size.*

- To inspect a specific cluster (e.g., in this example data set, the right auditory cortex), either move the cursor in the MIP (by left-clicking and dragging the cursor, or right-clicking the MIP background which will activate a pulldown menu).

- Alternatively, click the cluster coordinates in the volume table, or type the coordinates in the co-ordinates section of the interactive window.

It is also possible to produce tables of statistical information for a single cluster of interest rather than for the whole volume. Firstly, select the relevant cluster in the MIP and then press the "current cluster" button in the p-values section of the interactive window. This will show coordinates and voxel-level statistics for local maxima (>4mm apart) in the selected cluster. This table is also surfable.

10.3.8 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:

1. 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 10.16.

2. Fitted responses: Plots adjusted data and fitted response across session/subject. SPM will prompt for a specific contrast and provides the option to choose different ordinates ("an explanatory variable", "scan or time", or "user specified"). If "scan or time", the plot will show adjusted or fitted data with errors added as shown in Figure 10.17.

3. Event-related responses: Plots adjusted data and fitted response across peri-stimulus time.

4. Parametric responses.

5. Volterra kernels.

For plotting event-related responses SPM provides three options

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.
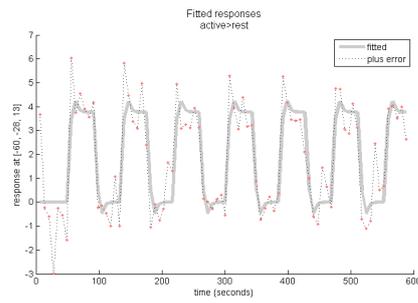
Figure 10.17: *Fitted responses.*
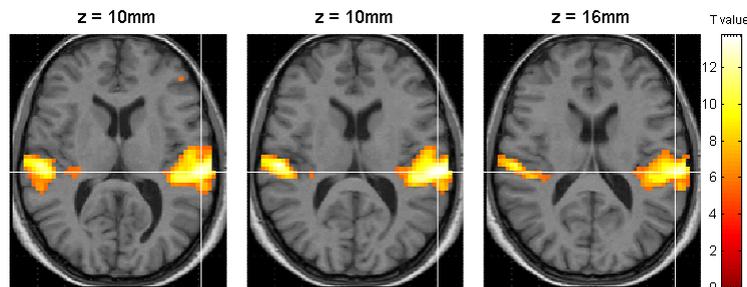
Figure 10.18: *Slices.*

10.3.9   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

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/hdr` in the working directory by using the *write filtered* option. In Figures 10.18, 10.19 and 10.20 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

- "Normalise (Write)" the two images `c1sM00223_002.img` and `c2sM00223_002.img` using the "Parameter File " `sM00223_002_seg_sn.mat` and a voxel size of [1 1 3].

- Selecting "Xtract Surface" from the "Render" pulldown menu.

- Selecting the gray and white matter images `wc1sM00223_002.img` and `wc2sM00223_002.img` created in the first step.

- Saving the results using the default options (Rendering and Surface).
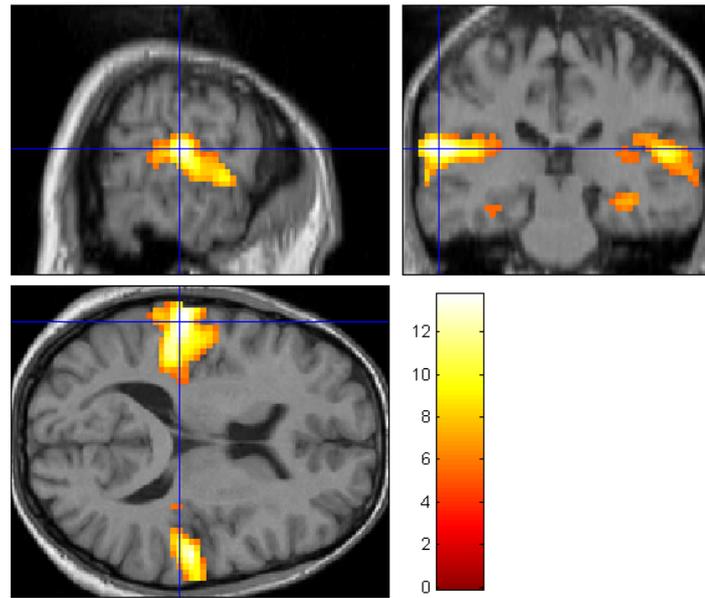
Figure 10.19: *Sections.*

SPM plots the rendered anatomical image in the graphics window and saves it as `render_wc1sM00223_002.mat`. The surface image is saved as `surf_wc1sM00223_002.mat`).

It is also possible to project and display the results on a surface mesh, we are going to use here one of the canonical mesh distributed with SPM (in MNI space). Press "Overlays" and choose "Render", then go in the `canonical` folder of your SPM installation and select file `cortex_20484.surf.gii` (this is a surface mesh stored using the GIfTI format) and you will obtain a figure similar to 10.21.

10.3.10 Miscellaneous

Other options (in the results controls panel):

- **clear**: clears lower subpanel of Graphics window,

- **exit**: exits the results section,

- **?**: launches help.

10.4 Bayesian analysis

10.4.1 Specification

Press the "Specify 1st-level" button. This will call up an fMRI specification job in the batch editor. 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).
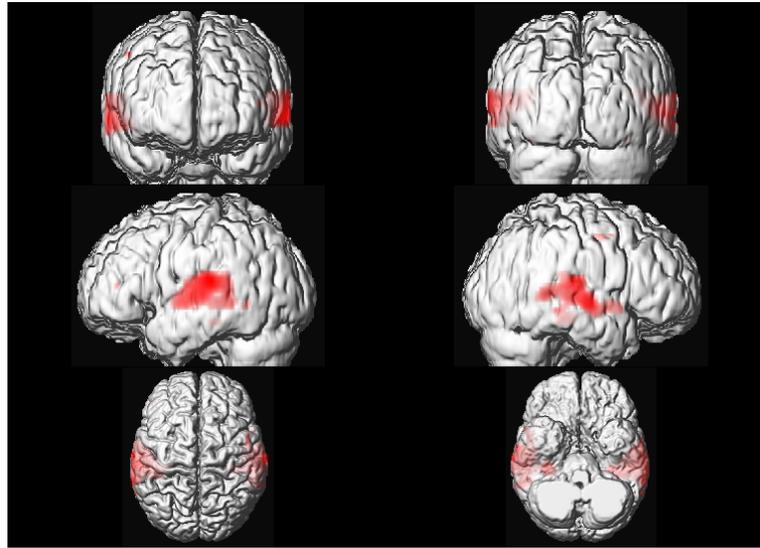
Figure 10.20: *Render.*

Render (Acrobat Reader required)

Figure 10.21: *3D Rendering using canonical mesh.*

- 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)[5]. 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/classical` directory).

- Save the job as `specify_bayesian.mat` and press the "Run" button.

10.4.2 Estimation

Press the "Estimate" button. This will call up the specification of an fMRI estimation job in the batch editor. Then

- Highlight the "Select SPM.mat" option and then choose the SPM.mat file saved in the `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 the "Run" butto.

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.

10.4.3 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,

---

[5]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.

- *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 10.22.

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[6],

- *Extent threshold [0],*

- Accept the default value, 0,

- *Plot effect size [Yes/No],*

- Select the default "Yes".

SPM will then plot a map of χ^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 10.23

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").

---

[6]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.
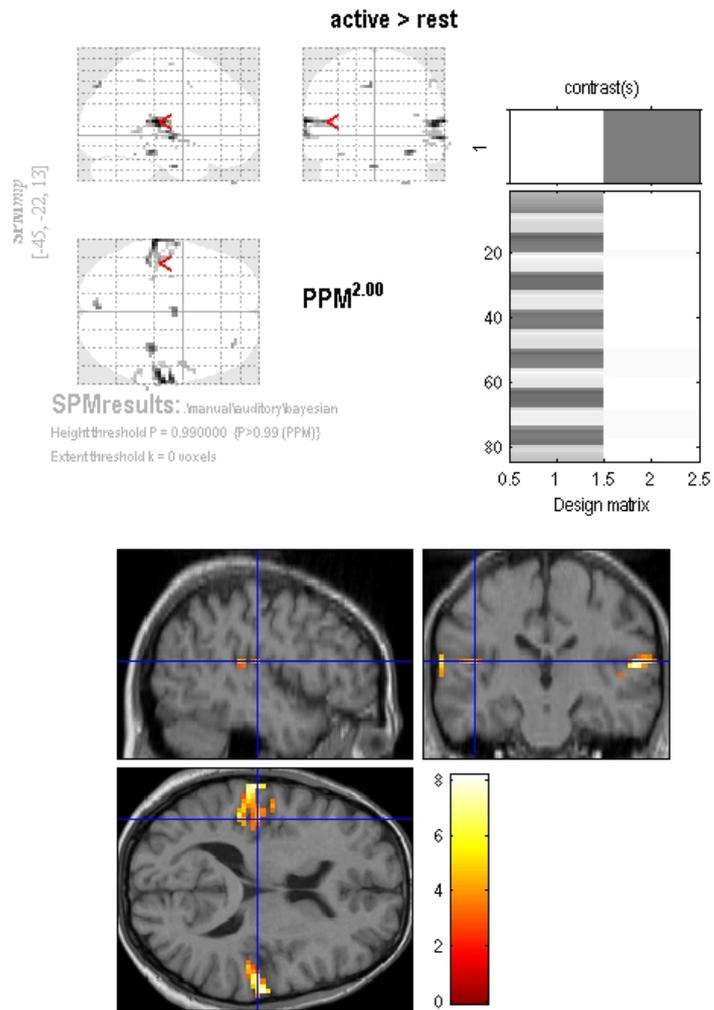
Figure 10.22: **Bayesian analysis:** *MIP and overlay of effect sizes at voxels where SPM is 99% sure that the effect size is greater than 2% of the global mean.*
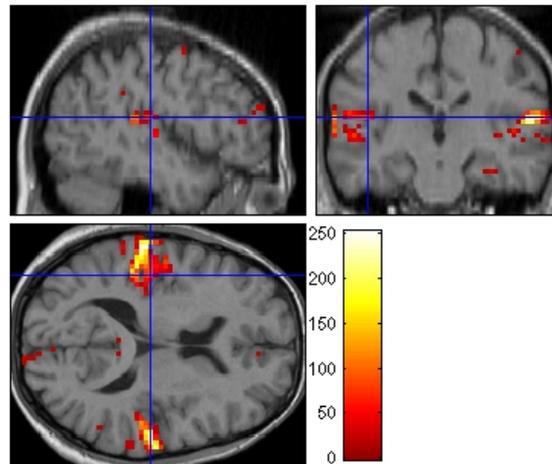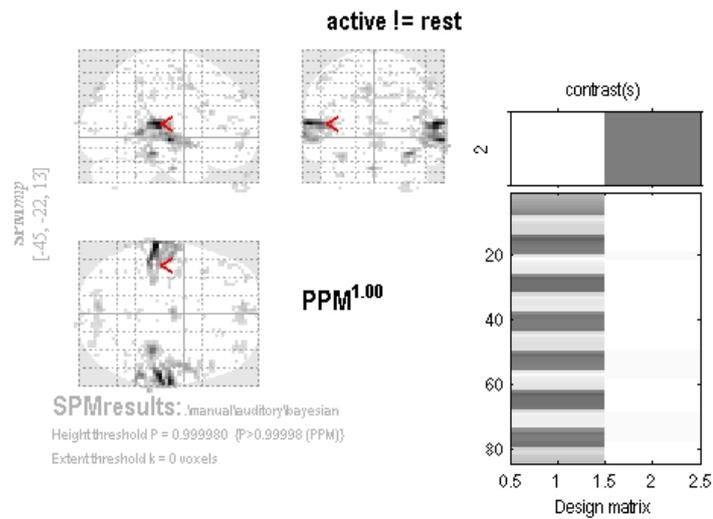
Figure 10.23: **Two-sided Bayesian analysis:** MIP and overlay of χ^2 statistic values at above threshold voxels. This shows regions where activity is different between active and rest conditions, whether positive or negative.

Chapter 11

Face fMRI data

As another, more sophisticated example, consider the data from a repetition priming experiment performed using event-related fMRI. Briefly, this is a 2×2 factorial study with factors "fame" and "repetition" where famous and non-famous faces were presented twice against a checkerboard baseline (for more details, see [36]). 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 11.1 and 11.2.

Images were acquired using continuous Echo-Planar Imaging (EPI) with TE=40ms, TR=2s and 24 descending slices (64×64 3×3 mm^2), 3mm thick with a 1.5mm gap. The data archive is available from the SPM website[1]. This contains 351 Analyze format functional images `sM03953_0005_*.img` of dimension 64×64×24 with 3×3×4.5 mm^3 voxels. A structural image is also provided in Analyze format (`sM03953_0007.img`).

To analyse the data, first create a new directory `DIR` eg. `C:\data\face_rep`, in which to place the results of your analysis. Then create 4 subdirectories (i) `jobs`, (ii) `categorical`, (iii) `parametric` and (iv) `bayesian`. As the analysis proceeds these directories will be filled with job-specification files, design matrices and models estimated using classical or Bayesian methods.

As well as the classical/Bayesian distinction we will show how this data can be analysed from a parametric as well as a categorical perspective. We will look at the main effects of fame and repetition and in the parameteric analysis we will look at responses as a function of "lag", that is, the number of faces intervening between repetition of a specific face.

Start up matlab, enter your jobs directory and type `spm fmri` at the MATLAB prompt. SPM will then open in fMRI mode with three windows (1) the top-left or "Menu" window, (2) the bottom-left or "Interactive" window and (3) the right-hand or "Graphics" window. Analysis then takes place in three major stages (i) spatial pre-processing, (ii) model specification, review and estimation and (iii) inference. These stages organise the buttons in SPM's base window.

11.1 Spatial pre-processing

11.1.1 Display

Display eg. the first functional image using the "Display" button. Note orbitofrontal and inferior temporal drop-out and ghosting. This can be seen more clearly by selecting "brighten" from the "Effects" tab in the "Colours" at the top of the Graphics window.

11.1.2 Realignment

Under the spatial pre-processing section of the SPM base window select REALIGN (EST & RES) from the REALIGN pulldown menu. This will call up a realignment job specification in the batch editor window. Then

- Highlight data, select "New Session", then highlight the newly created "Session" option.

---

[1]Face Repetition dataset: http://www.fil.ion.ucl.ac.uk/spm/data/face_rep/
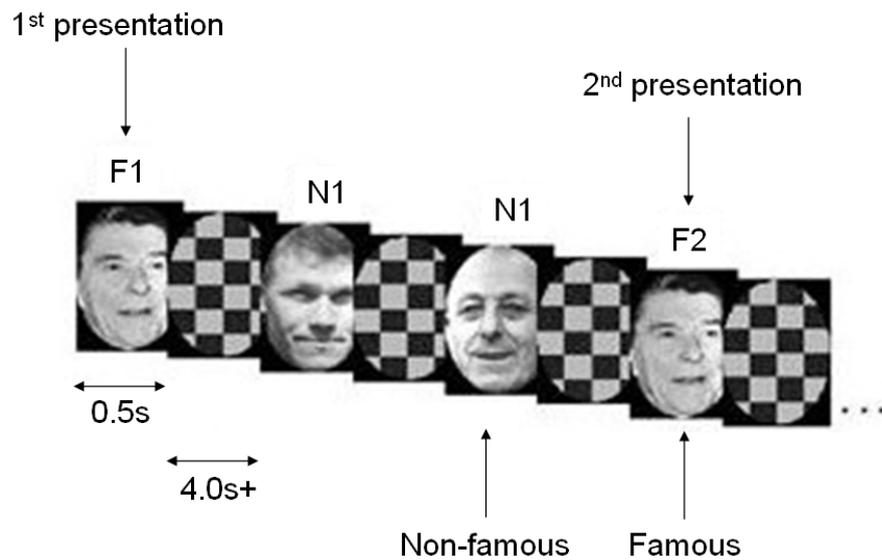
Figure 11.1: **Face repetition paradigm**: *There were 2 presentations of 26 Famous and 26 Nonfamous Greyscale photographs, for 0.5s each, randomly intermixed. The minimal Stimulus Onset Asynchrony (SOA)=4.5s, with probability 2/3 (ie 1/3 null events). The subject made one of two right finger key presses denoting whether or not the subject thought the face was famous.*
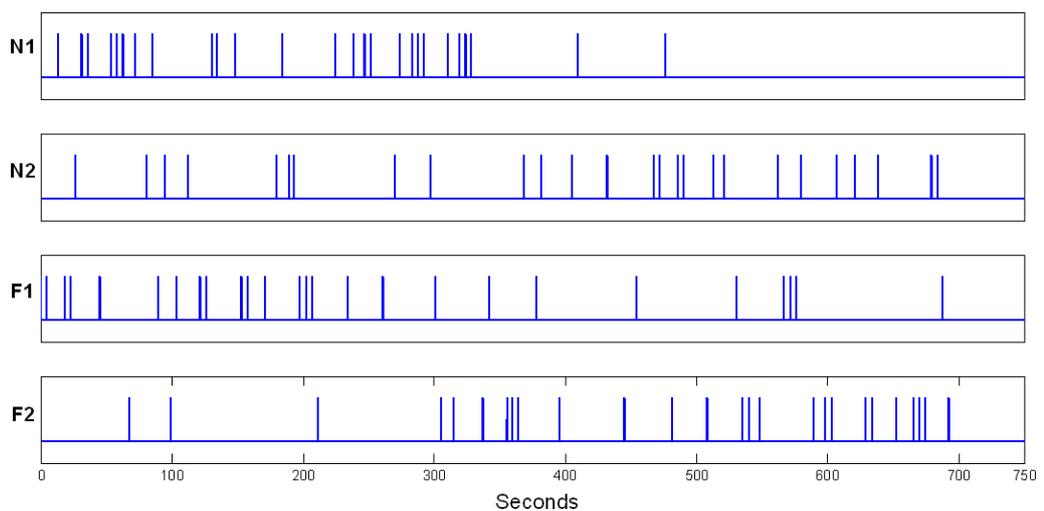
Figure 11.2: *Time series of events.*
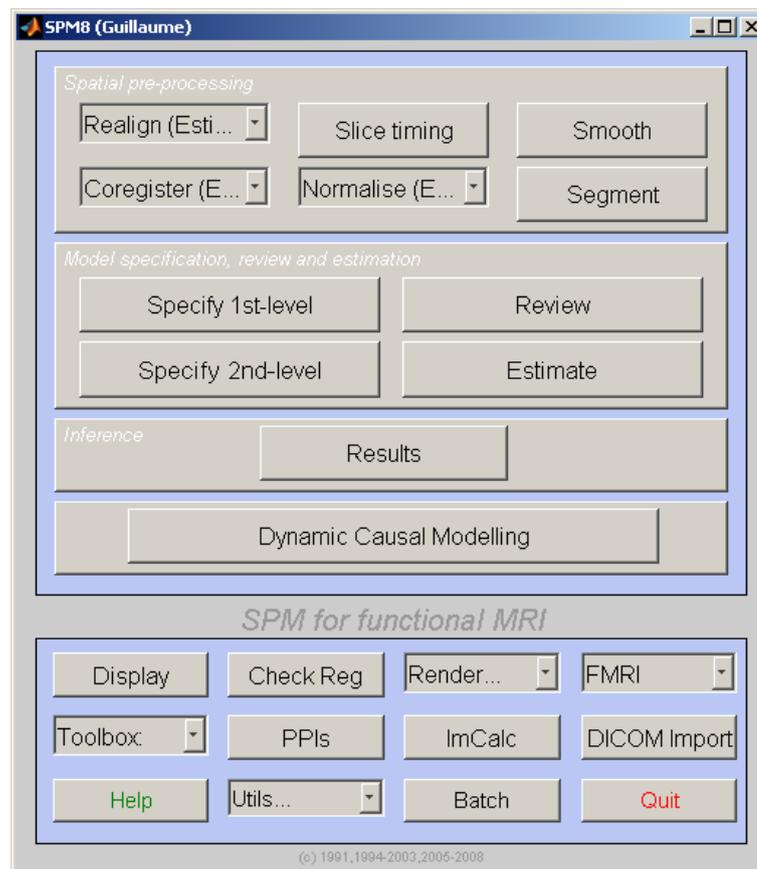
Figure 11.3: *The SPM base window comprises three sections (i) spatial pre-processing, (ii) model specification, review and estimation and (iii) inference.*
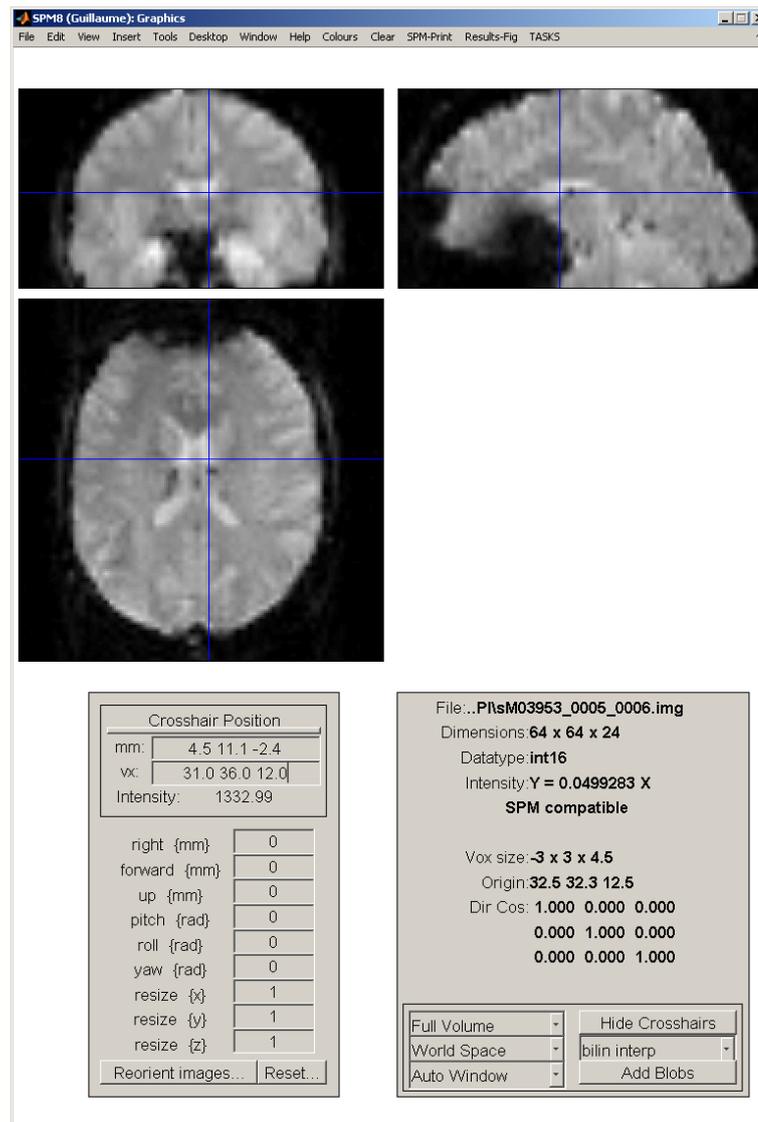
Figure 11.4: *Signal dropout in EPI images.*
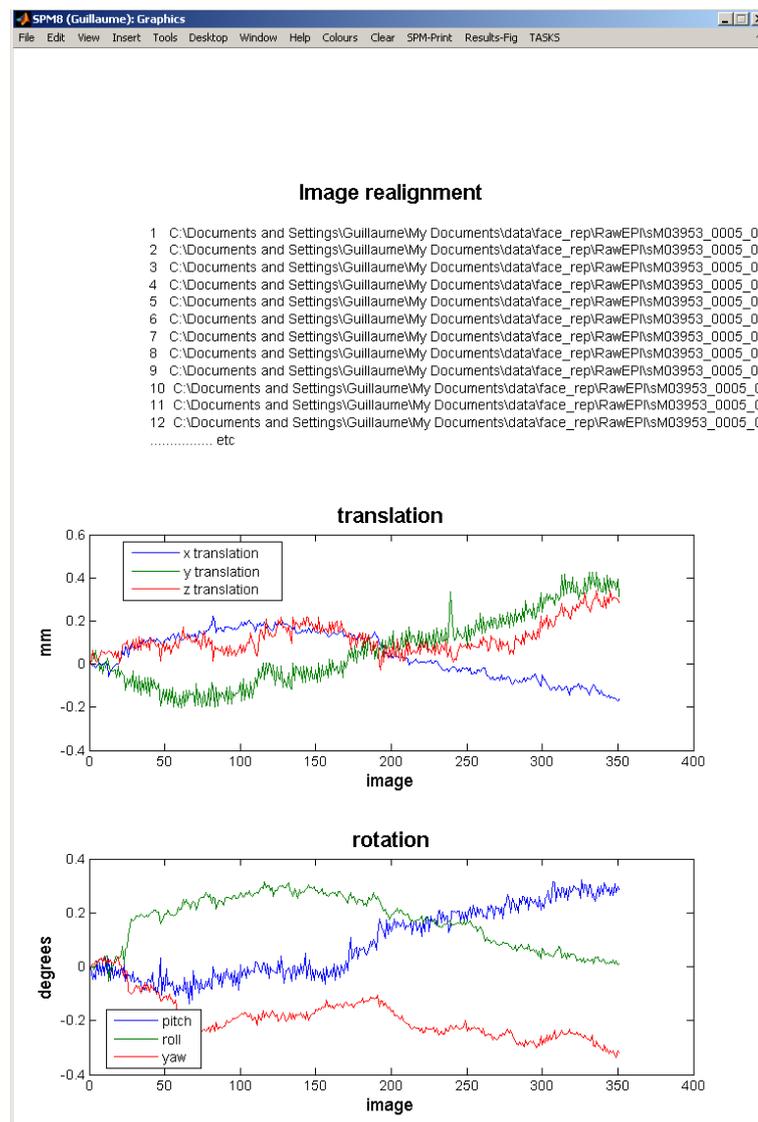
Figure 11.5: **Realignment of face data**: *Movement less than the size of a voxel, which for this data set is 3mm, is not considered problematic.*

- 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 batch editor window (green triangle).

This will run the realign job which will write realigned images into the directory where the functional images are. These new images will be prefixed with the letter "`r`". SPM will then plot the estimated time series of translations and rotations shown in Figure 11.5. 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.

11.1.3   Slice timing correction

Press the SLICE TIMING button. This will call up the specification of a slice timing job in the batch editor window. Note that these data consist of N=24 axial slices acquired continuously with a TR=2s (ie TA = TR - TR/N, where TA is the time between the onset of the first and last slice of one volume, and the TR is the time between the onset of the first slice of one volume and the first slice of next volume) and in a descending order (ie, most superior slice was sampled first). The data however are ordered within the file such that the first slice (slice number 1) is the most inferior slice, making the slice acquisition order [24 23 22 ... 1].

- Highlight "Data" and select "New Sessions"

- Highlight the newly create "Sessions" option, "Specify Files" and select the 351 realigned functional images using the filter ^r.*.

- Select "Number of Slices" and enter 24.

- Select TR and enter 2.

- Select TA and enter 1.92 (or 2 - 2/24).

- Select "Slice order" and enter 24:-1:1.

- Select "Reference Slice", and enter 12.

- Save the job as slice_timing.mat and press the "Run" button.

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

11.1.4   Coregistration

Select COREGISTER (ESTIMATE) from the Coregister pulldown menu. This will call up the specification of a coregistration job in the batch editor window.

- Highlight "Reference Image" and then select the mean functional image meansM03953_0005_0006.img.

- Highlight "Source Image" and then select the structural image eg. sM03953_0007.img.

- Press the "Save" button and save the job as coreg.job

- Then press the "Run" button.

SPM will then implement a coregistration between the structural and functional data that maximises the mutual information. The image in figure 11.6 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.

11.1.5   Segmentation

Press the SEGMENT button. This will call up the specification of a segmentation job in the batch editor 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 the Run button. SPM will segment the structural image using the default tissue probability maps as priors. SPM will create, by default, gray and white matter images and bias-field corrected structral image. These can be viewed using the CheckReg facility as described in the previous section. Figure 11.7 shows the gray matter image, c1sM03953_0007.img, along with the original structural[2].

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.

---

[2]Segmentation can sometimes fail if the source (structural) image is not close in orientation to the MNI templates. It is generally advisable to manually orient the structural to match the template (ie MNI space) as close as possible by using the "Display" button, adjusting x/y/z/pitch/roll/yaw, and then pressing the "Reorient" button.
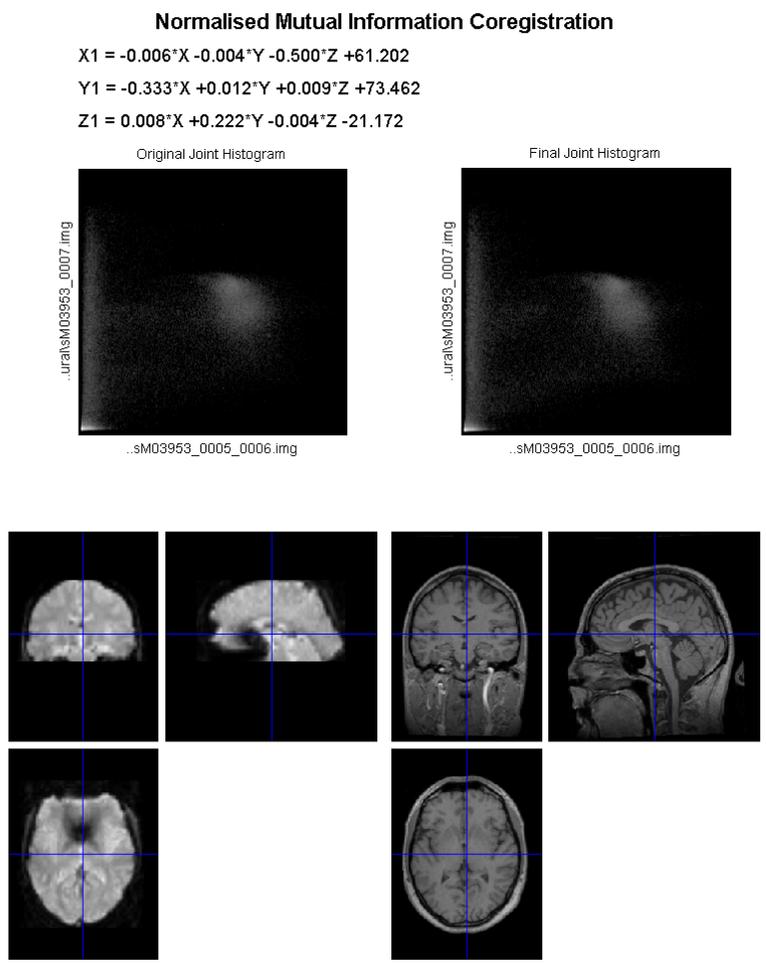
Figure 11.6: *Mutual Information Coregistration of Face data.*
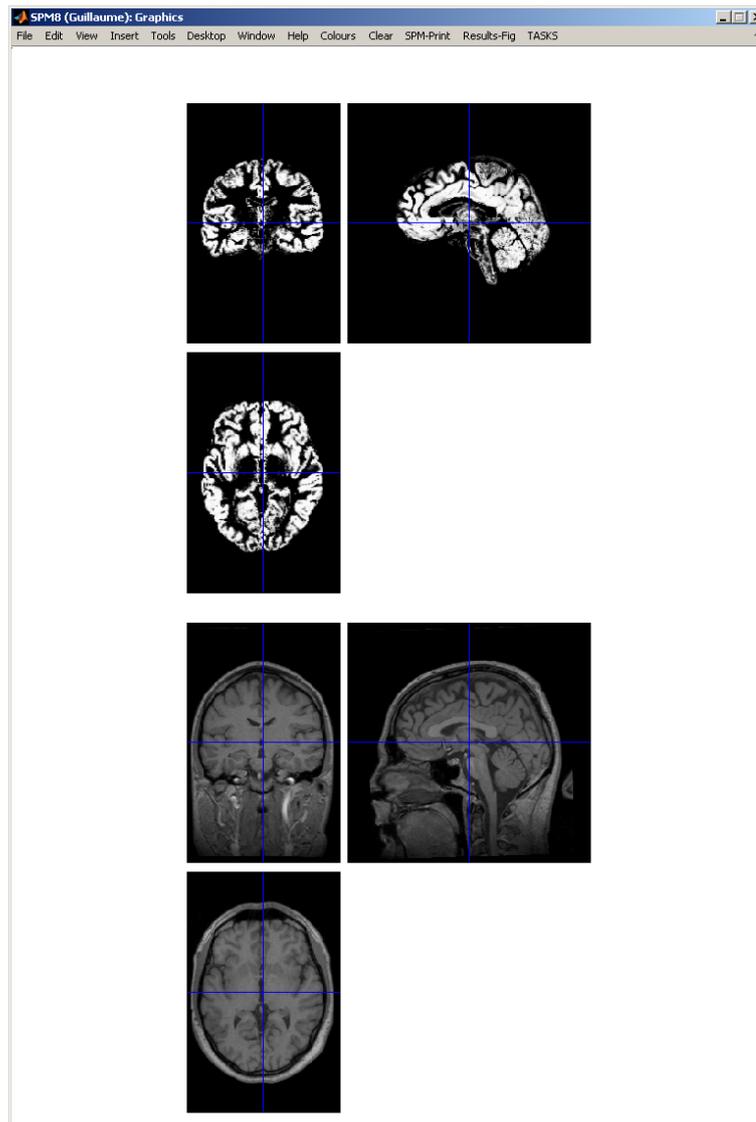
Figure 11.7: *Gray matter (top) produced by segmentation of structural image (below).*

11.1.6 Normalise

Select NORMALISE (WRITE) from the NORMALISE pulldown menu. This will call up the specification of a normalise job in the batch editor window.

- Highlight "Data", select "New Subject".

- Open "Subject", highlight "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][3].

- Press "Save", save the job as normalise.mat and then press the `Run` button.

SPM will then write spatially normalised files to the functional data directory. These files have the prefix "`w`".

If you wish to superimpose a subject's functional activations on their own anatomy[4] you will also need to apply the spatial normalisation parameters to their (bias-corrected) anatomical image. To do this

- Select NORMALISE (WRITE), highlight 'Data', select "New Subject".

- Highlight "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` button.

11.1.7 Smoothing

Press the SMOOTH button[5]. This will call up the specification of a smooth job in the batch editor window.

- Select "Images to Smooth" and then select the spatially normalised files created in the last section eg. `war*.img`.

- Save the job as `smooth.mat` and press `Run` button.

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

---

[3]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.

[4]Beginners may wish to skip this step, and instead just superimpose functional activations on an "canonical structural image".

[5]The smoothing step is unnecessary if you are only interested in Bayesian analysis of your functional data.
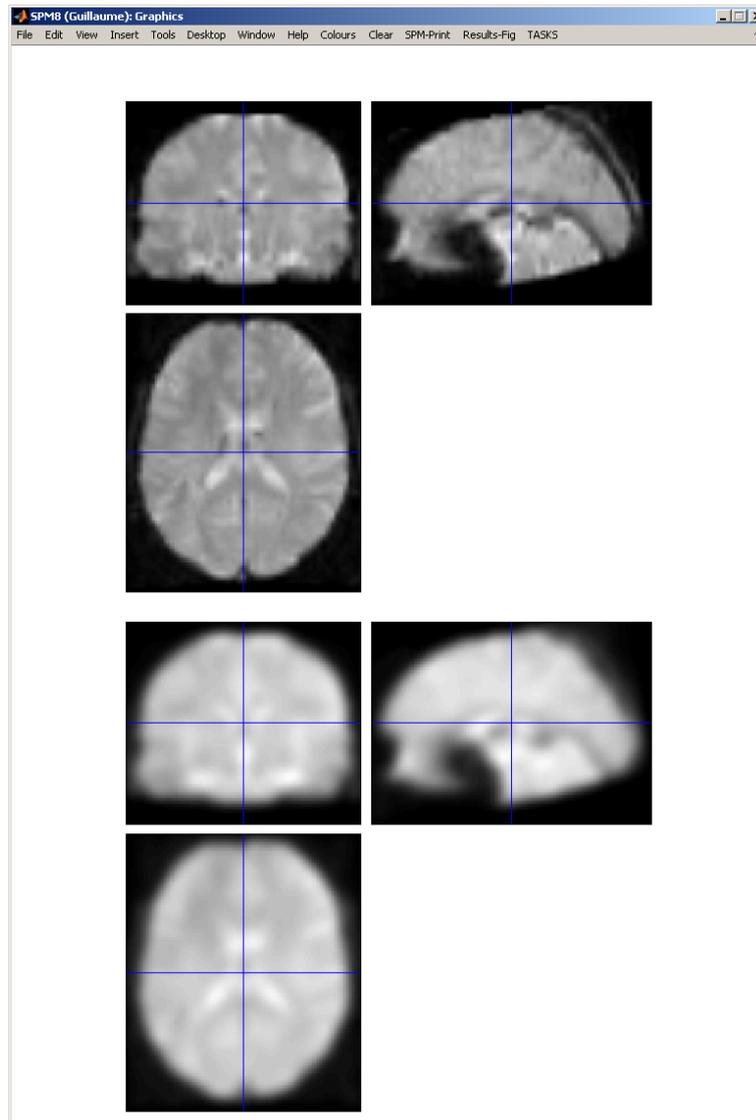
Figure 11.8: *Functional image (top) and 8mm-smoothed functional image (bottom). These images were plotted using SPM's "CheckReg" facility.*

11.2 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.[6]

- 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 batch editor window. Then

- In the "Timing paramaters" option,

- Highlight "Units for design" and select "Scans",

- Highlight "Interscan interval" and enter 2,

- Highlight "Microtime resolution" and enter 24,

- Highlight "Microtime onset" and enter 12. These last two options make the creating of regressors commensurate with the slice-time correction we have applied to the data, given that there are 24 slices and that the reference slice to which the data were slice-time corrected was the 12th (middle slice in time).

- Highlight "Data and Design" and select "New Subject/Session".

- Highlight "Scans" and use SPM's file selector to choose the 351 smoothed, normalised, slice-time corrected, realigned functional images ie `swarsM.img`. These can be selected easily using the `^swar.*` filter, and select all. Then press "Done".

- Highlight "Conditions" and select "New condition"[7].

- Open the newly created "Condition" option. Highlight "Name" and enter "N1". Highlight "Onsets" and enter `sot{1}`. Highlight "Durations" and enter 0.

- Highlight "Conditions" and select "Replicate condition".

- Open the newly created "Condition" option (the lowest one). Highlight "Name" and change to "N2". Highlight "Onsets" and enter `sot{2}`.

- Highlight "Conditions" and select "Replicate condition".

- Open the newly created "Condition" option (the lowest one). Highlight "Name" and change to "F1". Highlight "Onsets" and enter `sot{3}`.

- Highlight "Conditions" and select "Replicate condition".

- Open the newly created "Condition" option (the lowest one). Highlight "Name" and change to "F2". Highlight "Onsets" and enter `sot{4}`.

- Highlight "Multiple Regressors" and select the `movepars.txt` file[8].

- Highlight "Factorial Design", select "New Factor", open the newly created "Factor" option, highlight "Name" and enter "Fam", highlight "Levels" and enter 2.

---

[6]Unlike previous analyses of these data in SPM99 and SPM2, we will not bother with extra event-types for the (rare) error trials.

[7]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.

[8]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.
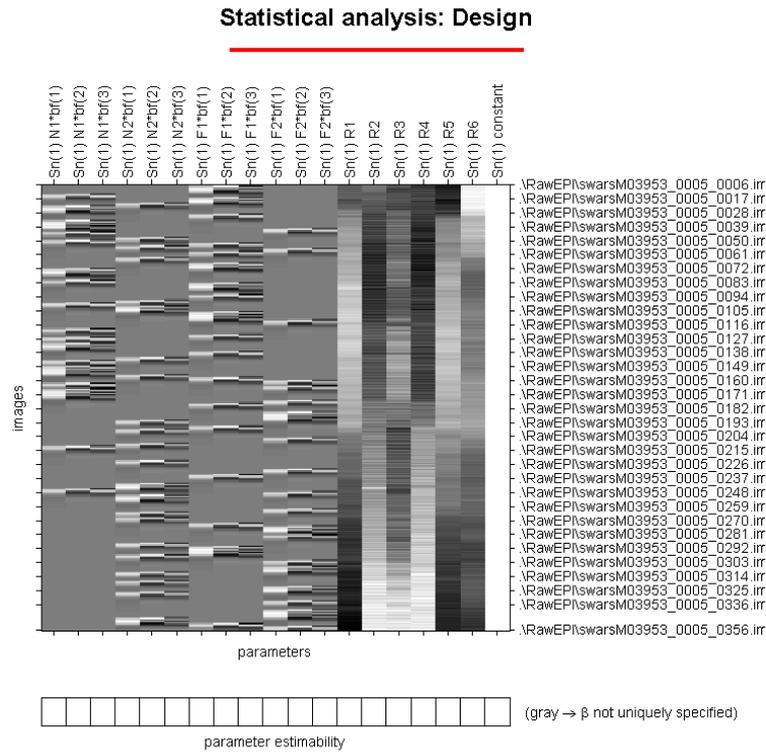
Statistical analysis: Design

Figure 11.9: *Design matrix.*

- Highlight "Factorial Design", select "New Factor", open the newly created "Factor" option, highlight "Name" and enter "Rep", highlight "Levels" and enter 2[9].

- Open "Canonical HRF" under "Basis Functions". Select "Model derivatives" and select "Time and Dispersion derivatives".

- Highlight "Directory" and select the `DIR/categorical` directory you created earlier.

- Save the job as `categorical_spec.mat` and press the `Run` button.

SPM will then write an `SPM.mat` file to the `DIR/categorical` directory. It will also plot the design matrix, as shown in Figure 11.9.

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 window clicking on which produces a pulldown menu. If you select the first item "Design Matrix" SPM will produce the image shown in Figure 11.9. If you select "Explore" then "Session 1" then "N1", SPM will produce the plots shown in Figure 11.10.

---

[9]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.
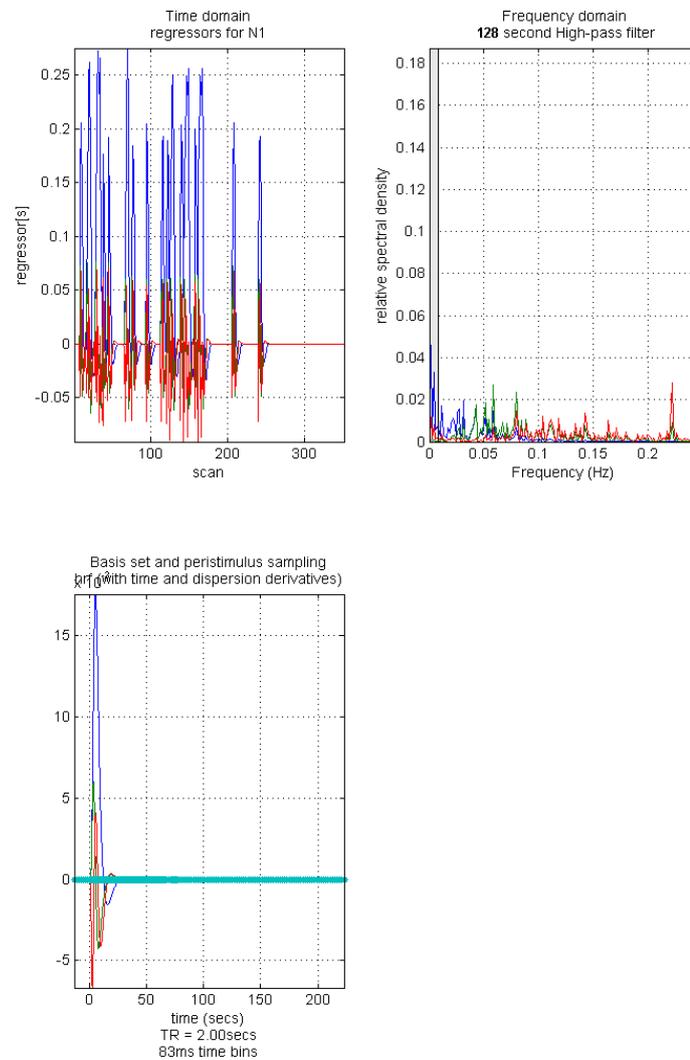
Figure 11.10: **_Exploring the design matrix in Figure 11.9_**. *This shows the time series of the "N1" regressor (top left), the three basis functions used to convert assumed neuronal activity into hemodynamic activity (bottom left), and a frequency domain plot of the three regressors for the basis functions in this condition (top right). The frequency domain plot shows that the frequency content of the "N1" condition is generally above the set frequencies that are removed by the High Pass Filter (HPF) (these are shown in gray - in this model we accepted the default HPF cut-off of 128s or 0.008Hz).*
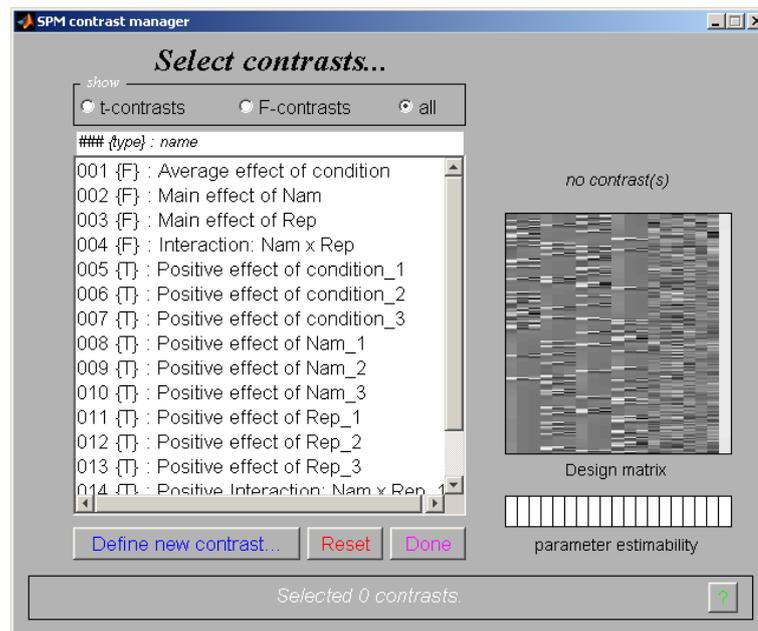
Figure 11.11: *Contrast Manager containing default contrasts for categorical design.*

11.2.1   Estimate

Press the ESTIMATE button. This will call up the specification of an fMRI estimation job in the batch editor window. Then

- Highlight the "Select SPM.mat" option and then choose the `SPM.mat` file saved in the `DIR/categorical` directory.

- Save the job as `categorical_est.job` and press `Run` button.

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

11.2.2   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 11.11.

- 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
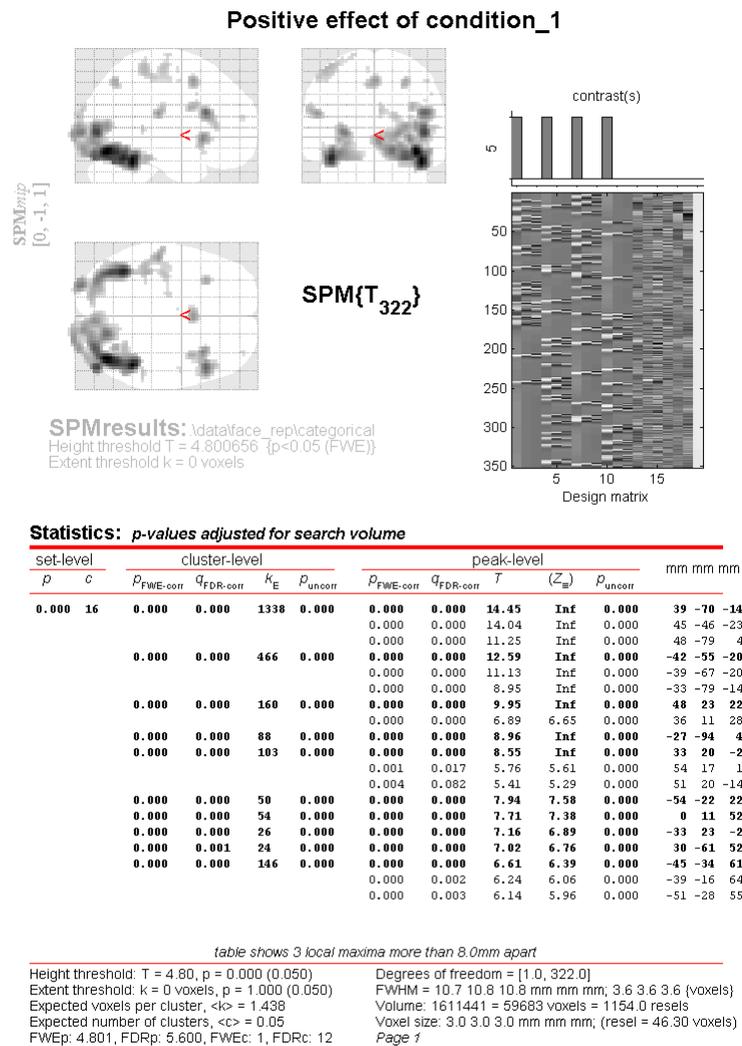
Figure 11.12: **MIP and Volume table for Canonical HRF**: *Faces > Baseline.*

- *Extent threshold {voxels} [0]*

- Accept the default value, 0.

SPM will then produce the MIP shown in Figure 11.12.

11.2.3   Statistical tables

To get a summary of local maxima, press the "whole brain" button in the p-values section of the interactive window. This will list all clusters above the chosen level of significance as well as separate (>8mm apart) maxima within a cluster, with details of significance thresholds and search volume underneath, as shown in Figure 11.12

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

- **x, y, z (mm)**: coordinates in MNI space for each maximum.

- **peak-level**: the chance (p) of finding (under the null hypothesis) a peak with this or a greater height (T- or Z-statistic), corrected (FWE or FDR)/ uncorrected for search volume.

- **cluster-level**: the chance (p) of finding a cluster with this many(ke) or a greater number of voxels, corrected (FWE or FDR)/ uncorrected for search volume.

- **set-level**: the chance (p) of finding this (c) or a greater number of clusters in the search volume.

Right-click on the MIP and select "goto global maximum". The cursor will move to [39 -70 -14]. You can view this activation on the subject's normalised, attenuation-corrected structural (`wmsM03953_0007img`), 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).

11.2.4   F-contrasts

To assess the main effect of repeating faces, as characterised by both the hrf *and* its derivatives, an F-contrats is required. This is really asking whether repetition changes the *shape* of the impulse response (e.g, it might affect its latency but not peak amplitude), at least the range of shapes defined by the three basis functions. Because we have told SPM that we have a factorial design, this required contrast will have been created automatically - it is number 3.

- Press "Results" and select the `SPM.mat` file in the `DIR/categorical` directory.

- Select the "F-contrast" toggle and the contrast number 3, as shown in Figure 11.13. Press "Done".

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

- Specify "Yes".

- 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/none]*

- Select none

- *threshold (F or p value)*

- Accept the default value, 0.001

- *Extent threshold {voxels} [0]*

- Accept the default value, 0
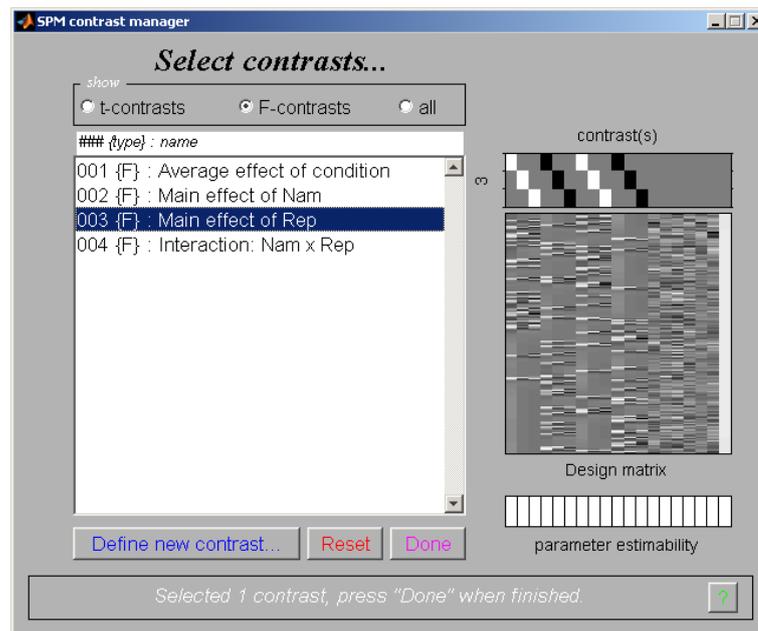
Figure 11.13: *Contrast manager showing selection of the first contrast "Main effect of Rep" (repetition: F1 and N1 vs F2 and N2)*

A MIP should then appear, the top half of which should look like Figure 11.14.

Note that this contrast will identify regions showing any effect of repetition (e.g, decreased or increased amplitudes) *within* those regions showing activations (on the canonical HRF) to faces versus baseline (at p<.05 uncorrected). Select "goto global max", which is in right ventral temporal cortex [42 -64 -8].

If you press plot and select "Event-related responses", then "F1", then "fitted response and PSTH", you will see the best fitting linear combination of the canonical HRF and its two derivatives (thin red line), plus the "selectively-averaged" data (peri-stimulus histogram, PSTH), based on an FIR refit (see next Chapter). If you then select the "hold" button on the Interactive window, and then "plot" and repeat the above process for the "F2" rather than "F1" condition, you will see two estimated event-related responses, in which repetition decreases the peak response (ie F2<F1), as shown in Figure 11.14.

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 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.

11.2.5   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,
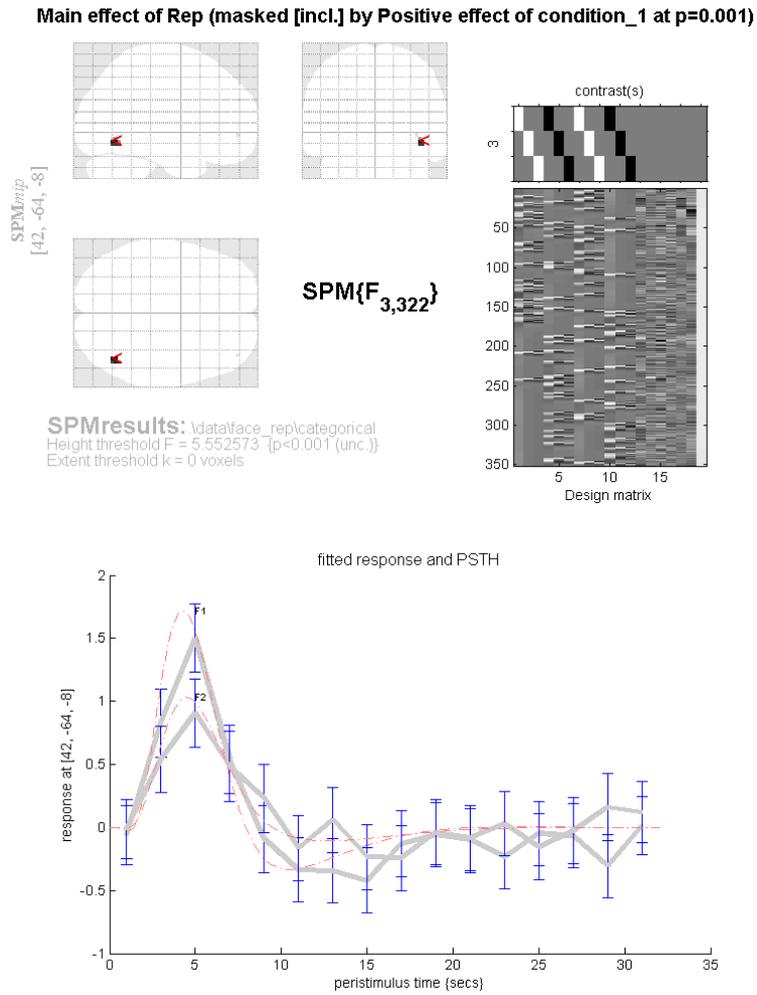
Figure 11.14: *MIP for Main effect of Rep, masked inclusively with Canonical HRF: Faces >
Baseline at p<.001 uncorrected. Shown below are the best-fitting responses and peri-stimulus
histograms (PSTH) for F1 and F2.*

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 11.15). (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 datae 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.

11.3 Modelling parametric responses

Before setting up the design matrix, we must first load into MATLAB the Stimulus Onsets Times (SOTs), as before, and also the "Lags", which are specific to this experiment, and which will be used as parametric modulators. The Lags code, for each second presentation of a face (N2 and F2), the number of other faces intervening between this (repeated) presentation and its previous (first) presentation. Both SOTs and Lags are represented by Matlab cell arrays, stored in the `sots.mat` file.

- At the MATLAB command prompt type `load sot`. This loads the stimulus onset times and the lags (the latter in a cell array called `itemlag`.

Now press the SPECIFY 1ST-LEVEL button. This will call up the specification of a fMRI specification job in the batch editor window. Then

- Press "Load" and select the `categorical_spec.mat` job file you created earlier.

- Open "Conditions" and then open the second "Condition".

- Highlight "Parametric Modulations", select "New Parameter".

- Highlight "Name" and enter "Lag", highlight values and enter `itemlag{2}`, highlight polynomial expansion and "2nd order".

- Now open the fourth "Condition" under "Conditions".

- Highlight "Parametric Modulations", select "New Parameter".

- Highlight "Name" and enter "Lag", highlight values and enter `itemlag{4}`, highlight polynomial expansion and "2nd order".

- Open "Canonical HRF" under "Basis Functions", highlight "Model derivatives" and select "No derivatives" (to make the design matrix a bit simpler for present purposes!).

- Highlight "Directory" and select `DIR/parametric` (having "unselected" the current definition of directory from the Categorical analysis).

- Save the job as `parametric_spec` and press the `Run` button.

This should produce the design matrix shown in Figure 11.16.
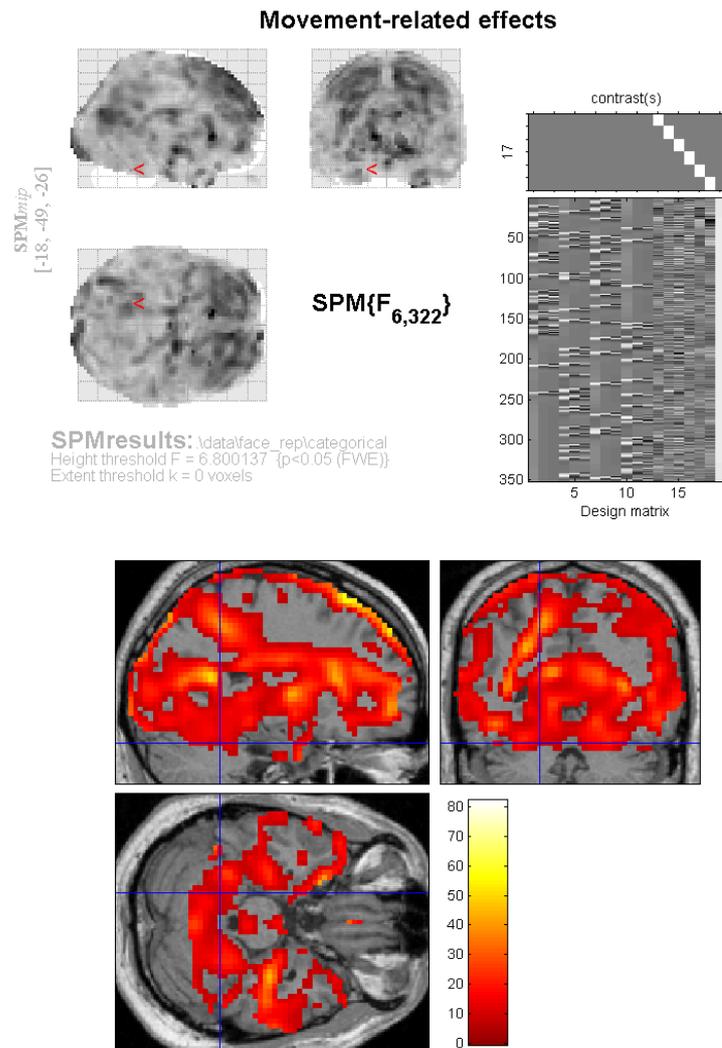
Figure 11.15: *Movement-related activations. These spurious 'activations' are due to residual movement of the head during scanning. These effects occur at tissue boundaries and boundaries between brain and non-brain, as this is where contrast differences are greatest. Including these regressors in the design matrix means these effects cannot be falsely attributed to neuronal activity.*
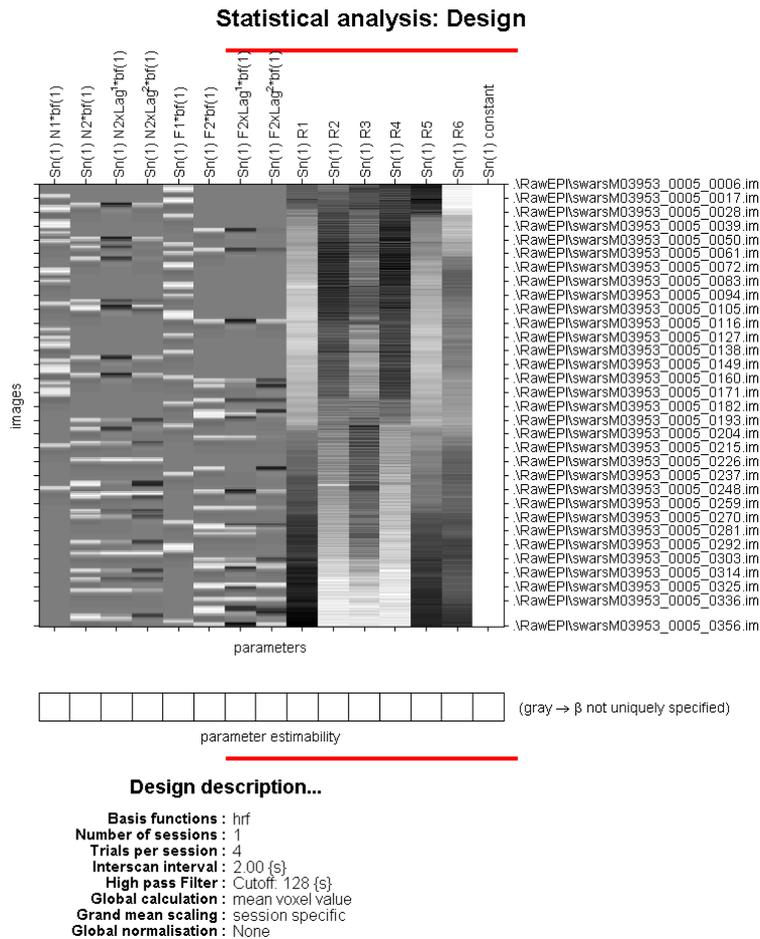
Figure 11.16: **Design matrix for testing repetition effects parametrically.** *Regressor 2 indicates the second occurence of a nonfamous face. Regressor 3 modulates this linearly as a function of lag (ie. how many faces have been shown since that face was first presented), and regressor 4 modulates this quadratically as a function of lag. Regressors 6,7 and 8 play the same roles, but for famous faces.*

11.3.1   Estimate

Press the ESTIMATE button. This will call up the specification of an fMRI estimation job in the batch editor window. Then

- Highlight the "Select SPM.mat" option and then choose the `SPM.mat` file saved in the `DIR/parametric` directory.

- Save the job as `parametric_est.job` and press the `Run` button.

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

11.3.2   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 ?*

- Accept what is offered

- *p value adjustment to control: [FWE/none]*

- Select None

- *Threshold {F or p value}*

- Accept the default value, 0.001

- *Extent threshold {voxels} [0]*

- Accept the default value, 0.

Figure 11.17 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 11.18. This was obtained by

- Right clicking on the MIP and selecting "global maxima".

- Pressing Plot, and selecting "parametric responses" from the pull-down menu.

- Which effect ? select "F2".

This shows a quadratic effect of lag, in which the response appears negative for short-lags, but positive and maximal for lags of about 40 intervening faces (note that this is a very approximate fit, since there are not many trials, and is also confounded by time during the session, since longer lags necessarily occur later (for further discussion of this issue, see the SPM2 example analysis of these data on the webpage).
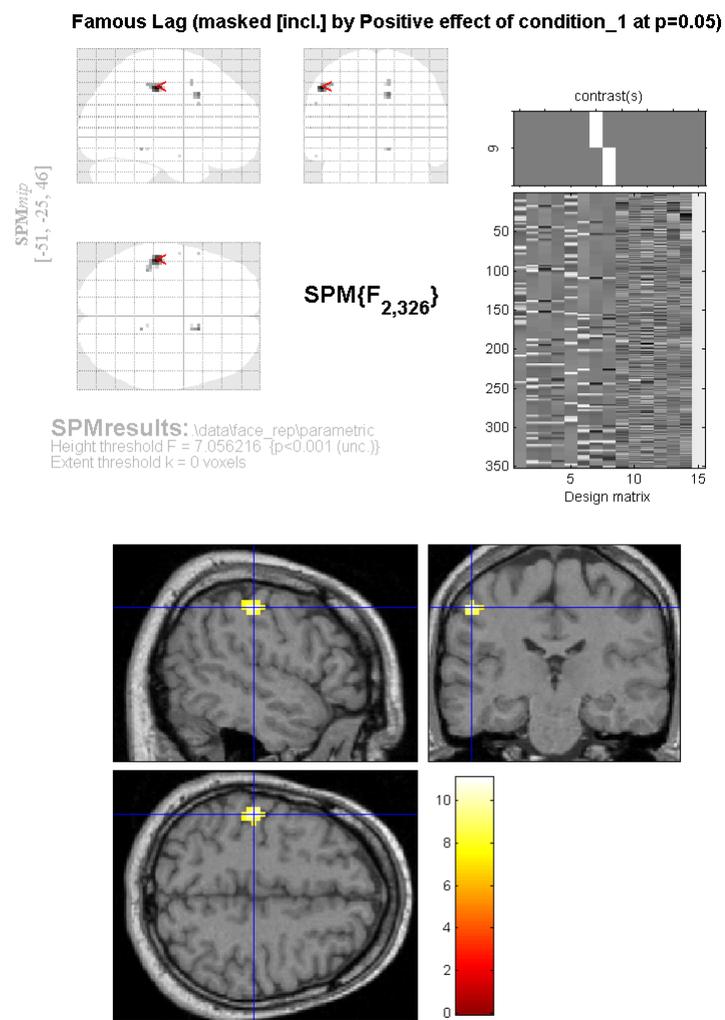
Figure 11.17: *MIP and overlay of parametric lag effect in parietal cortex.*
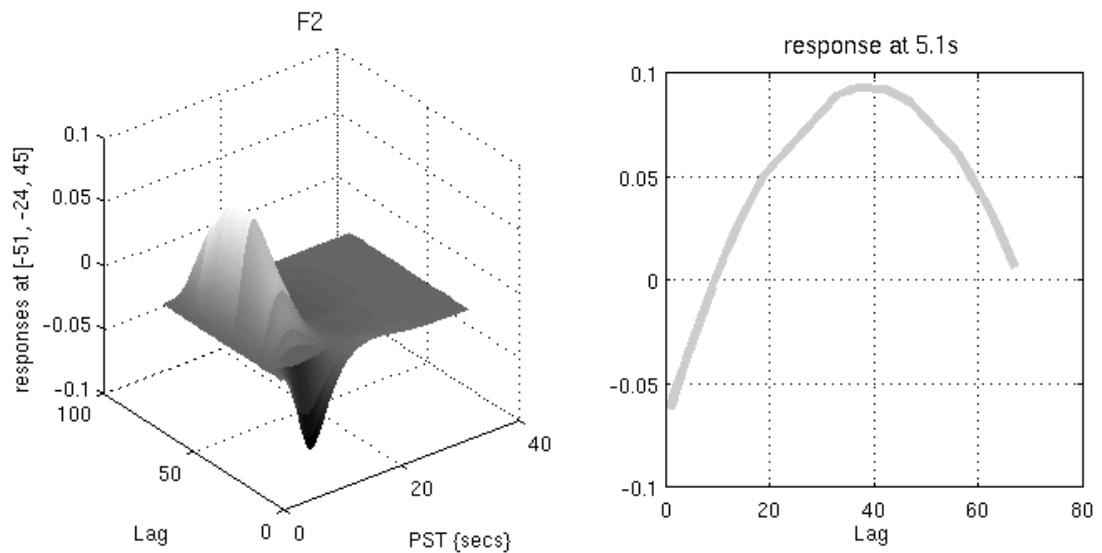
Figure 11.18: *Response as a function of lag.*

11.4   Bayesian analysis

11.4.1   Specification

Press the SPECIFY 1ST-LEVEL button. This will call up an fMRI specification job in the batch editor window. Then

- Load the `categorical_spec.mat` job file created for the classical analysis.

- Open "Subject/Session", highlight "Scans".

- Deselect the smoothed functional images using the 'unselect all' option available from a right mouse click in the SPM file selector (bottom window).

- Select the unsmoothed functional images using the `^wa.*` filter and "select all" option available from a right mouse click in the SPM file selector (top right window). The Bayesian analysis uses a spatial prior where the spatial regularity in the signal is estimated from the data. It is therefore not necessary to create smoothed images if you are only going to do a Bayesian analysis.

- Press "Done".

- Highlight "Directory" and select the `DIR/bayesian` directory you created earlier (you will first need to deselect the `DIR/categorical` directory).

- Save the job as `specify_bayesian.mat` and press the `Run` button.

11.4.2   Estimation

Press the ESTIMATE button. This will call up the specification of an fMRI estimation job in the batch editor window. Then

- Highlight the "Select SPM.mat" option and then choose the `SPM.mat` file saved in the `DIR/bayesian` subdirectory

- Highlight "Method" and select the "Choose Bayesian 1st-level" option.

- Save the job as `estimate_bayesian.job` and press the `Run` button.
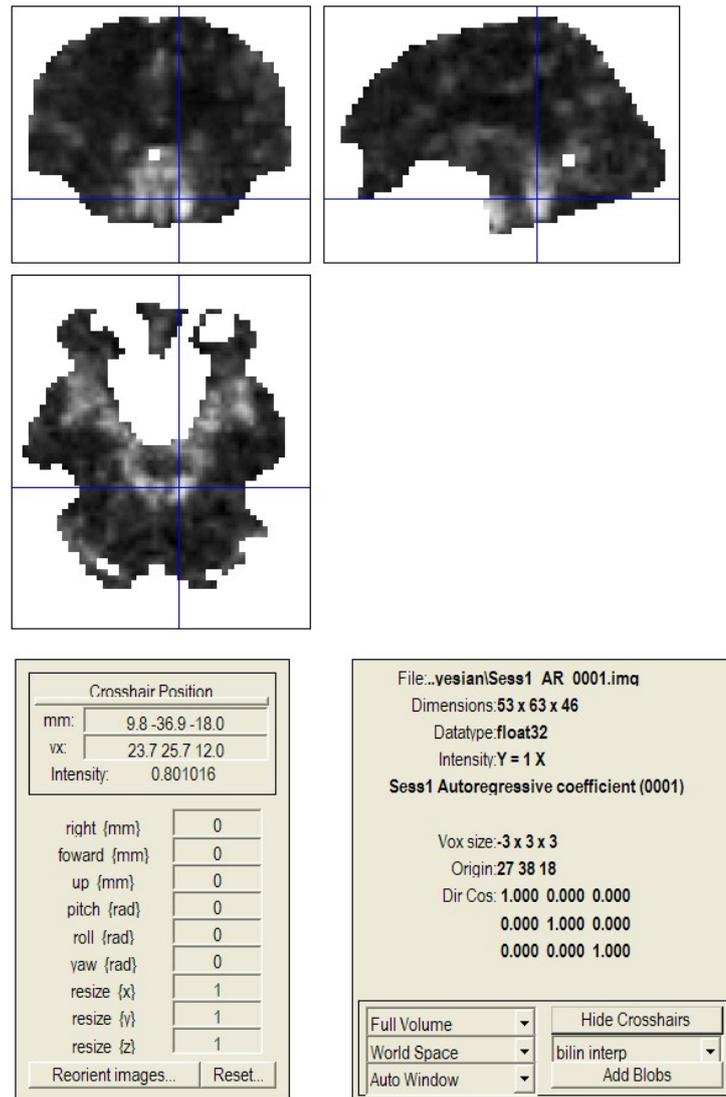
Figure 11.19: *Bayesian analysis: Estimated AR(1) coefficient image indicating heterogeneity near the circle of Willis*

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

- An `SPM.mat` file.

- Images `Cbeta_k.img` where k indexes the kth estimated 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_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 pth AR coefficient. See eg. Figure 11.19.

- Images `con_i.img` and `con_sd_i.img` which are the mean and standard deviation of the ith pre-defined contrast.

11.4.3   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).[10]

Finally, if we want to ask where is there a signal greater than 1% (with a certain confidence) to faces versus baseline, we need to create a new contrast that takes the AVERAGE of the parameter estimates for the canonical HRF across the four conditions (N1 to F2), rather than the default `Positive effect of condition_1` contrast, which actually calculates the SUM of the parameter estimates for the canonical HRF across conditions (the average vs sum makes no difference for the classical statistics).

- Press "Results".

- Select the `SPM.mat` file created in the last section.

- Press "Define new contrast", enter the name "AVERAGE Canonical HRF: Faces > Baseline", press the "T-contrast" radio button, enter the contrast [1 0 0 1 0 0 1 0 0 1 0 0]/4, press "submit", "OK" and "Done".

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

- Specify No

- *Title for comparison*

- Enter "AVERAGE Canonical HRF: Faces > Baseline"

- *Effect size threshold for PPM*

---

[10]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.
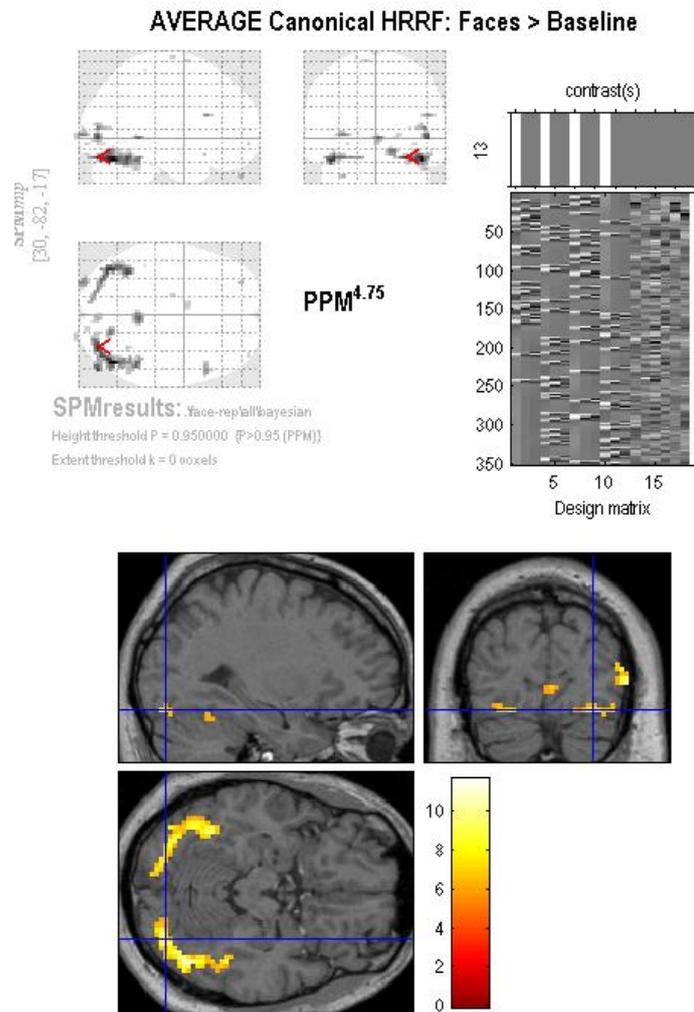
Figure 11.20: *Bayesian analysis: MIP and overlay of effect sizes at voxels where PPM is 95% sure that the effect size is greater than 1% of the global mean. The cursor is at the location* $x = 30, y = -82, z = -17mm$

- 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 11.20.

Chapter 12

Face group fMRI data

12.1 Introduction

These examples illustrate multisubject "random effects" analyses or "second-level" models of fMRI data [58][1]. 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 12×2 second timebins. This uses 12 observations (contrast images) per subject. Data are analysed using a "One-way ANOVA" with 12 levels.

12.2 Data

The data come from the "implicit" condition of the Henson et al. study [36]. Although the 1st-level design matrices (and therefore resulting contrast images) used do not correspond exactly to those used in that study.

It is also the same study from which one subject is used to illustrate a single-subject fixed effects analysis (see chapter 11 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 (64×64 3×3 mm^2), 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).

---

[1]This chapter has been largely cannibalised from an earlier document, available from `http://www.fil.ion.ucl.ac.uk/spm/data/face_rfx/spm2_face_rfx.doc`, 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([0.5 0.5],eye(Nf))` more generally) was examined.

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

- http://www.fil.ion.ucl.ac.uk/spm/data/face_rfx/cons_can.zip

The 12 (subjects) x 3 (basis functions) `con*.img`s 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/face_rfx/cons_informed.zip

The 12 (subjects) x 12 (basis functions) x 2 (contrast-types) `con*.img`s 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/face_rfx/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:\data\face group`, in which to place the results of your analysis. Then create 3 subdirectories (i) `Canonical`, (ii) `Informed`, and (iii) `FIR`. As the analysis proceeds these directories will be filled with job-specification files, design matrices and estimated models.

12.3   Canonical HRF

For the main effect versus baseline, these happen to correspond to the contrast images numbered 3-14 in 1st-level model `M1i`, ie:

- `con_0006.img` (canonical HRF, subject 1)

- `con_0007.img` (canonical HRF, subject 2)

- ...

- `con_0017.img` (canonical HRF, subject 12)

These images comprise the data for `M2c`, which is simply a "One-sample t-test". This can be implemented as follows.

- Start up MATLAB and type `spm fmri` at the prompt

- Press the "Specify 2nd-level" button. This will open the batch editor.

- In the "Design", "One-sample t-test" option, select "Scans".

- Choose "Select Files" and use the SPM file selector to choose contrast images 6 to 17.

- Highlight "Directory", "Select files" and select the subdirectory `canonical`, to place the design matrix in.

- Save the job file as eg. `DIR/canonical.mat`.

- Press the `Run` button (green arrow).

SPM will then show you the design matrix shown in Figure 12.1. This is simply a single column of 1's which will appear as a white box on a white background. This design is encoded in the `SPM.mat` file that is written to the output directory. Then press "Estimate", select the **SPM.mat** file just created, and press the `Run` button. SPM will now estimate the parameters, that is, the size of the population effect at each voxel. This is simply the average of the `con*.img`s you have specified.
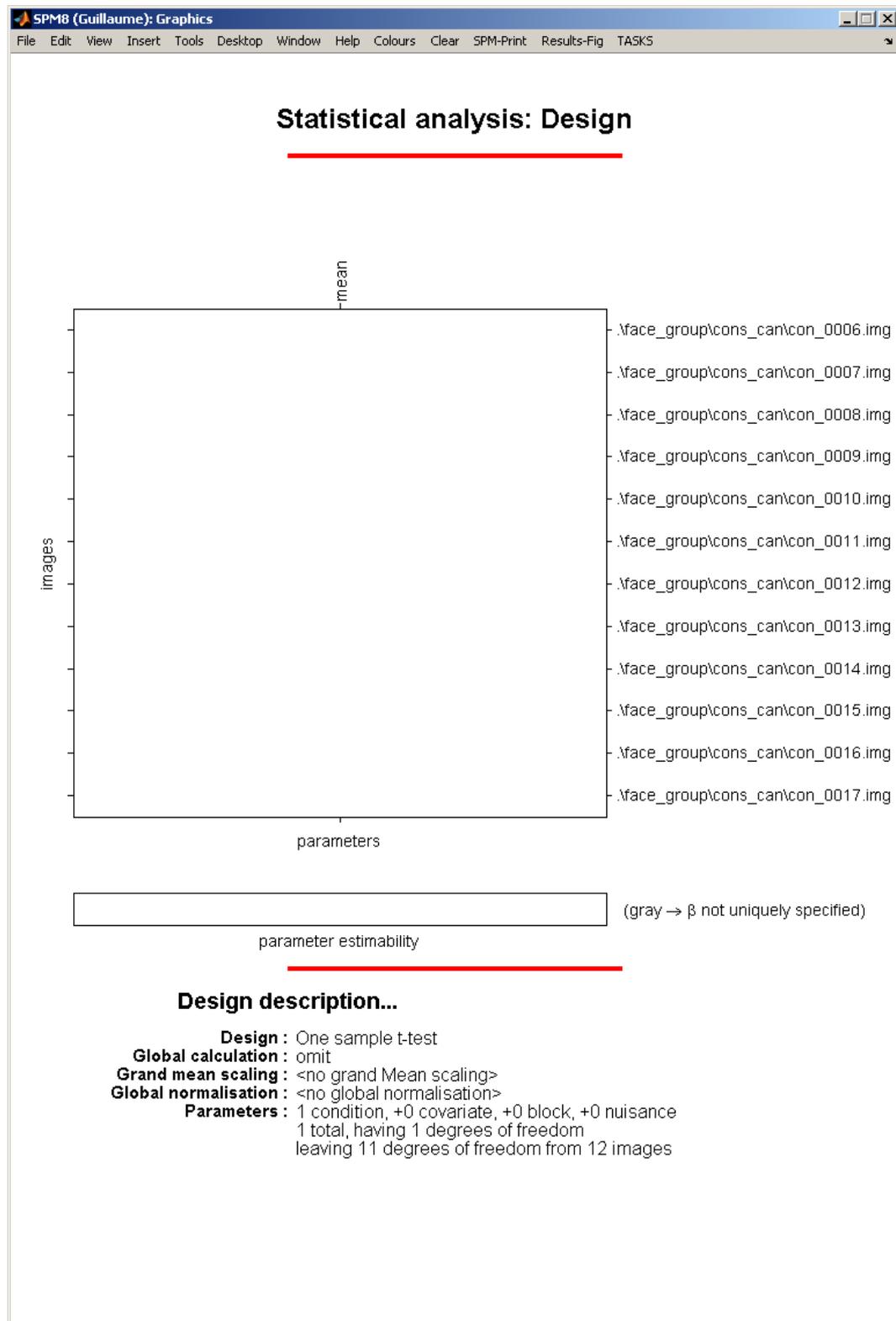
Figure 12.1: **Design matrix for canonical responses**. *This corresponds to a one-sample t-test.*

- 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, SMA, and, at a more liberal threshold, left motor cortex). You can then press the volume to get a table of stastical information for clusters of activated voxels. SPM's graphics window should look like Figure 12.2.

12.4   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.

SPM8 (Guillaume): Graphics

File Edit View Insert Tools Desktop Window Help Colours Clear SPM-Print Results-Fig TASKS

Faces vs Baseline: Canonical HRF

SPMmip
[0, 0, 0]

contrast(s)

SPM{$F_{1,11}$}

SPMresults: \data\face_group\canonical
Height threshold F = 94.701300 {p<0.05 (FWE)}
Extent threshold k = 0 voxels

Design matrix

Statistics: *p-values adjusted for search volume*

set-level		cluster-level				peak-level					mm mm mm
p	c	$p_{FWE\text{-}corr}$	$q_{FDR\text{-}corr}$	k_E	p_{uncorr}	$p_{FWE\text{-}corr}$	$q_{FDR\text{-}corr}$	F	$(Z_\equiv)$	p_{uncorr}	
0.000	17			45		0.000	0.036	547.52	6.36	0.000	42 −48 −30
						0.003	0.498	160.81	5.28	0.000	45 −63 −24
				72		0.000	0.246	277.52	5.78	0.000	−30 −60 −27
						0.006	0.535	144.80	5.18	0.000	−9 −78 −21
						0.011	0.575	127.99	5.06	0.000	−18 −66 −21
				8		0.000	0.246	251.67	5.69	0.000	−48 12 33
				31		0.001	0.382	192.97	5.45	0.000	6 −84 −3
						0.002	0.382	187.35	5.42	0.000	3 −75 6
						0.014	0.610	122.01	5.01	0.000	12 −87 −12
				14		0.003	0.497	166.76	5.31	0.000	3 15 45
				2		0.004	0.535	152.29	5.22	0.000	24 −99 −6
				2		0.007	0.535	140.56	5.15	0.000	−21 −42 −21
				7		0.007	0.535	139.89	5.14	0.000	−45 −48 −33
				5		0.008	0.535	137.04	5.12	0.000	−24 −63 0
				7		0.008	0.535	135.44	5.11	0.000	30 −69 −24
				5		0.014	0.610	121.89	5.01	0.000	−21 −78 −24
				1		0.021	0.766	112.16	4.93	0.000	15 −48 −9
				5		0.022	0.766	111.11	4.92	0.000	18 −75 −21
				1		0.027	0.777	106.85	4.88	0.000	−15 3 3

table shows 3 local maxima more than 8.0mm apart

Height threshold: F = 94.70, p = 0.000 (0.050)
Extent threshold: k = 0 voxels, p = 1.000 (0.050)
Expected voxels per cluster, <k> = 0.285
Expected number of clusters, <c> = 0.17
FWEp: 94.701, FDRp: 547.516

Degrees of freedom = [1.0, 11.0]
FWHM = 12.2 12.3 12.8 mm mm mm; 4.1 4.1 4.3 {voxels}
Volume: 1392120 = 51560 voxels = 642.1 resels
Voxel size: 3.0 3.0 3.0 mm mm mm; (resel = 71.48 voxels)
Page 1

Figure 12.2: *Main population effect of faces vs baseline, as characterised using the Canonical HRF.*

- In "Factorial design specification", highlight "Design" and then choose "Full Factorial".

- Under "Factors" create a single "New Factor".

- In this "Factor", type in "Basis" for Name and enter 3 under "Levels".

- Highlight "Independence" and select "No". SPM will then take into account possible correlations between these repeated measures (see section on Nonsphericity below for further discussion).

- Now highlight "Specify cells", and create 3 new cells.

- For the first cell, set "Levels" to 1, and enter the canonical contrast images under scans (ie contrast images numbered 0003 to 0014).

- For the second cell, set "Levels" to 2, and enter the temporal derivative contrast images under scans (ie contrast images numbered 0015 to 0026).

- For the third cell, set "Levels" to 3, and enter the dispersion derivative contrast images under scans (ie contrast images numbered 0027 to 0038.

- Highlight "Directory", "Specify files" and select the subdirectory "informed", to place the design matrix in.

- Save the job file as eg. `DIR/informed.mat`.

- Press the `Run` button in the batch editor.

SPM will then show you the design matrix shown in Figure 12.3. This design is encoded in the `SPM.mat` file that is written to the output directory. Then press "Estimate", select the `SPM.mat` file just created, and press the `Run` button. SPM will now estimate the parameters of the model (and hyperparameters governing the nonsphericity).

12.4.1   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 12.4. 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.

12.4.2   Informed Results

- Now press the "Results" button.

- Select the `SPM.mat` file.
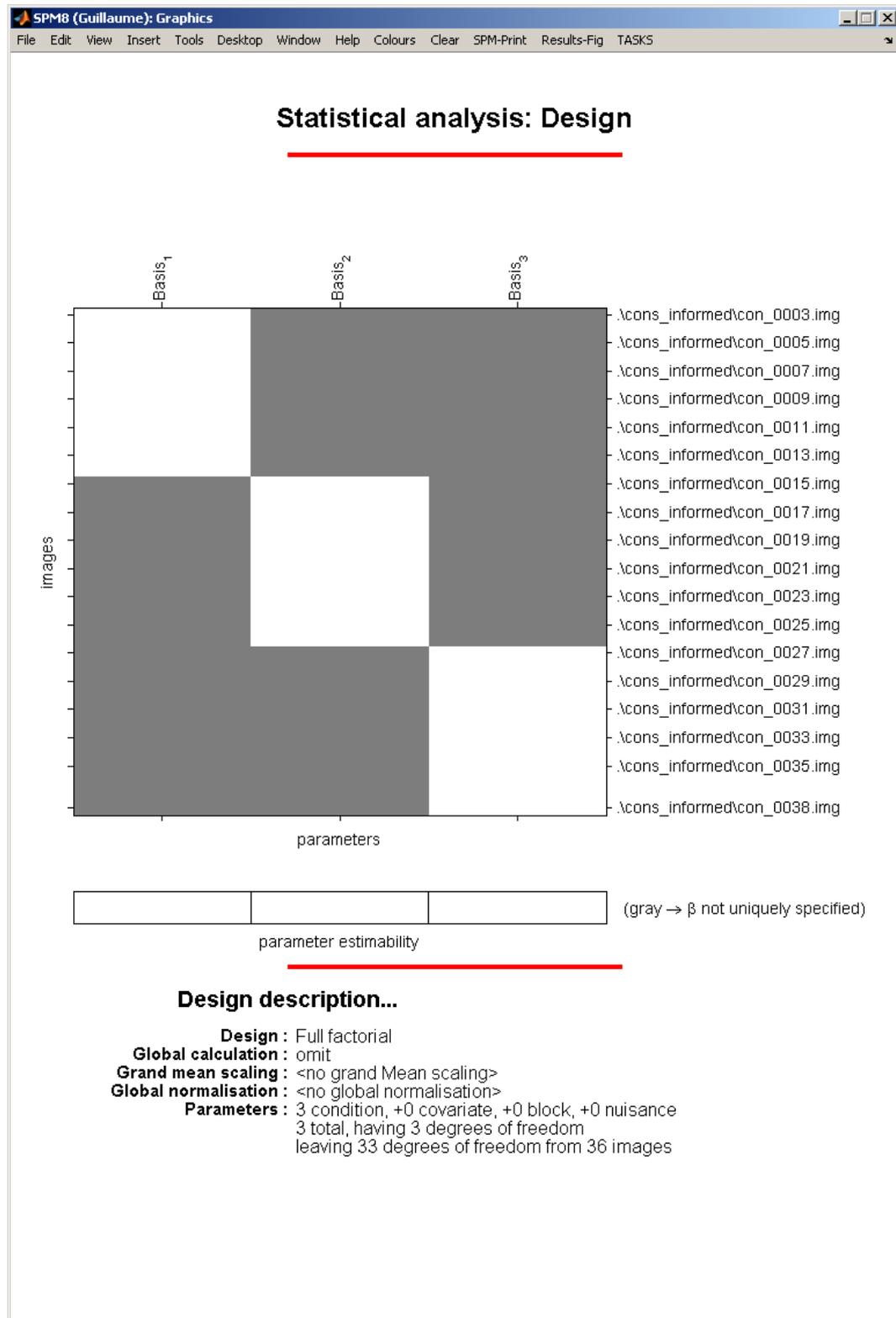
Figure 12.3: ***Design matrix for informed basis set.*** *This corresponds to a one-way ANOVA with three levels (but no constant term, since we want to test whether the basis functions are different from zero, not whether they are different from each other).*
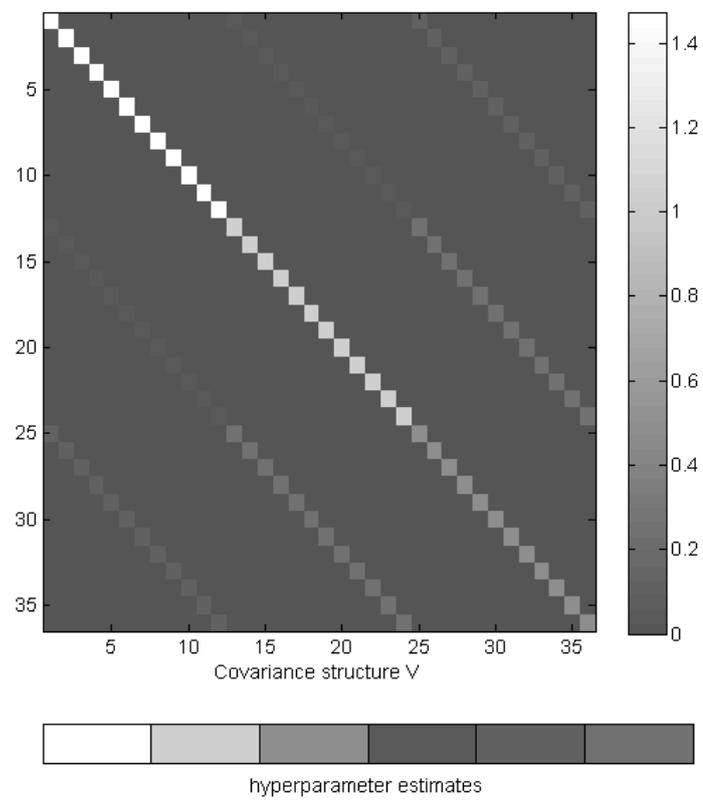
Figure 12.4: **Estimated covariance matrix for informed basis set.** *The 6 differently valued hyperparameters are shown in different shades of gray.*

- 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 the identity matrix [1 0 0; 0 1 0; 0 0 1].[2]

- 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 12.5.

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 12.5) produces more significant results than the corresponding F-contrast in the model with the canonical HRF shown in Figure 12.2. 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 12.6. 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).

12.4.3 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 12.7.

At a FWE correct p-value of 0.05, note more voxels (including now left motor cortex) and higher Z-values (e.g, 39 -57 -30, Z=7.53) for this main effect vs baseline compared to the equivalent T-contrast ([1]) in the model that uses only the canonical HRF (as in previous Section). The main reason for this increased power is the increase in the degrees of freedom, which entails better estimators of the underlying error (co)variance. The price of this increased power is a stronger assumption about the nonsphericity, namely that it has the same structure across (activated) voxels - the "pooling device", see Glaser et al. (2003) [30].

Finally, evaluate the F-contrasts [0 1 0] and [0 0 1]. These are shown in Figures 12.8 and 12.9. These contrasts reveal voxels that load (positively or negatively) on the temporal and dispersion derivatives respectively. These contrasts reveal that there is significant variability (at p<.05 corrected) that is not captured by the canonical HRF alone (see below for more discussion; see also to Henson et al (2000) [35].

---

[2]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.
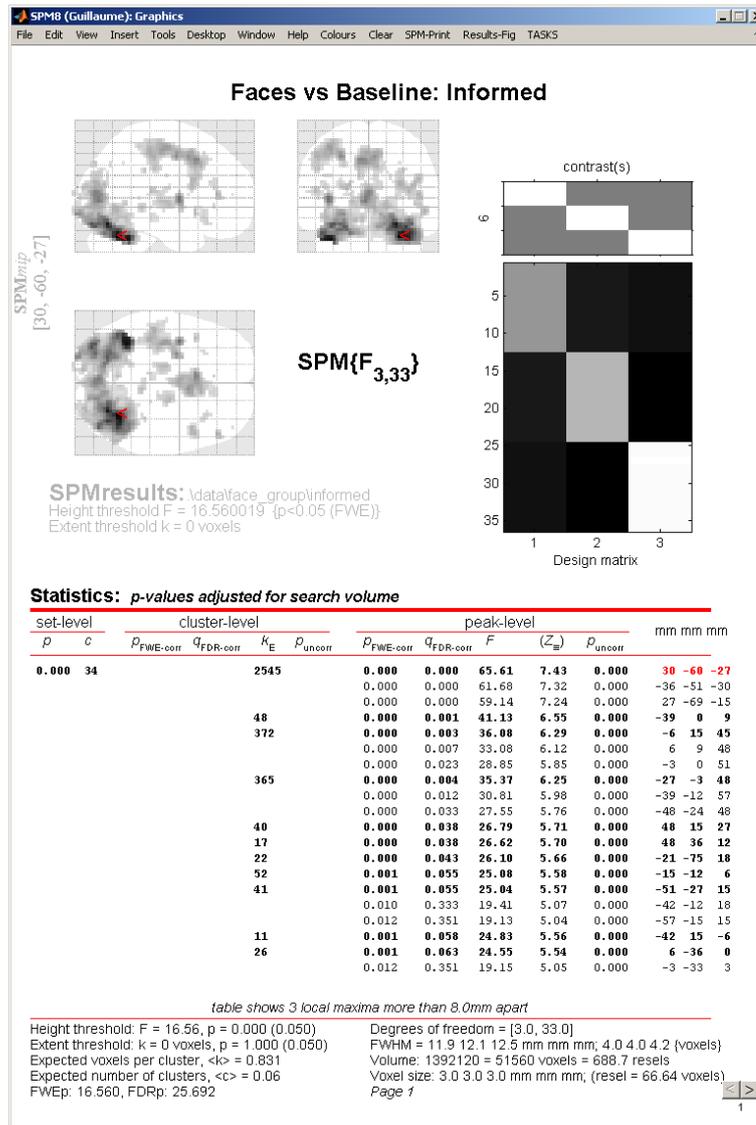
Figure 12.5: *Main population effect of faces, as characterised with the informed basis set.*
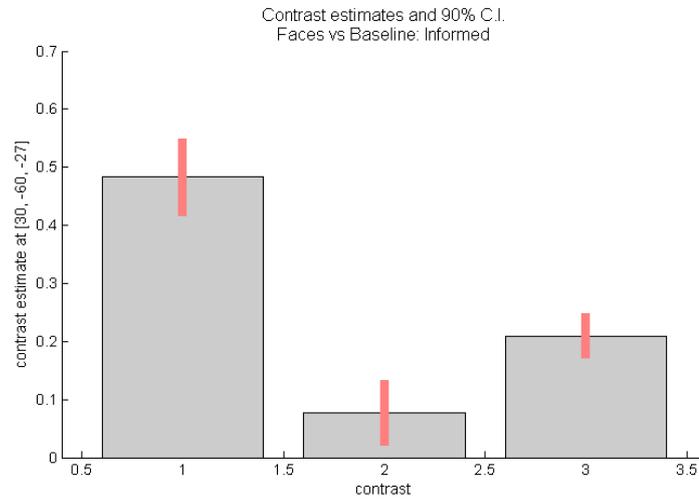
Figure 12.6: *Plotting the three basis functions for the global maximum showing reliable effects of the canonical HRF and its time and dispersion derivatives.*
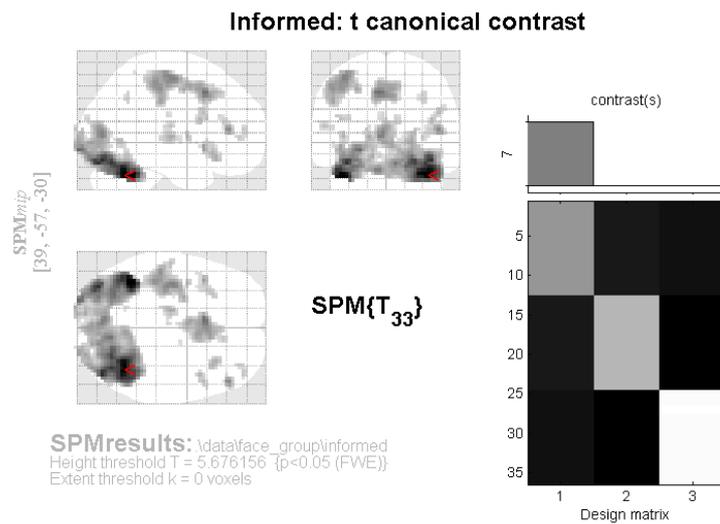
Figure 12.7: *Main population effect of faces, as characterised with the canonical HRF using a [1 0 0] t-contrast on the informed basis coefficients.*
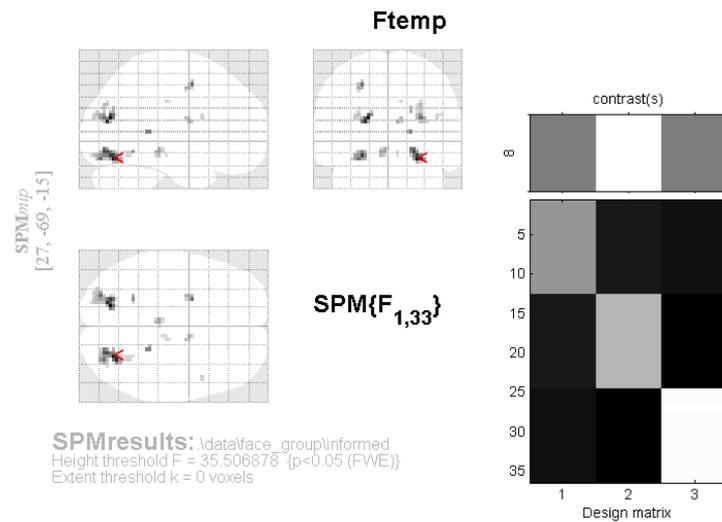
Figure 12.8: *Significantly non-zero temporal derivative coefficients. These voxels show responses earlier or later than canonical responses.*
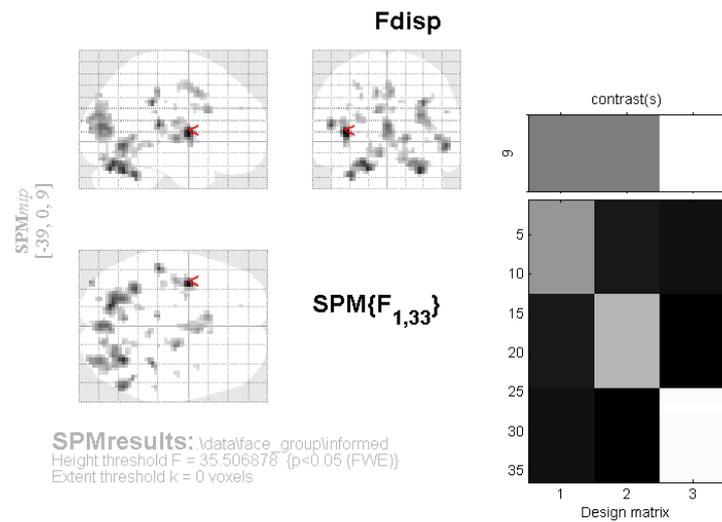
Figure 12.9: *Significantly non-zero dispersion derivative coefficients. These voxels show responses narrower or wider than canonical responses.*

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 [34]). 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).

12.5   FIR basis set

For this example, 12 contrast images per subject are taken to the 2nd-level. These are the contrast images:

- `con_fir_bin01_sub01.img` (FIR bin 1, subject 1)

- `con_fir_bin01_sub02.img` (FIR bin 1, subject 2)

- ...

- `con_fir_bin02_sub01.img` (FIR bin 2, subject 1)

- ...

These images comprise the data for `M2f`, which is simply a "One-way ANOVA" with 12-levels (one for each time-bin). This can be implemented as follows.

- Start up MATLAB and type `spm fmri` at the prompt.

- Press the "Specify 2nd-level" button.

- The options for "Factorial design specification"[3] appear.

- Highlight "Design" and then choose "Full Factorial".

- Under "Full Factorial" and 'Factors' create a single "New Factor".

- In this "Factor", type in "TimeBin" for "Name" and enter 12 under "Levels".

- Highlight "Independence" and select "No". SPM will then take into account possible correlations between these repeated measures.

- Now highlight "Specify cells", and create 12 new cells.

- For the first cell, set "Levels" to `1`, and enter the contrast images for time bin 1 under scans. This is most easily done by changing the filter to `.*fir_bin01.*`.

- For the second cell, set "Levels" to `2`, and, under scans, enter the contrast images for time bin 2 This is most easily done by changing the filter to `.*fir_bin02.*`.

- Similarly for Levels 3 to 12.

---

[3]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.
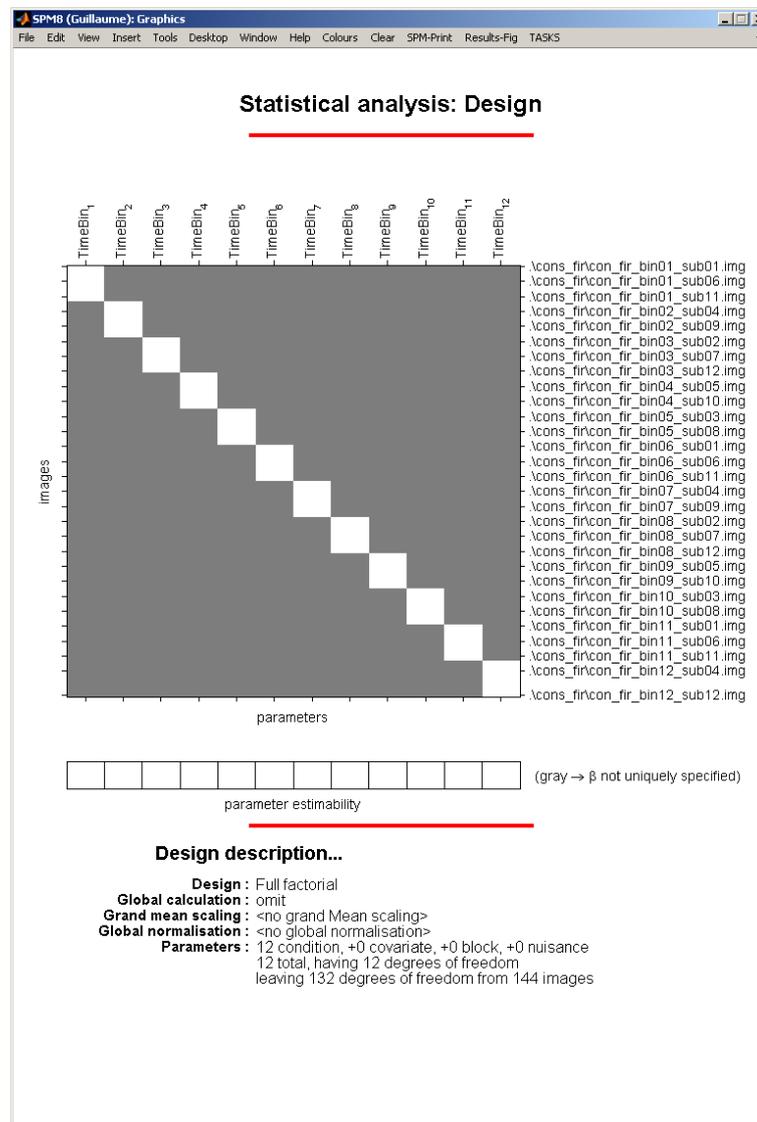
Figure 12.10: *Design matrix for FIR basis set. This corresponds to a one-way ANOVA with 12 levels.*

- 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 batch editor.

SPM will then show you the design matrix shown in Figure 12.10. This design is encoded in the `SPM.mat` file that is written to the output directory. Then press "Estimate", select the `SPM.mat` file just created, and press the button `Run`. SPM will now estimate the parameters of the model.

12.5.1   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).
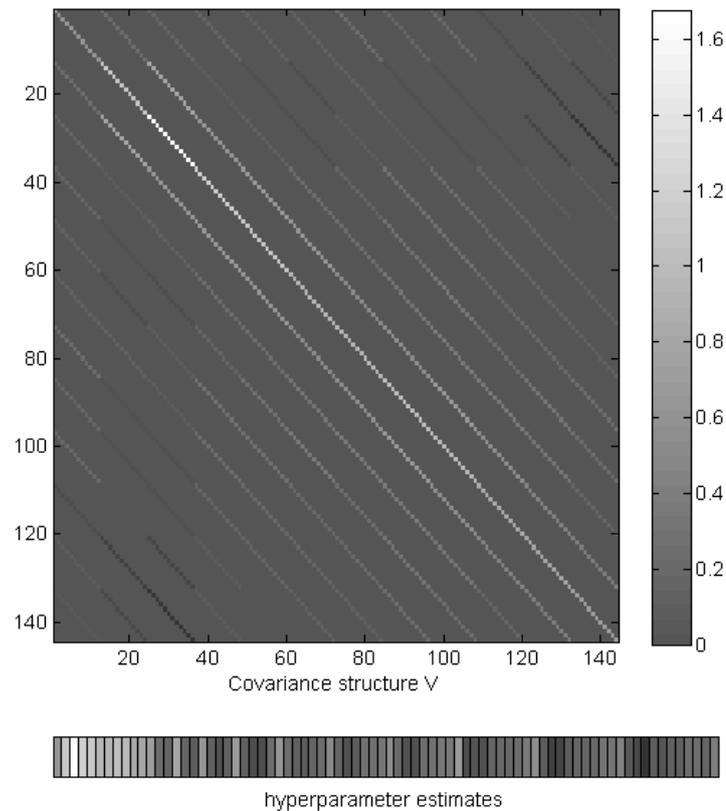
Figure 12.11: *Estimated covariance matrix for FIR basis set. The differently valued hyperparameters are shown in different shades of gray. Notice that the most variable responses occur in the third time bin (scans 25 to 36) corresponding to responses 4-6 seconds post stimulus, ie. at the peak of the 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 12.11. 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 "inhomogeniety 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).

12.5.2   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".[4]

- 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 12.12 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 12.11).

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 12.12. Selecting a voxel and plotting this contrast (using the *plot* button) will reveal that most voxels have a fairly "canonical" shape over the 12 timebins. One can also test for more constrained shapes of event-related responses within this model. For example, one can test for "canonical-shaped" responses by evaluating a contrast whose weights trace out SPM's canonical HRF (every 2s). To do this, switch to the MATLAB window for a moment and type:

- xBF.dt = 1

- xBF.name = 'hrf (with time and dispersion derivatives)';

- xBF.length = 32;

- xBF.order = 1;

- xBF = spm_get_bf(xBF);

This returns the canonical and two derivatives in the matrix xBF.bf (type help spm_get_bf for more info), with one value every 1 second. For convenience, then define:

- all   = xBF.bf(2:2:24,:)';

- can   = all(1,:);

- tem   = all(2,:);

- dis   = all(3,:);

These commands downsample the basis functions every 2s, which is the bin-width of the FIR. If you type corrcoef(all'), you will see that the basis functions are slightly correlated (in the off-diagonal terms), due to this undersampling every 2s.

- In the contrast manager press "Define new contrast" (select T).

- Enter can as the contrast weights (defined in MATLAB workspace as above), and "Can-weighted FIR" as the name.

This produces the MIP in Figure 12.13. 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)
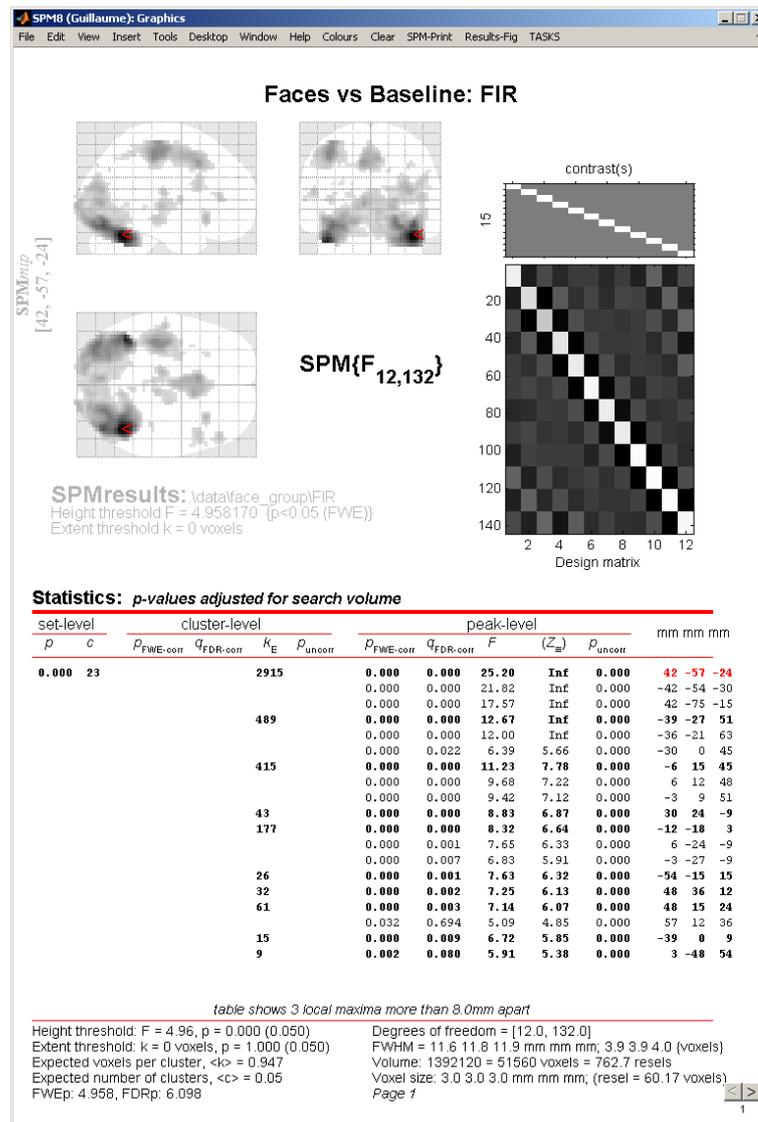
Figure 12.12: *Main population effect of faces, as characterised with the FIR basis set.*
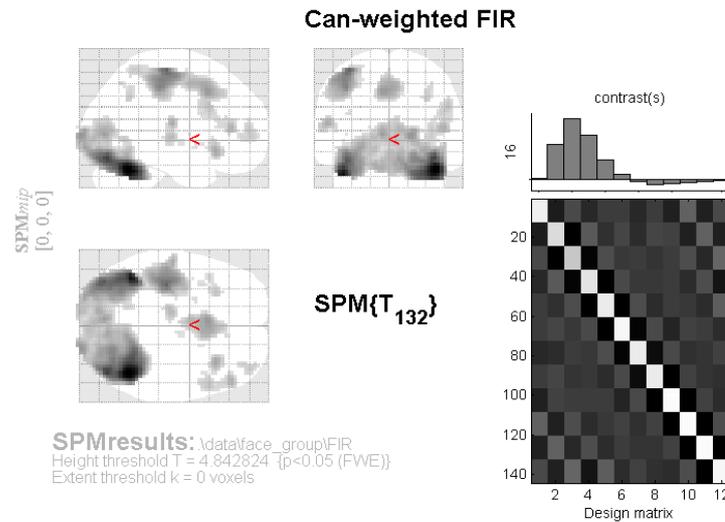
Figure 12.13: *Main population effect of faces, as characterised with a canonically weighted contrast of FIR bases.*

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.

You can see, in Figure 12.14 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.

You will see, in Figure 12.15 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

---

[4]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.
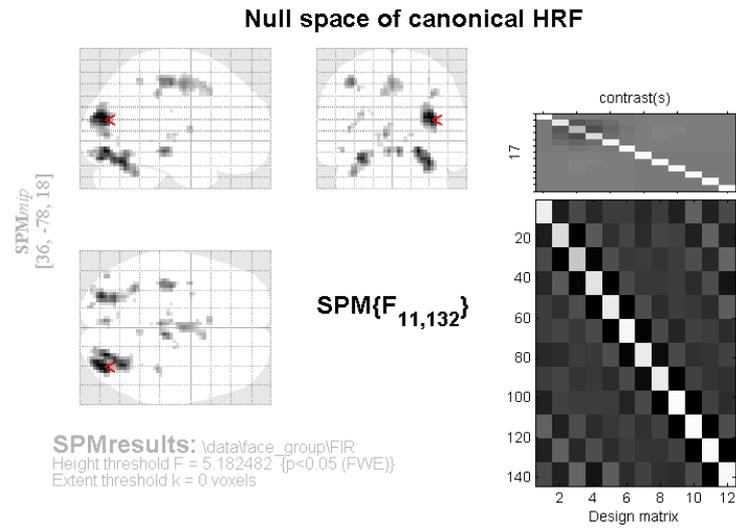
Figure 12.14: *Regions expressing variability across subjects not captured by canonical HRF.*
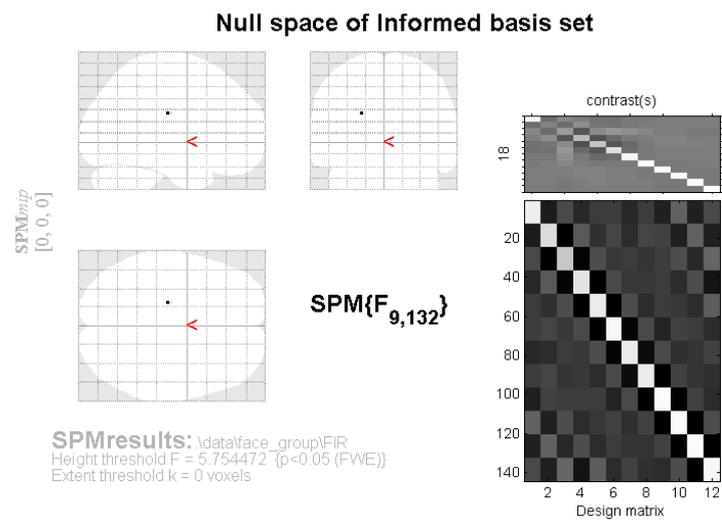
Figure 12.15: *Regions expressing variability across subjects not captured by informed basis set.*

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 [35] for further details.

Chapter 13

Verbal Fluency PET data

13.1 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 13.1.

Scan:	1	2	3	4	5	6	7	8	9	10	11	12
Subject 1	A	B	A	B	A	B	A	B	A	B	A	B
Subject 2	B	A	B	A	B	A	B	A	B	A	B	A
Subject 3	A	B	A	B	A	B	A	B	A	B	A	B
Subject 4	B	A	B	A	B	A	B	A	B	A	B	A
Subject 5	A	B	A	B	A	B	A	B	A	B	A	B

Table 13.1: **Conditions for PET data:** *(A) word shadowing and (B) word generation.*

The files are named `./p#/snrp#_##.{img,hdr}` and are SPM compatible (Analyze) images following realignment, normalization and smoothing with a 16mm isotropic Gaussian kernel with `#` indicating the subject and `##` the scan. The data set is available from the SPM website[1].

To analyse the data, first create a new directory `DIR`, eg. `c:\data\pet`, in which to place the results of your analysis. Then create 4 subdirectories (i) `single`, (ii) `subject-condition`, (iii) `subject-time` and (iv) `multiple`. As the analysis proceeds these directories will be filled with job-specification files, design matrices and estimated models.

13.2 Single subject

Firstly, we will analyse the data from a single subject. This can be implemented as follows.

- Start up MATLAB and type `spm pet` at the prompt

- Press the "Basic models" button.

- In 'Factorial design specification', choose the 'Flexible Factorial' design.

- Highlight 'Factors' and create a new Factor and enter 'Word' for the name.

- Then, under 'Specify Subject or all Scans and Factors', highlight 'Subjects' and create a new subject.

---

[1]Verbal Fluency PET dataset: http://www.fil.ion.ucl.ac.uk/spm/data/fluency/

- Highlight 'Scans', select 'Specify Files' and use the SPM file selector to choose the 12 images for that subject. This can be most easily achieved by specifying '.*snrp1.*' as a filter in the file selector.

- Under 'Conditions' enter the vector [1 2 1 2 1 2 1 2 1 2 1 2].

- Under 'Main effects and interactions', create a single main effect with factor number equal to 1

- Under 'Covariates', create a new covariate and enter 'Time' for 'Name' and the vector '1:12'.

- Under 'Global calculation' choose 'Mean'

- Under Global normalisation and Normalisation, choose 'Proportional' scaling.[2]

- Under Global normalisation and Overall grand mean scaling, select YES.

- Highlight Directory, Specify files and select the subdirectory 'single', to place the design matrix in.

- Save the job file as eg. `DIR/single_design.mat`.

- Press the `Run` button (green arrow).

SPM will then show you the design matrix shown in Figure 13.1. 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.

13.3   Multiple subjects

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

---

[2]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.
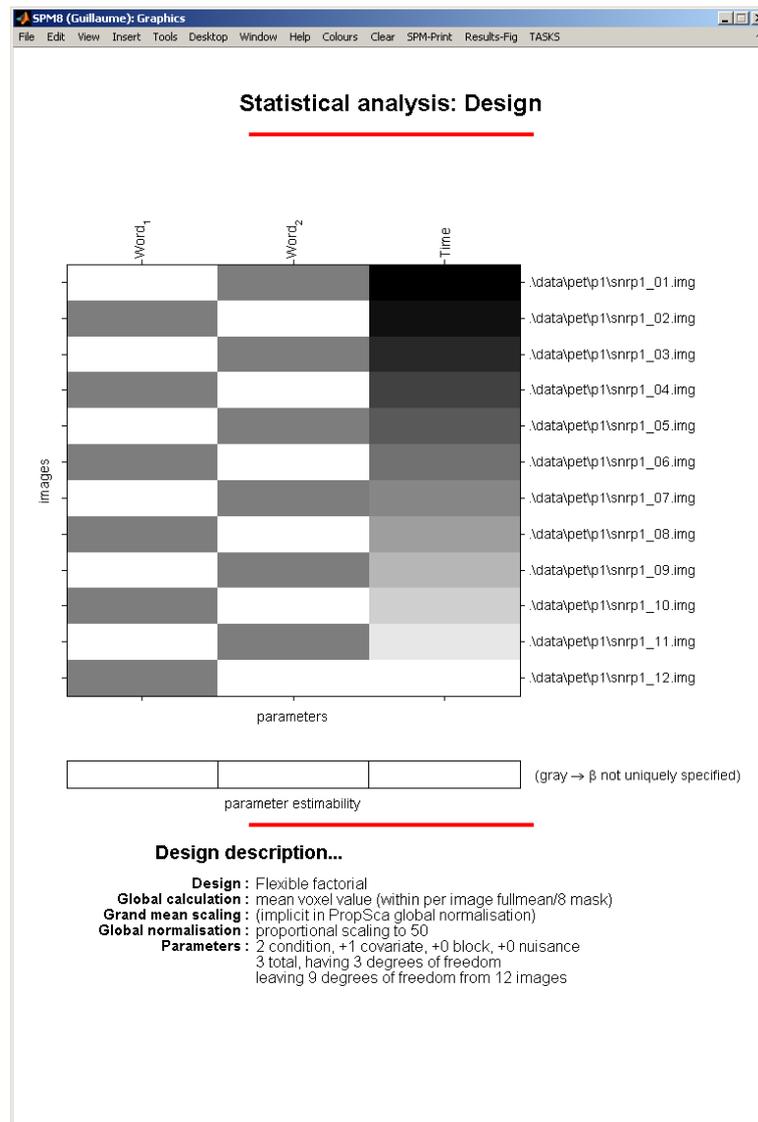
Figure 13.1: *Design matrix for single-subject data. The first two columns model responses to word shadowing and word generation. The third column models time-varying responses.*

13.3.1 Subject and Condition design

First we set up a design that allows us to test for the main effects of 'Subject' and 'Condition'. The design can be set-up as follows.

- Start up MATLAB and type `spm pet` at the prompt

- Press the 'Basic Models' button.

- In 'Factorial design specification', under 'Design', choose the 'Flexible Factorial' design.

- Highlight 'Factors' and create a new Factor.

- Enter 'Subject' for the name and select 'Equal' under 'Variance'.

- Then create another factor and call it 'Word'

- Then, under 'Specify Subject or all Scans and Factors', highlight 'Subjects' and create a 5 new subjects.

- For the first subject, highlight 'Scans', select 'Specify Files' and use the SPM file selector to choose the 12 images for the first subject. This can be most easily achieved by specifying `.*snrp1.*` as a filter in the file selector and then using a right click to 'select all'.

- Under 'Conditions' enter the vector [1 2 1 2 1 2 1 2 1 2 1 2].

- Repeat the specification of scans and conditions for each of the four other subjects, remembering that the order of conditions has been balanced across the group (see Table 13.1).

- Under 'Main effects and interactions', create two main effects, the first with factor number 1 (ie. Subject) and the second with factor number 2 (ie. Word).

- Under Masking, select 'Relative' for 'Threshold Masking' and accept the default value of 0.8. Voxels with mean value less than 0.8 of the mean are deemed extra-cranial and will be excluded from the analysis.

- Under 'Global calculation' choose 'Mean'

- Under, Global normalisation, and Normalisation, select 'ANCOVA'.

- Highlight Directory, Specify files and select the subdirectory 'subject-condition', to place the design matrix in.

- Save the job file as eg. `DIR/sc_design.mat`.

- Press the `Run` button.

SPM will then show you the design matrix shown in Figure 13.2. This design is encoded in the `SPM.mat` file that is written to the output directory.

13.3.2 Subject and Time design

We now set up a design that allows us to test for the effects of Time (ie. scan number) and Subject. If you have already specified the Subject and Conditions design, then you can set up the Subject and Time design by editing the `sc_design.mat` file (and just changing the name of the second factor, Conditions vector and output directory - see below). Otherwise, the design can be set-up as follows.

- Start up MATLAB and type `spm pet` at the prompt

- Press the 'Basic Models' button.

- In 'Factorial design specification', under 'Design', choose the 'Flexible Factorial' design.

- Highlight 'Factors' and create a new Factor.
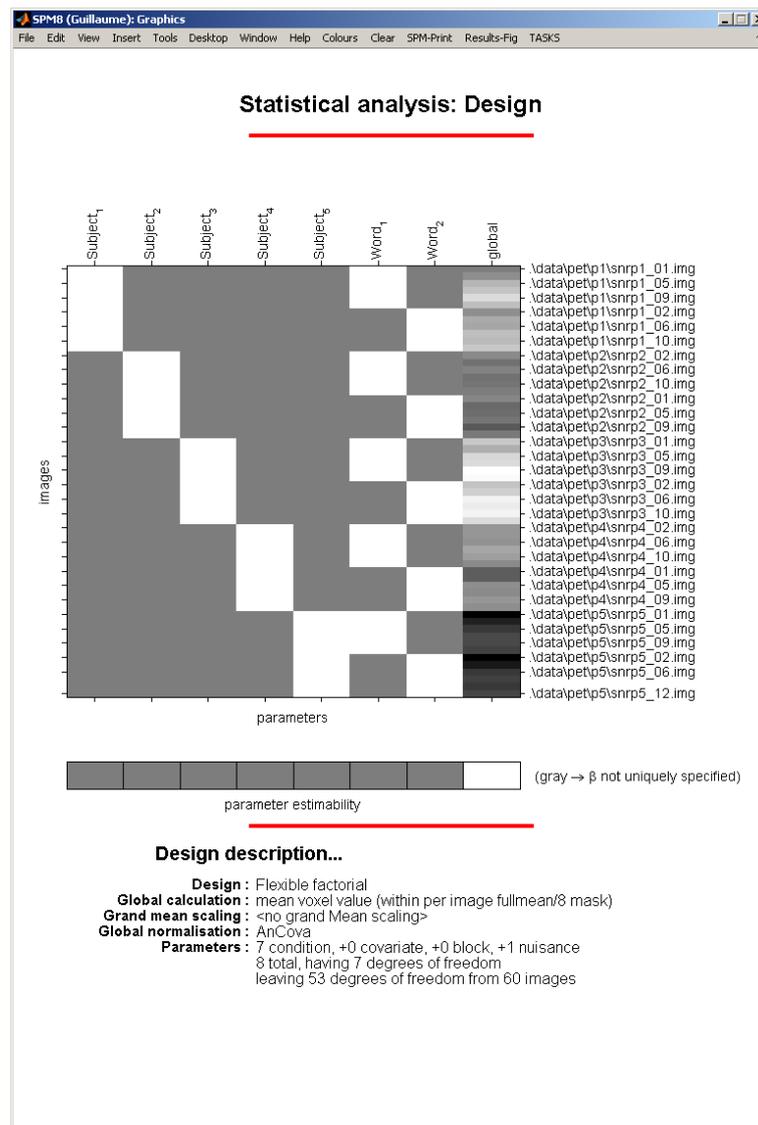
Figure 13.2: *Subjects and Conditions design for multiple-subject data. The first five columns model effect and the next two columns model condition effects. The last colum models global effects (ANCOVA).*

- Enter 'Subject' for the name and select 'Equal' under 'Variance'.

- Then create another factor and call it 'Time'. This factor extends over time for each subject.

- Then, under 'Specify Subject or all Scans and Factors', highlight 'Subjects' and create a 5 new subjects.

- For the first subject, highlight 'Scans', select 'Specify Files' and use the SPM file selector to choose the 12 images for the first subject. This can be most easily achieved by specifying `.*snrp1.*` as a filter in the file selector and then using a right click to 'select all'.

- Under 'Conditions' enter the vector [1:12].

- Repeat the specification of scans and conditions for each of the four other subjects.

- Under 'Main effects and interactions', create two main effects, the first with factor number 1 (ie. Subject) and the second with factor number 2 (ie. Time).

- Under Masking, select 'Relative' for 'Threshold Masking' and accept the default value of 0.8. Voxels with mean value less than 0.8 of the mean are deemed extra-cranial and will be excluded from the analysis.

- Under 'Global calculation' choose 'Mean'

- Under, Global normalisation, and Normalisation, select 'ANCOVA'.

- Highlight Directory, Specify files and select the subdirectory 'subject-condition', to place the design matrix in.

- Save the job file as eg. `DIR/st_design.mat`.

- Press the `Run` button.

SPM will then show you the design matrix shown in Figure 13.3. This design is encoded in the `SPM.mat` file that is written to the output directory.

13.3.3   Subject by Condition design

This design models the interacts between 'Subject' and 'Condition'. It allows effects to be assessed separately for each subject. It will also allow us to implement a conjunction analysis over subjects.

If you have already specified the Subject and Conditions or Subject and Time designs then this design can be more easily specified by editing the `sc_design.mat` or `st_design.mat` files (and changing the name of the second factor, removing main effects, adding the interaction term and specifying a new output directory - see below). Otherwise, the design can be set-up as follows.

- Start up MATLAB and type `spm pet` at the prompt

- Press the "Basic Models" button.

- In 'Factorial design specification', under 'Design', choose the 'Flexible Factorial' design.

- Highlight 'Factors' and create a new Factor.

- Enter 'Subject' for the name and select 'Yes' under ANCOVA, as we will be implementing ANCOVA-by-subject. Select 'Equal' under 'Variance'.

- Then create another factor and call it 'Word'

- Then, under 'Specify Subject or all Scans and Factors', highlight 'Subjects' and create a 5 new subjects.

- For the first subject, highlight 'Scans', select 'Specify Files' and use the SPM file selector to choose the 12 images for the first subject. This can be most easily achieved by specifying '.*snrp1.*' as a filter in the file selector and then using a right click to 'select all'.
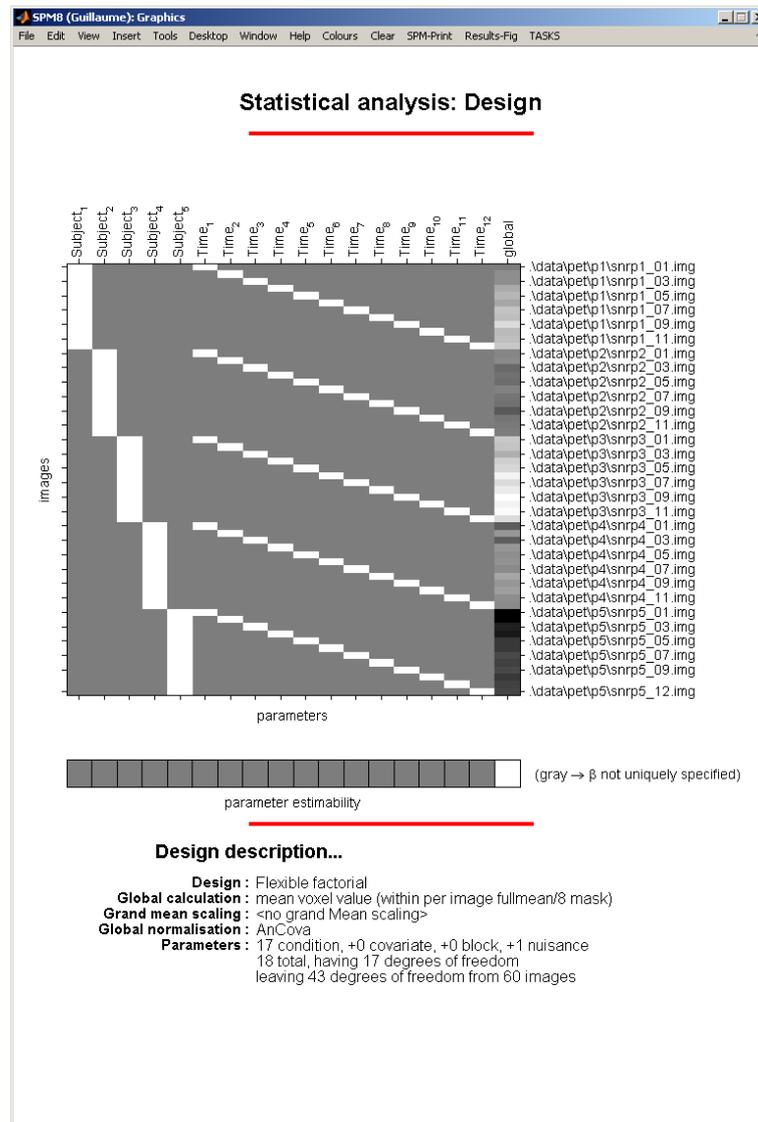
Figure 13.3: *Subjects and Time design for multiple-subject data. The first five columns model subjects effects and the next 12 model time effects. The last column models global effects (ANCOVA).*

- Under 'Conditions' enter the vector [1 2 1 2 1 2 1 2 1 2 1 2].

- Repeat the specification of scans and conditions for each of the four other subjects, remembering that the order of conditions has been balanced across the group (see Table 13.1).

- Under 'Main effects and interactions', create an interaction with factor numbers equal to [1 2]. This will create a block in the design matrix that models interactions between the factors 'Subject' and 'Word'.

- Under Masking, select 'Relative' for 'Threshold Masking' and accept the default value of 0.8. Voxels with mean value less than 0.8 of the mean are deemed extra-cranial and will be excluded from the analysis.

- Under 'Global calculation' choose 'Mean'

- Highlight Directory, Specify files and select the subdirectory `multiple`, to place the design matrix in.

- Save the job file as eg. `DIR/multi_design.mat`.

- Press the `Run` button.

SPM will then show you the design matrix shown in Figure 13.4. 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.

13.3.4 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 13.5. As shown, SPM will automatically pad '0 0 -1 1' with zeros at the end. To examine group effects:

- Press the 'Results' button.

- Select the SPM.mat file.

- In the contrast manager press 'Define new contrast' (select T)

- Specify e.g. `All: Gen > Shad` (name) and '-1 1 -1 1 -1 1 -1 1 -1 1' and select it (press 'Done') (contrast).
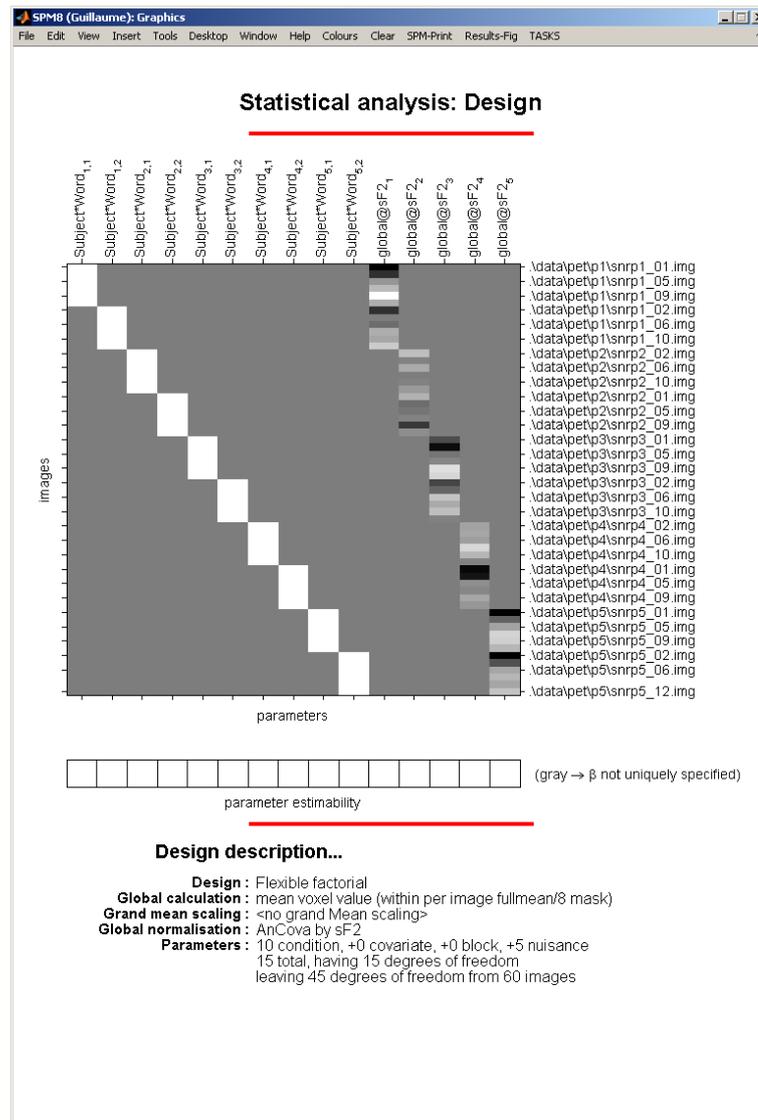
Figure 13.4: *Subject by Condition design for multiple-subject data. The first ten columns model interactions between 'Subject' and 'Word'. The last five columns model out global effects for each subject. Inclusion of these last five regressors implements a so-called 'ANCOVA-by-subject' normalisation.*
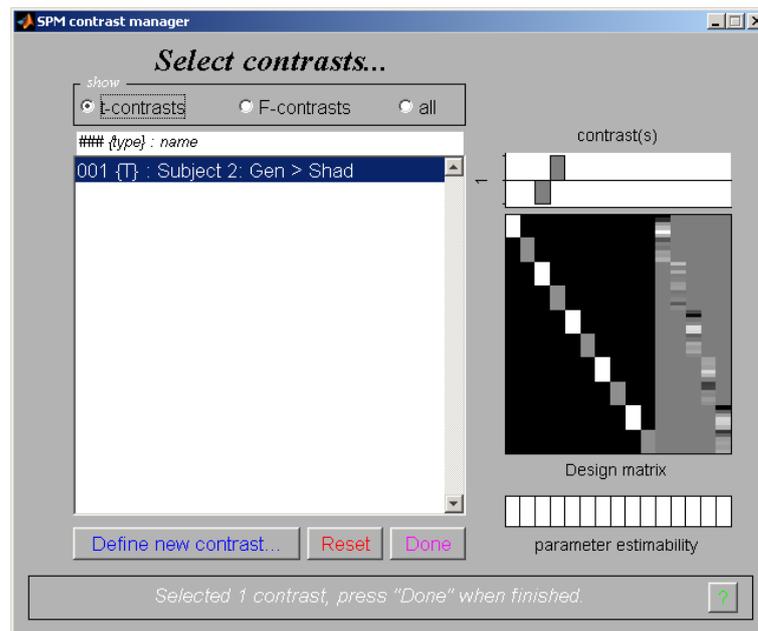
Figure 13.5: *Activation contrast for subject 2. Note that the block of the design matrix encoding the experimental conditions is now coloured differently. This is because we have allowed the variance of responses over subjects to be different between word shadowing and generation conditions. This 'nonsphericity' 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.*

- 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.

13.3.5   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.
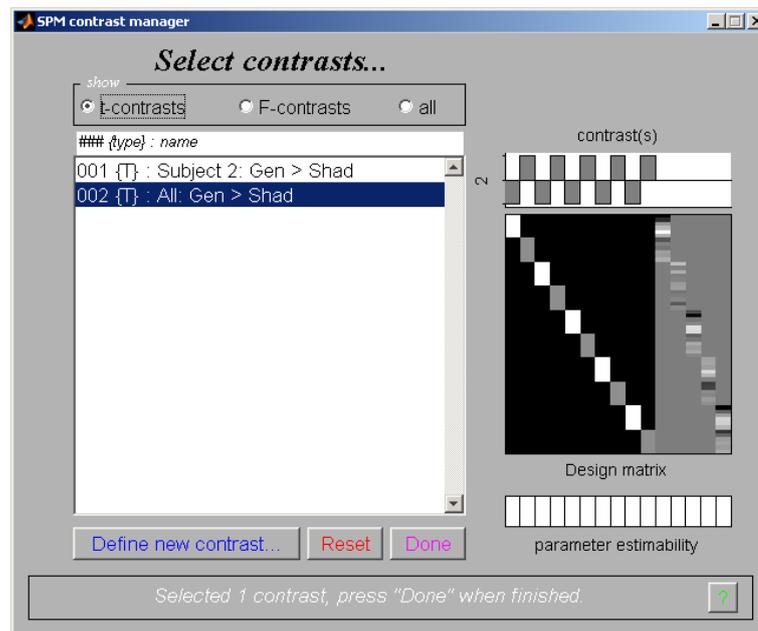
Figure 13.6: *Activation contrast for all subjects.*

13.3.6 MIPs and results tables

The above contrast specifications should configure the contrast manager to appear as in Figure 13.6 and will configure SPM's graphics window to look like Figure 13.7.

SPM will also produce a number of files: images containing weighted parameter estimates are saved as `con_0002.hdr/img`, `con_0003.hdr/img`, etc. in the output directory. Images of T-statistics are saved as `spmT_0002.hdr/img`, `spmT_0003.hdr/img`, etc., also in the output directory. A number of further options are available from SPM Interactive window shown in Figure 13.8.

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 13.9. As in the previous example, this will list all clusters above the chosen level of significance as well as separate (>8mm apart) maxima within a cluster, with details of significance thresholds and search volume underneath. The columns show, from right to left:

- **x, y, z (mm):** coordinates in Talairach space for each maximum.

- **peak-level:** the chance (p) of finding (under the null hypothesis) a peak with this or a greater height (T- or Z-statistic), corrected / uncorrected for search volume.

- **cluster-level:** the chance (p) of finding a cluster with this or a greater size (ke), corrected / uncorrected for search volume.

- **set-level:** the chance (p) of finding this or a greater number of clusters (c) in the search volume.

Its also worth noting that

- The table is surfable: clicking a row of cluster coordinates will move the pointer in the MIP to that cluster, clicking other numbers will display the exact value in the Matlab window (e.g. 0.000 = 6.1971e-07).

- To inspect a specific cluster, either move the cursor in the MIP (by L-clicking & dragging the cursor, or R-clicking the MIP background which will activate a pulldown menu).
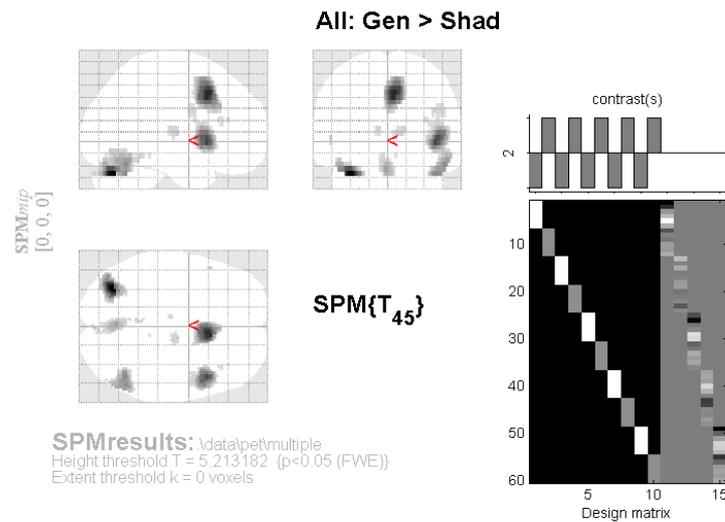
Figure 13.7: *SPMs graphics window displays (Left) a maximum intensity projection (MIP) on a glass brain in three orthogonal planes. The MIP is surfable: right-clicking in the MIP will activate a pulldown menu, left-clicking on the red cursor will allow it to be dragged to a new position, (Right) the design matrix (showing the selected contrast). The design matrix is also surfable: right-clicking will show parameter names, left-clicking will show design matrix values for each scan.*
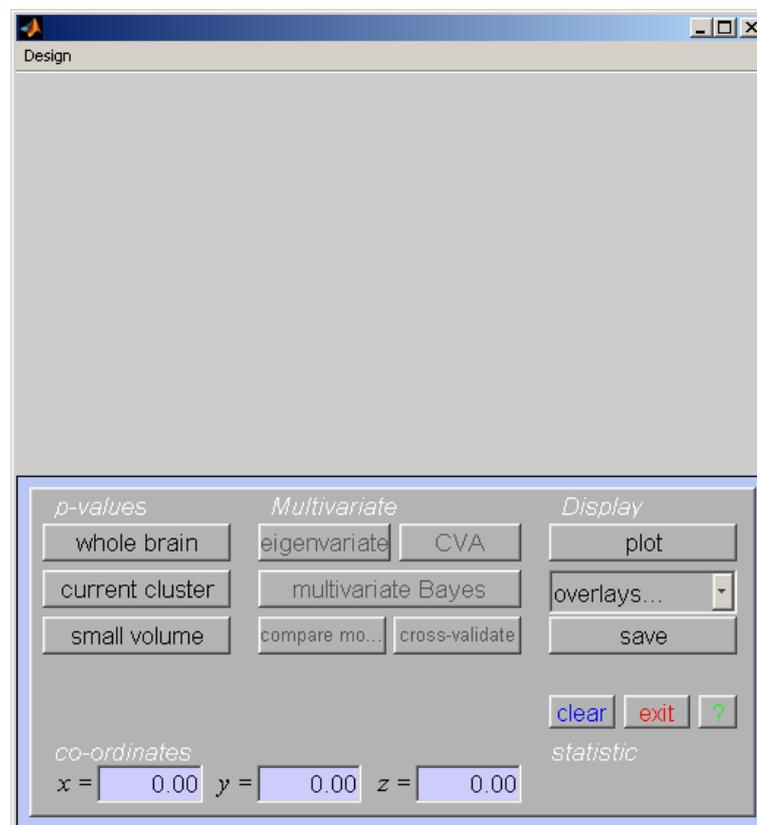
Figure 13.8: *SPM's interactive window.*

Statistics: *p-values adjusted for search volume*

set-level		cluster-level				peak-level					mm mm mm
p	c	$p_{FWE\text{-}corr}$	$q_{FDR\text{-}corr}$	k_E	p_{uncorr}	$p_{FWE\text{-}corr}$	$q_{FDR\text{-}corr}$	T	$(Z_\equiv)$	p_{uncorr}	
0.000	16	0.000	0.000	138	0.000	0.000	0.000	11.04	7.64	0.000	-34 -70 -28
						0.000	0.009	7.31	5.90	0.000	-44 -74 -24
		0.000	0.000	452	0.000	0.000	0.000	9.82	7.14	0.000	6 16 40
		0.000	0.000	300	0.000	0.000	0.000	9.14	6.84	0.000	44 16 0
						0.041	0.833	5.29	4.64	0.000	38 12 16
		0.000	0.000	173	0.000	0.000	0.009	7.39	5.95	0.000	44 -58 -28
						0.000	0.009	7.35	5.93	0.000	52 -58 -20
						0.002	0.087	6.42	5.38	0.000	50 -66 -24
		0.000	0.000	112	0.000	0.000	0.025	6.93	5.69	0.000	-2 -66 -24
						0.012	0.418	5.73	4.94	0.000	4 -78 -24
						0.014	0.472	5.65	4.89	0.000	2 -86 -28
		0.013	0.374	3	0.257	0.010	0.406	5.77	4.97	0.000	-52 20 4
		0.000	0.019	20	0.008	0.011	0.406	5.76	4.96	0.000	10 -10 8
		0.008	0.263	5	0.148	0.016	0.472	5.63	4.87	0.000	-8 -16 12
		0.000	0.012	24	0.004	0.016	0.472	5.61	4.86	0.000	44 4 28
						0.035	0.736	5.34	4.68	0.000	46 6 20
		0.006	0.231	6	0.116	0.018	0.472	5.59	4.84	0.000	-6 -48 -16
		0.026	0.520	1	0.520	0.021	0.538	5.52	4.80	0.000	-6 -54 -16
		0.026	0.520	1	0.520	0.030	0.713	5.40	4.72	0.000	6 -84 -28

table shows 3 local maxima more than 8.0mm apart

Height threshold: T = 5.21, p = 0.000 (0.050)	Degrees of freedom = [1.0, 45.0]
Extent threshold: k = 0 voxels, p = 1.000 (0.050)	FWHM = 9.8 10.6 15.6 mm mm mm; 4.9 5.3 3.9 {voxels}
Expected voxels per cluster, <k> = 2.519	Volume: 880432 = 55027 voxels = 472.2 resels
Expected number of clusters, <c> = 0.05	Voxel size: 2.0 2.0 4.0 mm mm mm; (resel = 102.26 voxels)
FWEp: 5.213, FDRp: 6.702, FWEc: 1, FDRc: 20	*Page 1*

Figure 13.9: *SPM results table. This appears below the MIP, shown in Figure 13.7, 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 13.10. The table is also surfable. Both in the 'volume' and 'cluster' options, p-values are corrected for the entire search volume.

13.3.7 Small volume correction

If one has an a priori anatomical hypothesis, eg. in the present example Broca's area will likely be activated during word generation, one may use the small volume correction option. Press the "small volume" button in SPM Interactive (bottom left) window and select a suitable region, e.g., a 30mm sphere with its centre at 44 16 0. The region can also be defined using mask images derived from previous imaging data. The corrected p-values will change, as shown in Figure 13.11.

13.3.8 Extracting data from regions

To extract a time course for data in this region of interest (this uses the SPM function `spm_regions.m`):

- Select "eigenvariate" from the "Multivariate" section in the Interactive window

- 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 13.12. It also lists the eigenvariate values Y in the Matlab window. Adjustment is with respect to the null space of a selected contrast. This means that any effects not spanned by the chosen contrast are removed from the data, before extraction. Adjustment can be omitted by selecting 'don't adjust', as above.

SPM extracts the eigenvariate values in a region, rather than the mean values, as the former is more robust to heterogeneity of response within a cluster. The mean value can be thought of as a special case of the eigenvariate if the corresponding eigenvector weights all voxels in a cluster equally. Effectively, the eigenvariate provides a weighted mean where atypical voxels are downweighted.

Statistics: *p-values adjusted for search volume*

	cluster-level				peak-level					mm mm mm		
$p_{\text{FWE-corr}}$	$q_{\text{FDR-corr}}$	k_E	p_{uncorr}		$p_{\text{FWE-corr}}$	$q_{\text{FDR-corr}}$	T	$(Z_\equiv)$	p_{uncorr}			
0.000	0.000	300	0.000		0.000	0.000	9.14	6.84	0.000	44	16	0
					0.000	0.000	8.45	6.51	0.000	48	12	4
					0.041	0.833	5.29	4.64	0.000	38	12	16

table shows 32 local maxima more than 4.0mm apart

Height threshold: T = 5.21, p = 0.000 (0.050)	Degrees of freedom = [1.0, 45.0]
Extent threshold: k = 0 voxels, p = 1.000 (0.050)	FWHM = 9.8 10.6 15.6 mm mm mm; 4.9 5.3 3.9 {voxels}
Expected voxels per cluster, <k> = 2.519	Volume: 880432 = 55027 voxels = 472.2 resels
Expected number of clusters, <c> = 0.05	Voxel size: 2.0 2.0 4.0 mm mm mm; (resel = 102.26 voxels)
FWEp: 5.213, FDRp: 6.702, FWEc: 1, FDRc: 20	

Figure 13.10: *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]*

set-level		cluster-level				peak-level					mm mm mm		
p	c	$p_{\text{FWE-corr}}$	$q_{\text{FDR-corr}}$	k_E	p_{uncorr}	$p_{\text{FWE-corr}}$	$q_{\text{FDR-corr}}$	T	$(Z_\equiv)$	p_{uncorr}			
0.000	5	0.000	1.000	300	0.000	0.000	0.000	9.14	6.84	0.000	44	16	0
						0.000	0.000	8.45	6.51	0.000	48	12	4
						0.006	1.000	5.29	4.64	0.000	38	12	16
		0.002	1.000	3	0.257	0.004	0.548	5.44	4.74	0.000	44	6	28
		0.001	1.000	5	0.148	0.004	0.589	5.40	4.72	0.000	50	2	24
						0.005	1.000	5.34	4.68	0.000	46	6	20
		0.004	1.000	1	0.520	0.005	1.000	5.37	4.70	0.000	34	36	20
		0.002	1.000	3	0.257	0.005	1.000	5.34	4.68	0.000	36	30	20

table shows 16 local maxima more than 4.0mm apart

Height threshold: T = 5.21, p = 0.000 (0.008)	Degrees of freedom = [1.0, 45.0]
Extent threshold: k = 0 voxels, p = 1.000 (0.008)	FWHM = 9.8 10.6 15.6 mm mm mm; 4.9 5.3 3.9 {voxels}
Expected voxels per cluster, <k> = 2.519	Volume: 66128 = 4133 voxels = 69.1 resels
Expected number of clusters, <c> = 0.01	Voxel size: 2.0 2.0 4.0 mm mm mm; (resel = 102.26 voxels)
FWEp: 4.510, FDRp: 6.344, FWEc: 312, FDRc: 312	

Figure 13.11: *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 13.10.*
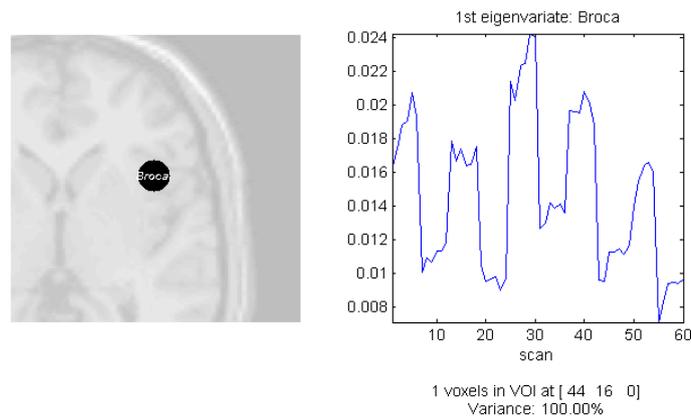
Figure 13.12: *Data extracted from a Volume of Interest (VOI).*

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

13.3.9 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 13.13.

- 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 13.14.

13.3.10 Conjunctions

To perform a conjunction approach across subjects:

- Press the 'Results' button.

- Select the SPM.mat file.
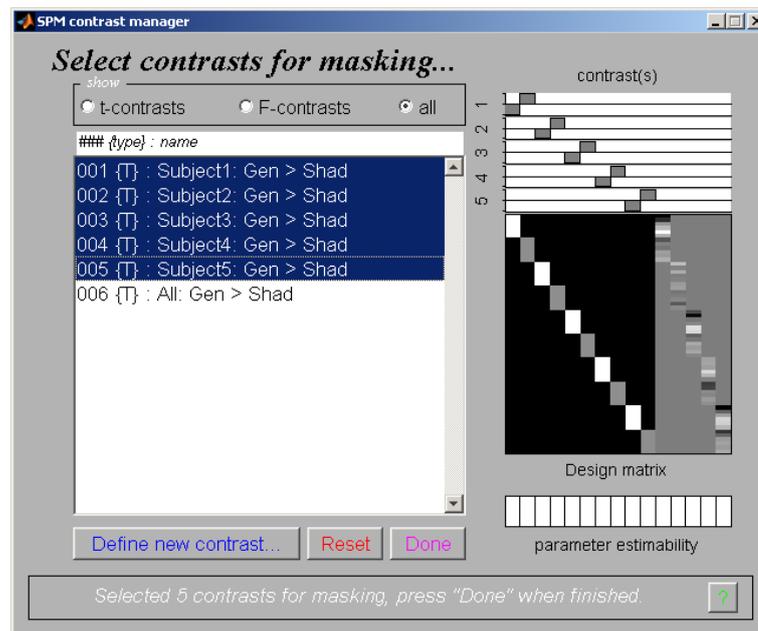
Figure 13.13: *SPM can produce maps based on multiple contrasts by holding down [control] whilst selecting contrasts. This can be used during masking and when making a conjunction inference.*

- Then hold down the [control] button whilst selecting all the individual contrasts. The contrast manager should then appear as in Figure 13.13 (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 orthogonolized with respect to the first contrast specified.

SPM should produce the MIP and table of results shown in Figure 13.15. 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.
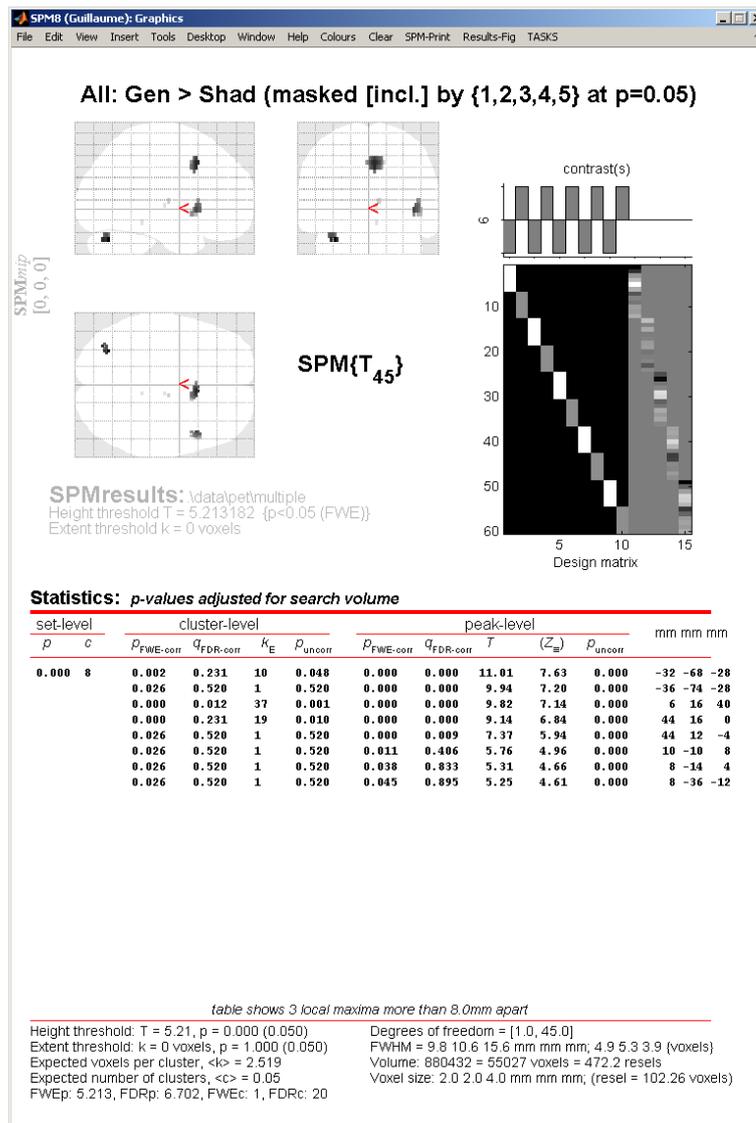
Figure 13.14: *The SPM shows results from the inclusive masking approach. It shows all voxels which are (a) significant at p < 0.05 corrected across all subjects and (b) significant at p < 0.05 uncorrected for each subject individually.*
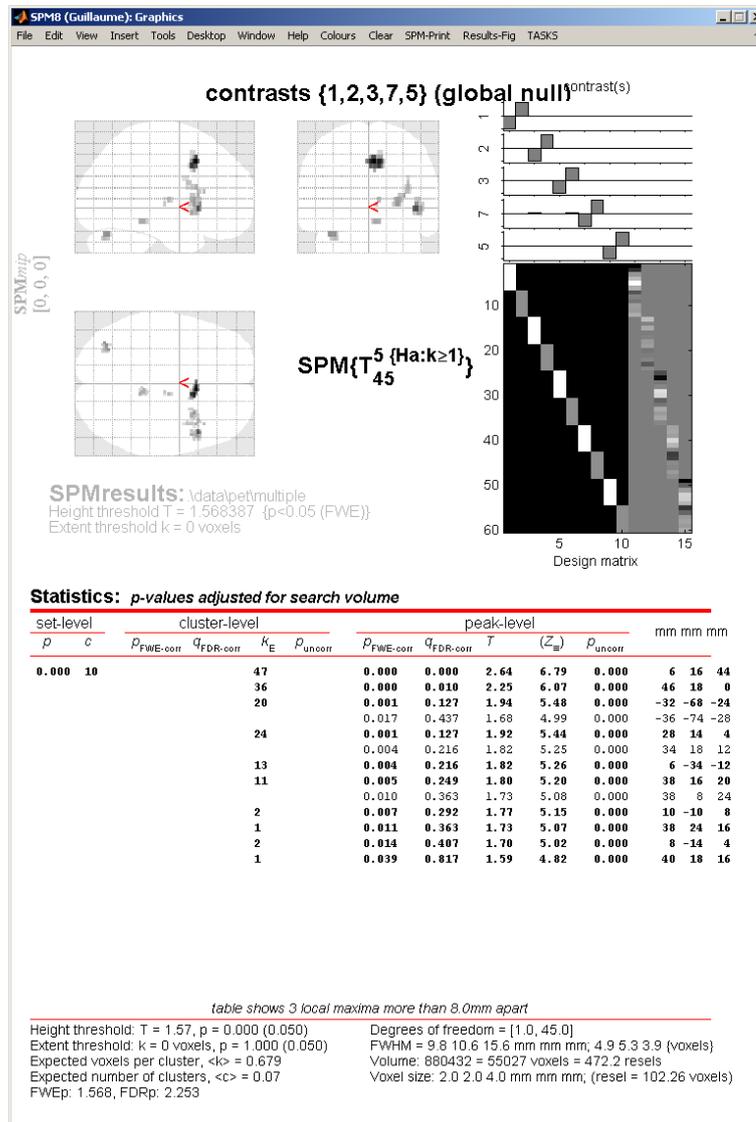
Figure 13.15: *Conjunction SPM.*

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 [24] and [55].

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 14

Dynamic Causal Modeling for fMRI

14.1 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 [20]. 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 14.1 for a summary of the differences between DCM implementations for fMRI and Event-Related Potentials (ERPs).
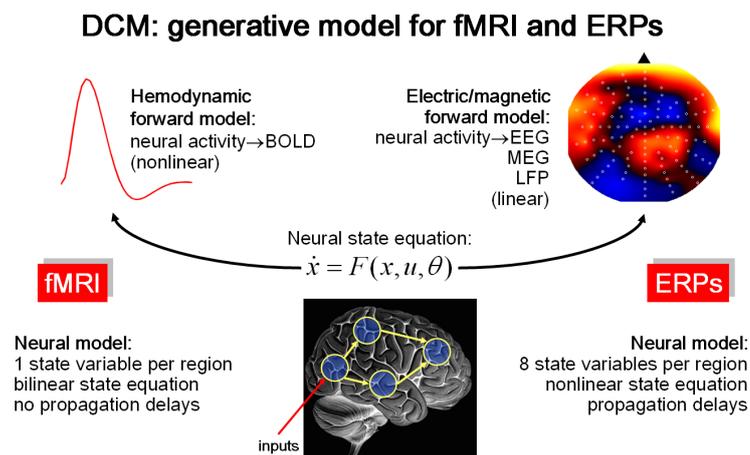
Figure 14.1: *A schematic overview of the differences between between the DCM implementations for fMRI and ERPs (as measured by EEG or MEG). Whereas the state equation of DCM for fMRI is bilinear and uses only a single state variable per region, that for ERPs is more complex and requires 8 state variables 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.*

As in state-space models, two distinct levels constitute a DCM (see Figure 14.2). 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"):

$$
\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ .. \\ \dot{z}_k \end{bmatrix} = \dot{z} = \frac{dz}{dt} = F(z, u, \theta^n) \tag{14.1}
$$

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

$$
\dot{z} = Az + \sum_{j=1}^{m} u_j B_j z + Cu \tag{14.2}
$$

The parameters of this bilinear neural state equation, $\theta^n = \{A, B_1, ..., B_m, C\}$, can be expressed as partial derivatives of F:

$$
\begin{aligned}
A &= \frac{\partial F}{\partial z} = \frac{\partial \dot{z}}{\partial z} \\
B_j &= \frac{\partial^2 F}{\partial z \partial u_j} = \frac{\partial}{\partial u_j} \frac{\partial \dot{z}}{\partial z} \\
C &= \frac{\partial F}{\partial u}
\end{aligned} \tag{14.3}
$$

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 jth input u_j ($k \times k$ matrices $B_1, ..., B_m$), 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 [22]. Briefly summarized, it consists of a set of differential equations that describe the relations between four hemodynamic state variables, using five parameters (θ^h). More specifically, changes in neural activity elicit a vasodilatory signal that leads to increases in blood flow and subsequently to changes in blood volume v and deoxyhemoglobin content q. The predicted BOLD signal y is a non-linear function of blood volume and deoxyhemoglobine content.
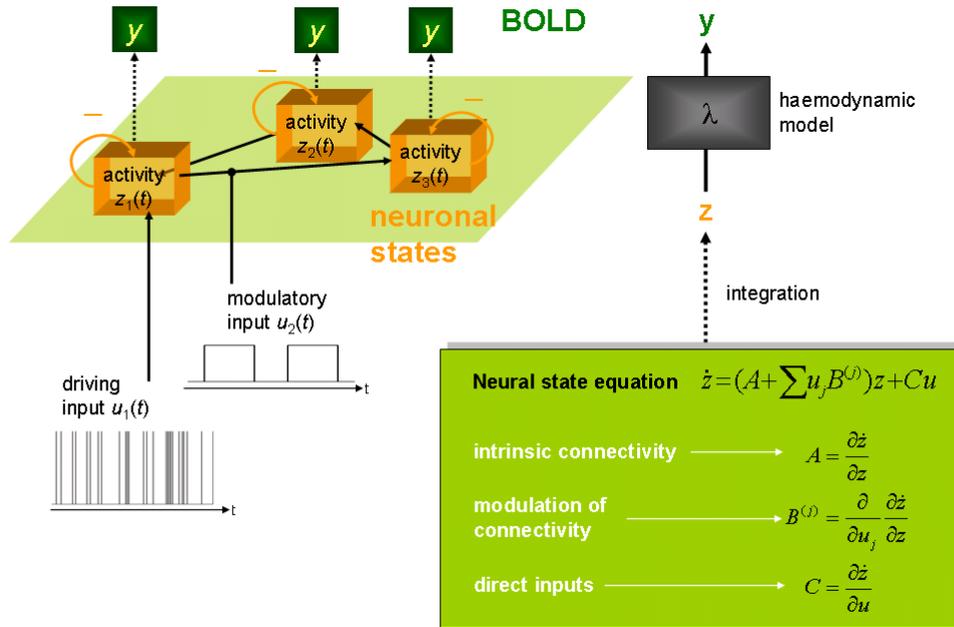
Figure 14.2: *Schematic summary of the conceptual basis of DCM. The dynamics in a system of interacting neuronal populations (orange boxes), which are not directly observable by fMRI, is modeled using a bilinear state equation (grey box). Integrating the state equation gives predicted neural dynamics (z) that enter a model of the hemodynamic response (λ) 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).*

Details of the hemodynamic model can be found in other publications [22]. 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

$$
\begin{aligned}
\dot{x} &= F(x, u, \theta) \\
y &= \lambda(x)
\end{aligned}
\tag{14.4}
$$

For any given set of parameters θ and inputs u, the joint state equation can be integrated and passed through the output nonlinearity λ 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):

$$
y = h(u, \theta) + X\beta + e
\tag{14.5}
$$

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. 14.5). As will be demonstrated below, at the single-subject level, these inferences concern the question of how certain one can be that a particular parameter or, more generally, a contrast of parameters, $c^T \eta_{\theta|y}$, exceeds a particular threshold γ (e.g. zero).

Under the assumptions of the Laplace approximation, this is easy to test (Φ_N denotes the cumulative normal distribution):

$$p(c^T \eta_{\theta|y} > \gamma) = \Phi_N \left(\frac{c^T \eta_{\theta|y} - \gamma}{c^T C_{\theta|y} c} \right) \tag{14.6}$$

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 γ. 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. 14.3).

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

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

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

$$z(\tau) = 0.5 z_0$$
$$= z_0 \exp(a\tau)$$
$$\boxed{a = \ln 2 / \tau}$$

The coupling parameter a thus describes the speed of the exponential change in z(t)

$0.5 z_0$

$\tau = \ln 2 / a$
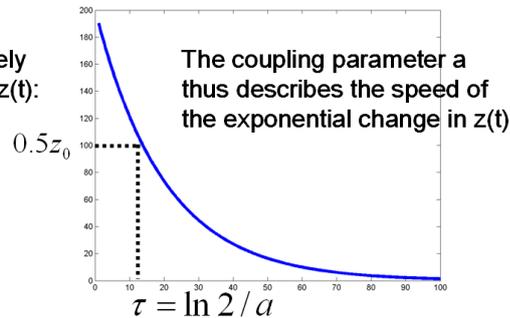
Figure 14.3: *A short mathematical demonstration, using a simple linear first-order differential equation as an example, explaining why the coupling parameters in a DCM are inversely proportional to the half-life of the modelled neural responses and are therefore in units of 1/s = Hertz.*

14.2   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

$$p(\theta|y, m) = \frac{p(y|\theta, m)p(\theta|m)}{p(y|m)} \tag{14.7}$$

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

$$p(y|m) = \int p(\theta|y, m)p(\theta|m)d\theta \tag{14.8}$$

Here, the number of free parameters (as well as the functional form) are considered by the integration. Unfortunately, this integral cannot usually be solved analytically, therefore an approximation to the model evidence is needed. One such approximation used by DCM, and many other models in SPM, is to make use of the Laplace approximation [1].

As shown in [62], 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, θ_p is the prior mean of the parameters, and C_p is the prior covariance):

$$
\begin{aligned}
lnp(y|m) &= accuracy(m) - complexity(m) \\
&= \left[-\frac{1}{2}ln|C_e| - \frac{1}{2}(y - h(u, \eta_{\theta|y}))^T C_e^{-1}(y - h(u, \eta_{\theta|y})) \right] \\
&- \left[\frac{1}{2}ln|C_p| - \frac{1}{2}ln|C_{\theta|y}| + \frac{1}{2}(\eta_{\theta|y} - \theta_p)^T C_p^{-1}(\eta_{\theta|y} - \theta_p) \right]
\end{aligned} \tag{14.9}
$$

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.

Two models can then be compared using the Bayes factor:

$$BF_{ij} = \frac{p(y|m_i)}{p(y|m_j)} \tag{14.10}$$

Given uniform priors over models, the posterior probability for model i is greater 0.95 if BF_{ij} is greater than twenty.

This results in a robust procedure for deciding between competing hypotheses represented by different DCMs. These hypotheses can concern any part of the structure of the modeled system, e.g. the pattern of 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.

14.3 Practical example

The following example refers to the "attention to visual motion" data set available from the SPM web site[2]. This data set was obtained by Christian Buchel and is described in [5].

---

[1]This should perhaps more correctly be referred to as a fixed-form variational approximation, where the fixed form is chosen to be a Gaussian. The model evidence is approximated by the negative free energy, F.

[2]Attention to visual motion dataset: http://www.fil.ion.ucl.ac.uk/spm/data/attention/

The archive contains the smoothed, spatially normalised, realigned, slice-time corrected images in the directory `functional`. The directory `structural` contains a spatially normalised structural image. All processing took place using SPM99, but the image files have been converted into NIfTI format.

Making a DCM requires two ingredients: (i) a design matrix and (ii) the time series, stored in VOI files. The regressors of the design matrix define the inputs for the DCM. Note that this means that the design matrix that is optimal for a given DCM is often somewhat different than the one for the corresponding GLM. DCM does not require the design matrix to be part of an estimated model, however. It just needs to be defined.

14.3.1 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. This is not the main topic of this chapter so you should already be familiar with these procedures, see 1 and 2 for more information. Here are the relevant details for this data set that you need to set up the GLM:

- The onsets for the conditions can be found in the file `factors.mat`. They are named `stat` (static), `natt` (no attention) and `att` (attention) and are defined in scans (not seconds). They are blocks of 10 TRs each.

- The TR is 3.22 seconds.

- There are 360 scans.

Let's specify a batch that will specify the model and estimate it.

1. The analysis directory should include

    (a) A directory named `functional`, which includes the preprocessed fMRI volumes.

    (b) A directory named `structural`, which includes a normalised T1 structural volume

    (c) File `factors.mat`.

    (d) You will also need to make a new directory called `GLM` that will contain the analysis.

2. In MATLAB type

    ```
    >> cd GLM
    >> spm fmri
    ```

3. From the main SPM window, click on the BATCH button.

4. From the SPM menu at the top of the Batch Editor, select "Stats > fMRI model specification".

5. Click DIRECTORY and choose the `GLM` directory that you made above.

6. UNITS FOR DESIGN [SCANS]

7. INTERSCAN INTERVAL [3.22]

8. Click DATA & DESIGN, Choose NEW "SUBJECT/SESSION"

9. Click SCANS and choose all the functional scans `snffM00587_00xx.img`. There should be 360 `*.img` files.

10. Load the MAT-file containing the individual conditions in MATLAB workspace:

    ```
    >> load factors.mat
    ```

    You can look at the loaded variables by typing the variable names. (`stat` = stationary, `natt` = no attention, `att` = attention)

    ```
    >> stat
    >> natt
    >> att
    ```

11. Create 3 NEW: CONDITION under CONDITIONS so that:

    - Condition 1: NAME = `Photic`, ONSETS = `[att natt stat]` and DURATIONS = `10`.
    - Condition 2: NAME = `Motion`, ONSETS = `[att natt]` and DURATIONS = `10`.
    - Condition 3: NAME = `Attention`, ONSETS = `att` and DURATIONS = `10`.

12. From the SPM menu at the top of the Batch Editor, select "Stats > model estimation".

13. For SELECT SPM.MAT, click on the DEPENDENCY button and choose the proposed item (the output from the previous module).

14. You should now be able to press the RUN green arrow at the top of the Batch Editor window. This will specify and estimate the GLM.

14.3.2 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).

1. From the main SPM window, click on the BATCH button.

2. Add a module "SPM > Stats > Contrast manager".

3. For SELECT SPM.MAT, enter the one that has been created in the previous step.

4. Under CONTRAST SESSIONS, choose one NEW: F-CONTRAST and three NEW: T-CONTRAST and enter

    - F-contrast: NAME = `Effects of interest`, F CONTRAST VECTOR = `eye(3)`.
    - T-contrast: NAME = `Photic`, T CONTRAST VECTOR = `[1 0 0]`.
    - T-contrast: NAME = `Motion`, T CONTRAST VECTOR = `[0 1 0]`.
    - T-contrast: NAME = `Attention`, T CONTRAST VECTOR = `[0 0 1]`.

5. Press the RUN green arrow at the top of the Batch Editor window. This will specify and estimate these 4 contrasts.

Here is now a step-by-step example for extracting the V5 time series:

1. Press RESULTS.

2. Select the `SPM.mat` file.

3. Choose the t-contrast for the `Motion` condition.

4. 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 global maxima that looks V5-ish, e.g. `[-36 -87 -3]` (by overlaying the activations onto the normalised structural image you should be able to identify V5 more easily).

8. Press the EIGENVARIATE button.

9. Name of region: `V5`

10. Adjust data for: `Effects of interest` (this effectively 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 (the "1" refers to session 1).

   You can now proceed to select time series for V1 (using the PHOTIC contrast) with an `8 mm` sphere centered on the global maxima (`[0 -93 18]`). Same thing with SPC (using the ATTENTION contrast with a $p \leq 0.001$ uncorrected threshold) and a sphere centered on the local maxima around `[-27 -84 36]`. This will create files `VOI_V1_1.mat` and `VOI_SPC_1.mat`.

14.3.3   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 [20]. 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 14.4, and can be made as follows:

1. Press the large `Dynamic Causal Modelling` button.

2. Choose SPECIFY.

3. Select the `SPM.mat` file you just created when specifying the GLM.

4. Name for `DCM_???.mat`: e.g. `mod_bwd` (for "attentional modulation of backward connection").

5. Select all VOIs in order `VOI_V1_1`, `VOI_V5_1`, `VOI_SPC_1`.

6. Include `Photic`: Yes

7. Include `Motion`: Yes

8. Include `Attention`: Yes
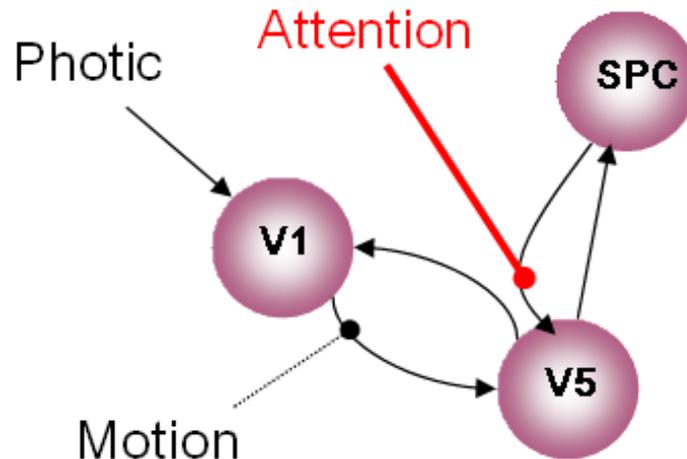
Figure 14.4: *DCM with attentional modulation of backwards connection. Dotted lines denote modulatory connections.*

9. Specify slice timings for each area. The default values are set to the last slice of the data, which was the default in the original DCM version. For sequential (as opposed to interleaved) data, this modelling option allows to use DCM in combination with any TR (slice timing differences) [46]. Here, we proceed with the default values.

10. Enter `0.04` for "Echo Time, TE[s]".

11. Modulatory effects: `bilinear`

12. States per region: `one`

13. Stochastic effects: `no`

14. 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. 14.5.

15. Specify Photic as a driving input into V1. See Fig. 14.5

16. Specify Motion to modulate the connection from V1 to V5. See Fig. 14.5

17. Specify Attention to modulate the connection from SPC to V5. See Fig. 14.5

A polite "Thank you" completes the model specification process. A file called `DCM_mod_bwd.mat` will have been generated.

You can now estimate the model parameters, either by pressing the DCM button again and choosing ESTIMATE, 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.mat`
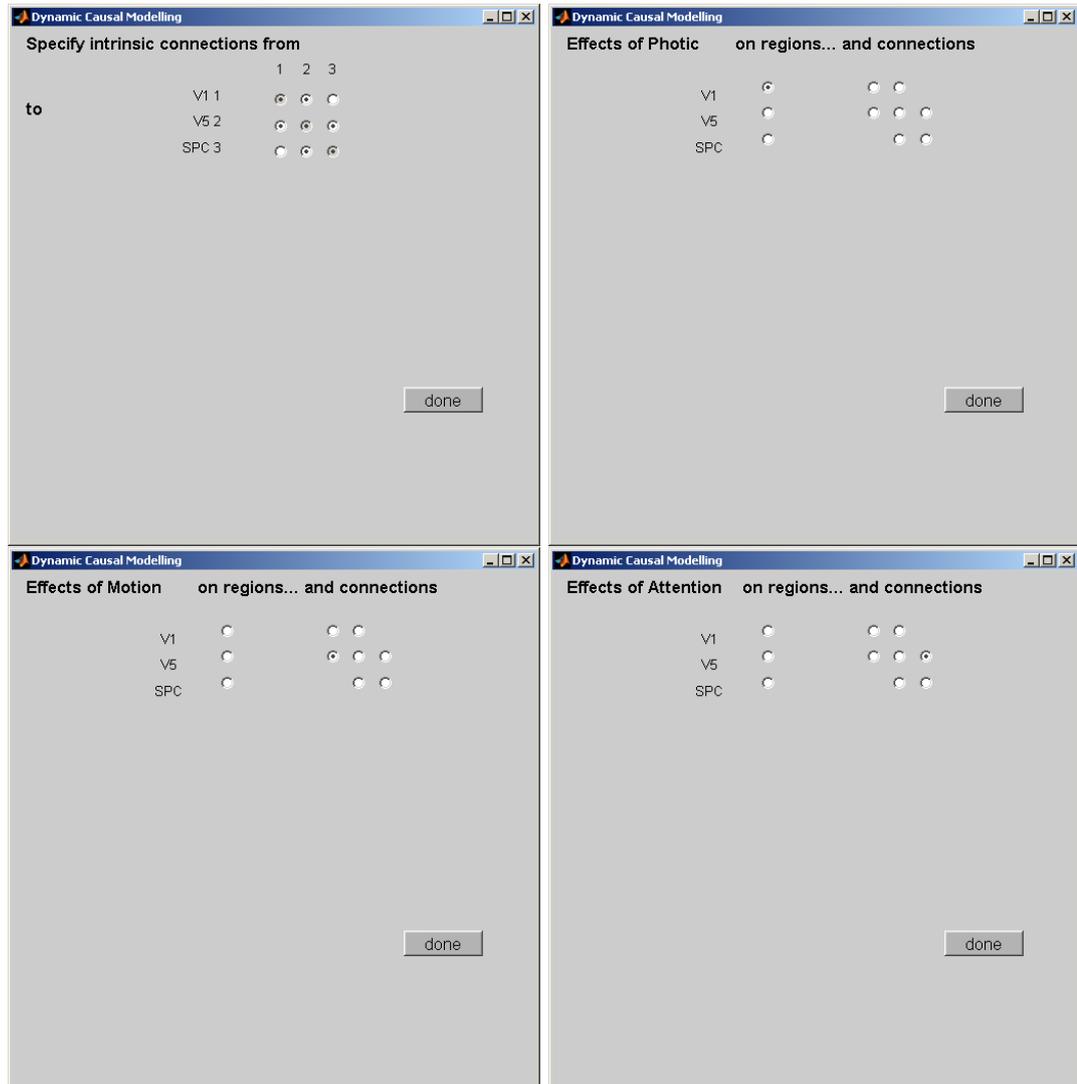
Figure 14.5: *Specification of model depicted in Fig 14.4.* **Top left:** *Filled circles define the structure of the intrinsic connections A such that eg. there are no connections from V1 to SPC or from SPC to V1.* **Top right:** *The filled circle specifies that the input* Photic *connects to region V1.* **Bottom left:** *The filled circle indicates that the input* Motion *can modulate the connection from V1 to V5. This specifies a "modulatory" connection.* **Bottom right:** *The filled circle indicates that* Attention *can modulate the connection from SPC to V5.*
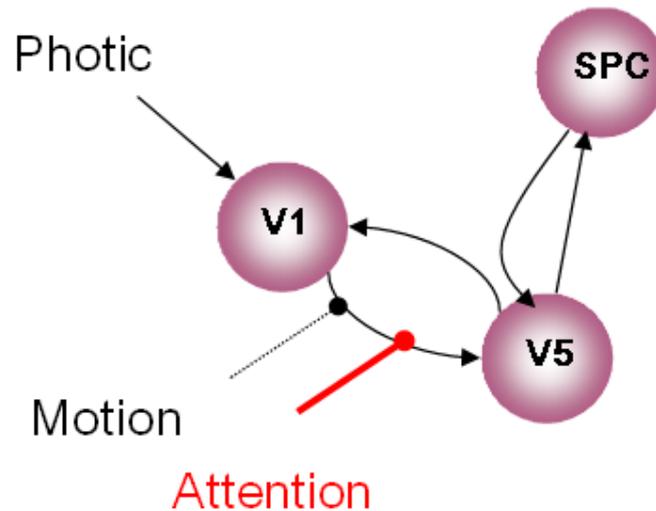
Figure 14.6: *DCM with attentional modulation of forwards connection. Dotted lines denote modulatory connections.*

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.Ep.A` (intrinsic connections), `DCM.Ep.B` (modulatory inputs) and `DCM.Ep.C` (driving inputs).

14.3.4 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 14.6.

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. In the Batch Editor window that opened, fill in the "BMS: DCM" module:

    (a) `Directory`: choose current directory,

    (b) `Data`: add a New Subject with a New Session and select the two models, e.g. in the order `DCM_mod_bwd.mat` and `DCM_mod_fwd.mat`,

    (c) `Inference method`: choose "Fixed effects (FFX)".

4. Press Run (the green triangle in the Batch Editor).

The Graphics window, Fig. 14.8, now shows a bar plot of the model evidence. You can see that our second model is better than the first one.
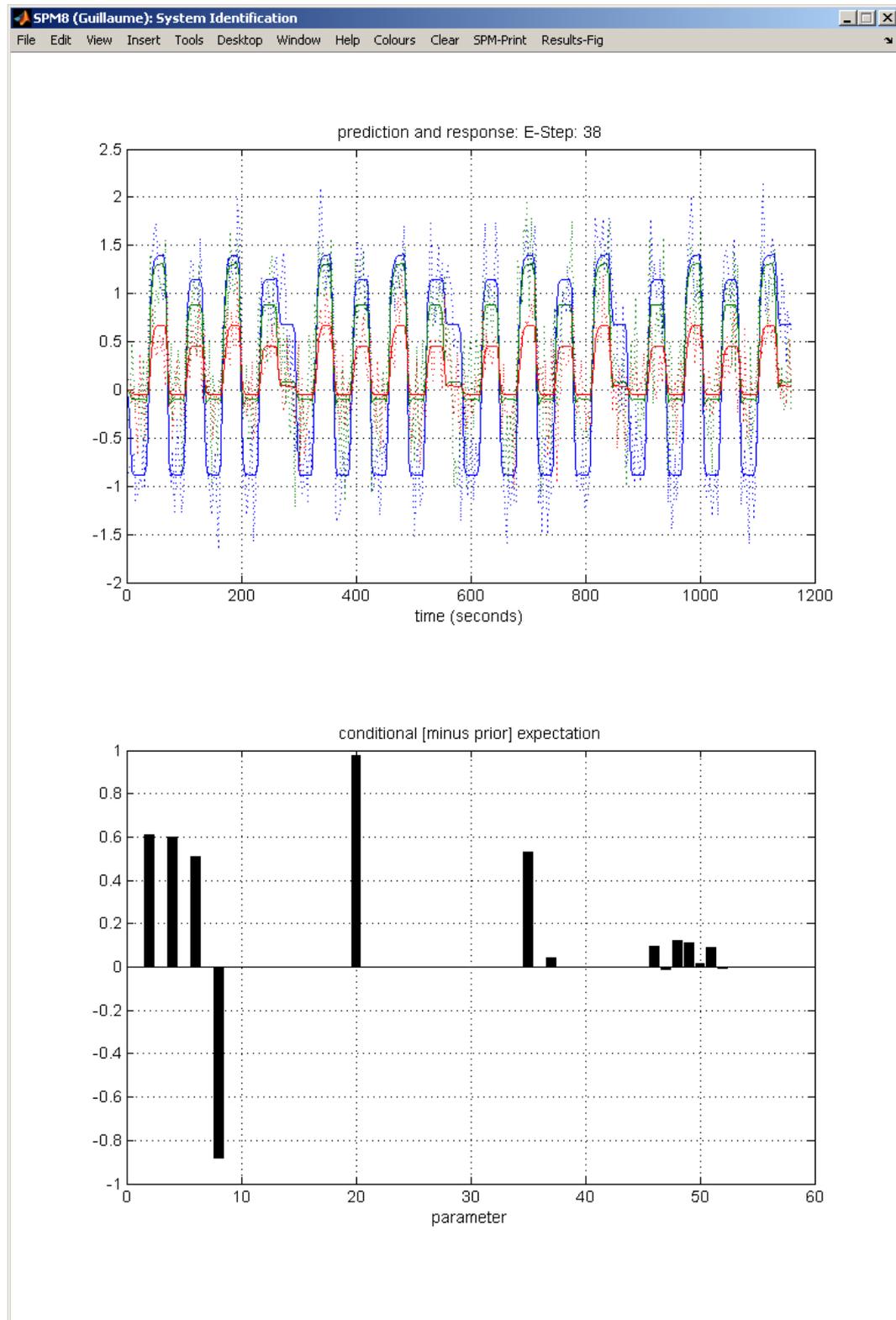
Figure 14.7: *Plot of predicted and observed response, after convergence and conditional expectation of the parameters.*
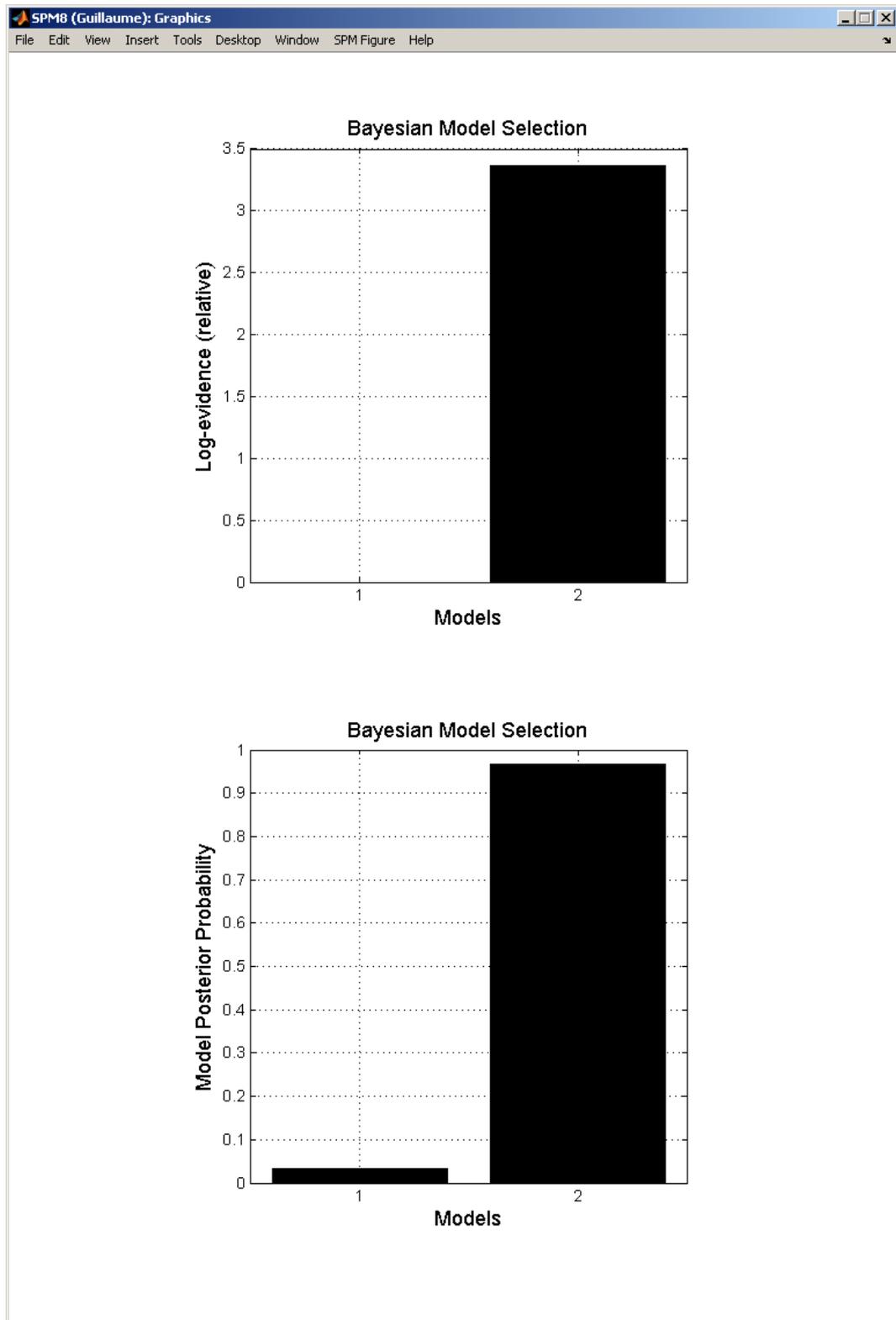
Figure 14.8: *Model 2 (shown in Fig 14.6) is preferred to model 1 (shown in Fig 14.4).*

Chapter 15

Psychophysiological Interactions (PPI)

15.1 Theoretical background

Psychophysiological interactions (PPI) and the related technique of physiophysiological interactions (ΦPI) are based on extensions to statistical models of factorial designs. Table 1 illustrates a classic 2×2 factorial design.

Table 15.1. 2 x 2 factorial design in Table format

		Factor A	
		Level 1	Level 2
Factor B	Level 1	A_1/B_1	A_2/B_1
	Level 2	A_1/B_2	A_2/B_2

The equation for factorial design is given by 15.1.

$$y = (A_2 - A_1)\beta_1 + (B_2 - B_1)\beta_2 + (A_2 - A_1)(B_2 - B_1)\beta_3 + G\beta_4 + \epsilon \qquad (15.1)$$

Notice that this equation includes both of the main effects terms $(A_2 - A_1)\beta_1$ for factor A, and $(B_2 - B_1)\beta_2$ for factor B, as well as the interaction term $(A_2 - A_1)(B_2 - B_1)\beta_3$. It also contains a term for the confounds $G\beta_4$ such as movement parameters, session effects, etc. The inclusion of main effects when estimating interactions is very important, and their inclusion in the design cannot be stressed enough. If the main effects are not included, then we cannot be sure that estimates of the interaction term are not confounded by main effects.

To extend the concept of factorial designs to PPI's the basic idea is to substitute (neural) activity from one cerebral region for one of the factors. Equation 15.2 illustrates this concept after substituting activity in area V1 for factor A.

$$y = V1\beta_1 + (B_2 - B_1)\beta_2 + (V1 \times (B_2 - B_1))\beta_3 + G\beta_4 + \epsilon \qquad (15.2)$$

Similarly, for psychophysiological interactions activity from 2 cerebral regions (V1 and posterior parietal (PP)) are used as the main effects, as shown in equation 15.3

$$y = V1\beta_1 + PP\beta_2 + (V1 \times PP)\beta_3 + G\beta_4 + \epsilon \qquad (15.3)$$

Again, notice that all 3 equations 15.1, 15.2 and 15.3 have 3 terms (aside from confounds and error) – the two main effects and the interaction. Therefore, the design matrix must include at
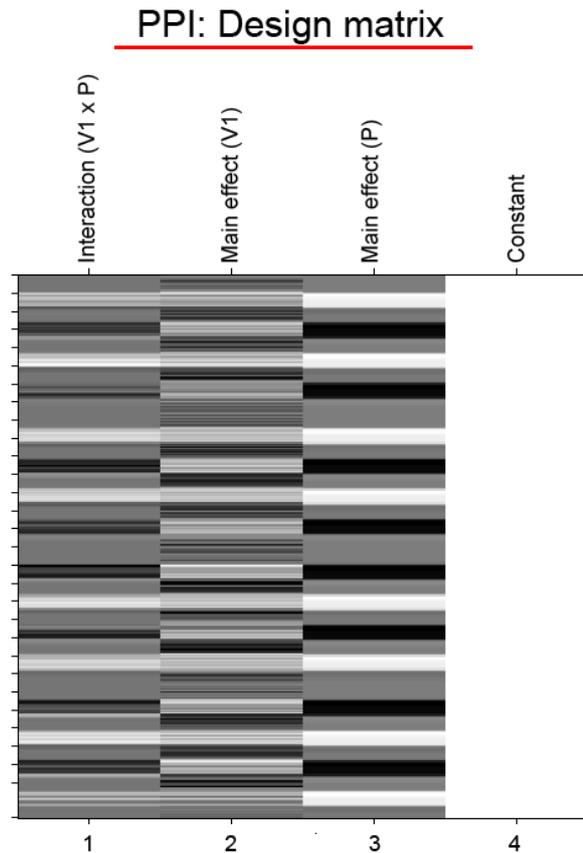
Figure 15.1: *Example design matrix for a PPI (or (ΦPI)). The main effects are BOLD activity from area V1, in column 2, and a psychological vector, e.g., attention vs. no attention (P), in column 3. Inference would typically focus on the interaction term, in column 1, using a contrast vector of* [1 0 0 0]. *In ΦPIs the third column would be BOLD activity from a second source region rather than the psychological factor.*

least 3 columns, one for each main effect and one for the interaction. A basic design matrix for PPI's is shown in Figure 15.1.

Both PPIs and ΦPIs can be conceived of as models of "contribution". PPIs occupy middle-ground between between models of functional vs. effective connectivity [17]. Functional connectivity (FC) is defined as the temporal correlation between spatially separated neurophysiological events [17]. FC analyses are typically model-free and do not specify a direction of influence, i.e., the influence of A on B is indistinguishable from the influence of B on A. In contrast, PPI's are based on regression models, and therefore a direction of influence is chosen based on the model. Effective connectivity (EC) is defined as the influence one neural system has on another [15]. PPIs are closely related to EC models, but because PPIs are generally very simple (i.e., 1 source region and 1 experimental factor, or 2 source regions in the case of ΦPIs) they are very limited models of EC.

The interaction between the source region and experimental context (or two source regions) can be interpreted in 2 different ways: 1) as demonstrating how the contribution of one region to another is altered by the experimental context or task, or 2) as an example of how an area's response to an experimental context is modulated by input from another region, Figure 15.2.
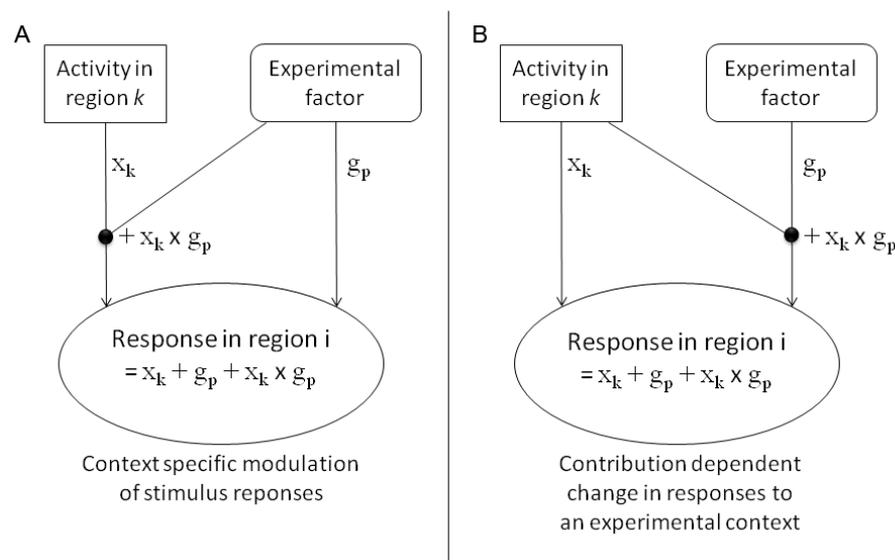
Figure 15.2: Two alternative interpretations of PPI effects. A) The contribution of one area (k) to another (i) is altered by the experimental (psychological) context. B) The response of an area (i) to an experimental (psychological) context due to the contribution of region (k). (Adapted from [17])

15.2 Psycho-Physiologic Interaction Analysis: Summary of Steps

Mechanistically, a PPI analysis involves the following steps.

1. Performing a standard GLM analysis.

2. Extracting BOLD signal from a source region identified in the GLM analysis.

3. Forming the interaction term (source signal x experimental treatment)

4. Performing a second GLM analysis that includes the interaction term, the source region's extracted signal and the experimental vector in the design. The inclusion of the source region's signal and the experimental vector is analogous to including the main effects in an ANOVA in order to make an inference on the interaction.

Forming the proper interaction term turns out to be a challenge because of the unique characteristics of fMRI (BOLD) data in which the underlying neural signal is convolved with a hemodynamic response function. However, interactions in the brain take place at the neural and not the hemodynamic level. Therefore, appropriate models of the interactions require the neural signal, which is not measured directly, but instead must be derived by deconvolving the HRF. The PPI software (`spm_peb_ppi.m`) was developed in order to provide robust deconvolution of the HRF and the proper derivation of the interaction term [29].

15.3 Practical example

The dataset in this exercise is from one subject who was studied in the [7] report and refers to the "attention to motion" dataset available from the SPM website[1]. It has already been described in the previous chapter for DCM.

The goal is to use PPI to examine the change in effective connectivity between V2 and V5 while the subject observes visual motion (radially moving dots) under the experimental treatments of attending vs. not attending to the speed of the dots. The psychophysiologic interaction can be

---

[1]http://www.fil.ion.ucl.ac.uk/spm/data/attention/

conceived of as looking for a significant difference in the regression slopes of V1 vs. V5 activity under the influence of the different attentional states [17].

15.3.1   GLM analysis - Design setup and estimation

This dataset has already been preprocessed (coregistered, normalised and smoothed) using an earlier version of SPM.

1. The analysis directory should include

    (a) A directory named `functional`, which includes the preprocessed fMRI volumes.

    (b) A directory named `structural`, which includes a T1 structural volume

    (c) Files: `factors.mat`,   `block_regressors.mat`,   `multi_condition.mat`   and `multi_block_regressors.mat`.

    (d) You will also need to make 2 empty directories called `GLM` and `PPI` for performing the analyses.

2. In MATLAB type

    ```
    >> cd GLM
    >> spm fmri
    ```

3. Start the Batch system by clicking the BATCH button.

4. From the SPM menu in the Batch window, click STATS and then select the modules FMRI MODEL SPECIFICATION, MODEL ESTIMATION and CONTRAST MANAGER, Figure 15.3.
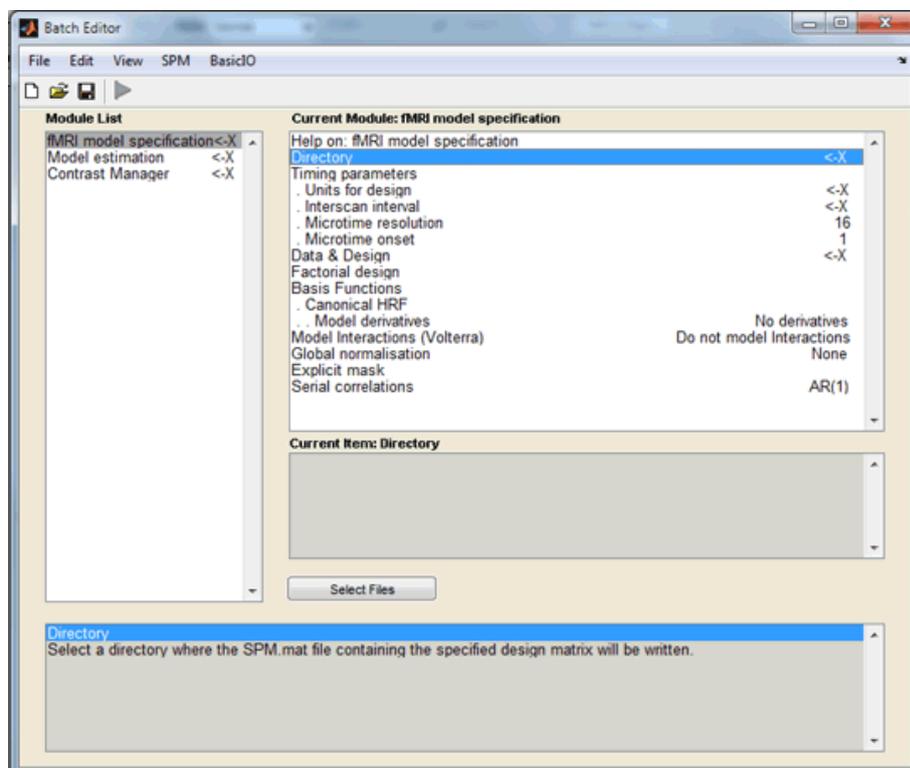
Figure 15.3: Batch Editor showing the FMRI MODEL SPECIFICATION, MODEL ESTIMATION and CONTRAST MANAGER modules.

Fill in the fMRI Model Specification

5. Click DIRECTORY and choose the `GLM` directory that you made above.

6. Units for design [scans]

7. Interscan interval [3.22]

8. Microtime resolution [16]

9. Microtime onset [1]

10. Click Data & Design. Then in the Current Item box click New: Subject/Session, Figure 15.4.
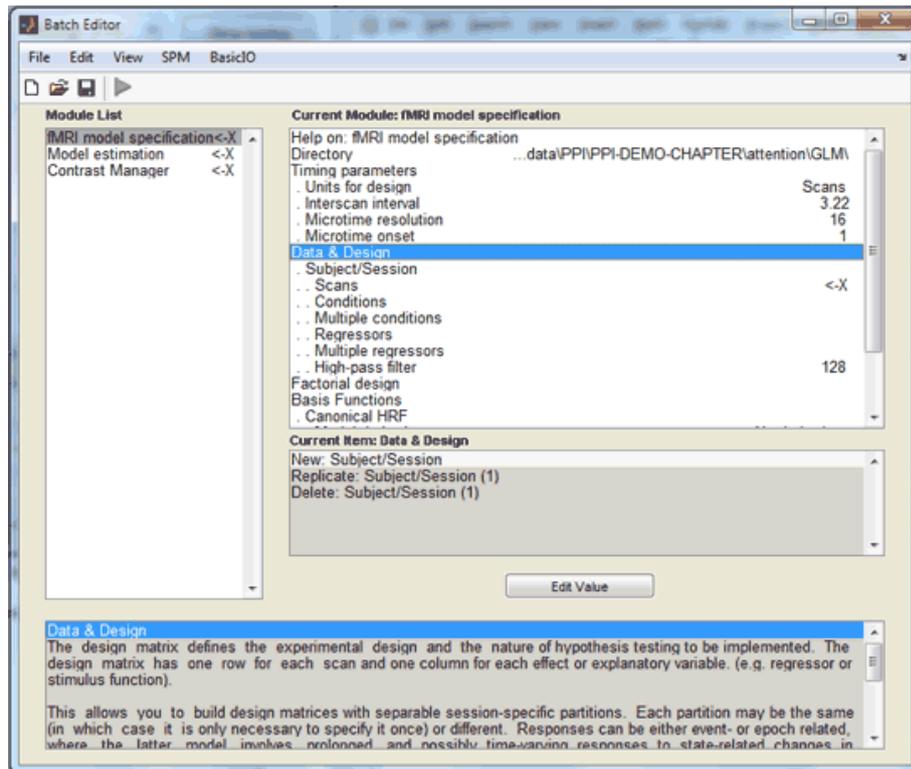
Figure 15.4: *Fill in the Data & Design*

11. Click Scans and choose all the functional scans `snffM00587_00xx.img`. There should be 360 `*.img` files.

12. The experimental conditions can be defined either individually or using a multiple condition `mat`-file. This exercise shows both methods for educational purposes. When doing an actual analysis you can just follow one of the two approaches below.

Define conditions individually

13. Load the mat file containing the individual conditions:

```
>> load factors.mat
```

You can look at the loaded variables by typing the variable names. (`stat` = stationary, `natt` = no attention, `att` = attention)

```
>> stat
>> natt
>> att
```
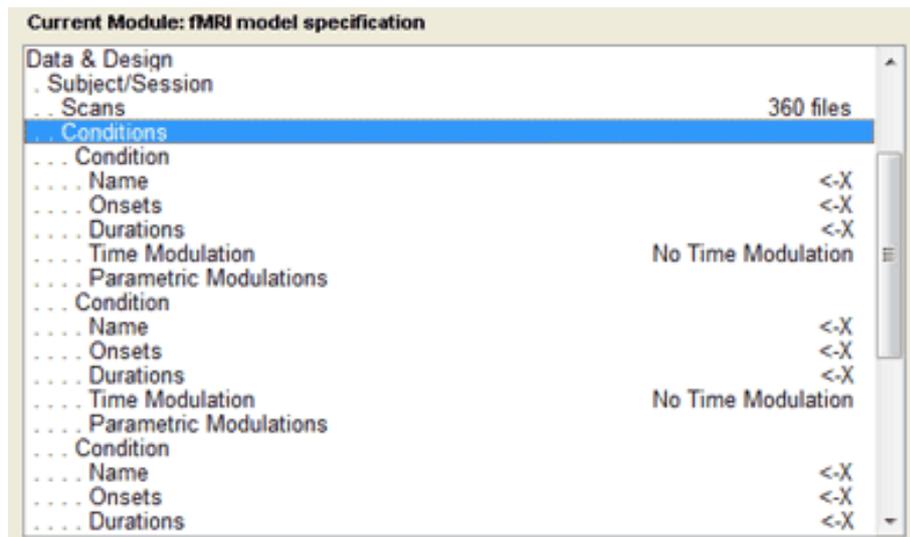
Figure 15.5: CURRENT MODULE *section of the* BATCH EDITOR *showing 3 Conditions to be filled in.*

14. Click CONDITIONS then in the CURRENT ITEM box click NEW: CONDITION 3 times, Figure 15.5.

15. Condition 1: Name = `Stationary`, ONSETS = `stat`, DURATIONS = 10.

16. Condition 2: Name = `No-attention`, ONSETS = `natt`, DURATIONS = 10.

17. Condition 3: Name = `Attention`, ONSETS = `att`, DURATIONS = 10.

18. Next you will enter 3 regressors to model block effects. This will account for the fact that the experiment took place over 4 runs that have been concatenated into a single session to make the PPI analysis easier. *Note: Only 3 of the 4 sessions need to be modeled by block regressors because the fourth one is modeled by the mean column of the design matrix.*

    First load the regressors:

    ```
    >> load block_regressor.mat
    ```

19. Click REGRESSORS then click NEW: REGRESSOR 3 times in the CURRENT ITEM box, Figure 15.6.

20. Regressor 1: NAME = `Block 1`, VALUE = `block1`

21. Regressor 2: NAME = `Block 2`, VALUE = `block2`

22. Regressor 3: NAME = `Block 3`, VALUE = `block3`

Define conditions using multiple condition and multiple regressor files

23. If you would like to look at the organization of the variables in the multiple condition file, first load it.

    ```
    >> load multi_condition.mat
    >> names
    >> onsets
    >> durations
    ```
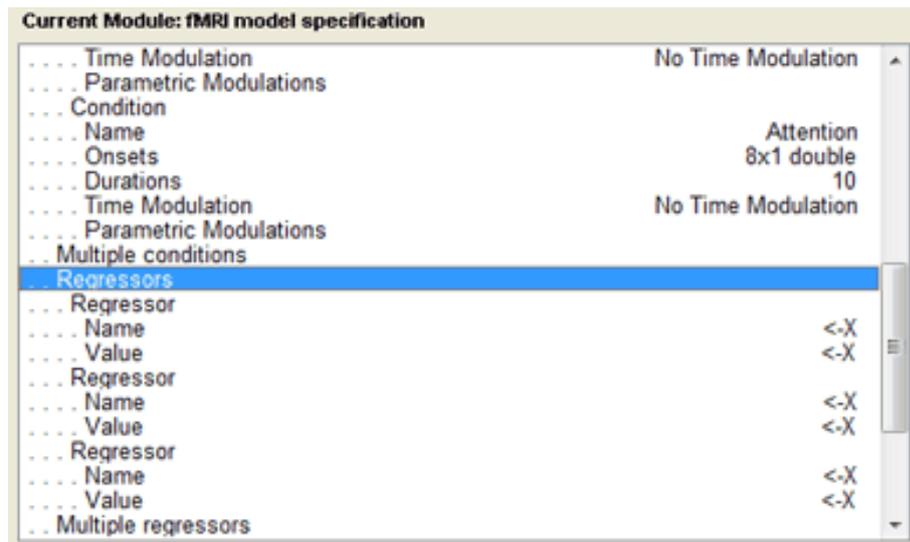
Figure 15.6: CURRENT MODULE *section of the* BATCH EDITOR *showing 3 Regressors to be filled in.*

The variables in a multiple condition file must always be named: 'names', 'onsets', and 'durations'. Notice that these three variables are cell arrays. *(Note: You only need to do this step if you want to look at the organization of the variables. In contrast to defining conditions individually, as shown above, when using a multiple condition file you do not have to load the file in order to enter it into the design.)*

24. To use the multiple conditions file in the design, click MULTIPLE CONDITIONS, then SPECIFY FILES in the Options box and choose the `multi_condition.mat` file.

25. Next you will enter the 3 regressors to model block effects by using a multiple regressor file. To look at the organization of the multiple regressor variable, first load it. *(Again you do not have to load the multiple regressor file in order to use it. This step is just to allow you to examine the file and the variables it contains.)*

```
>> load multi_block_regressor.mat
>> R
```

Notice that this file contains a single variable, `R`, which is a 360 x 3 matrix. The number of rows is equal to the number of scans, and each regressor is in a separate column.

26. To use the multiple regressor file, click MULTIPLE REGRESSORS then select the `multi_block_regressor.mat` file.

Complete the design setup

27. HIGH-PASS FILTER [192] (Note: most designs will use a high-pass filter value of 128. However, this dataset requires a longer high-pass filter in order not to lose the low frequency components of the design.)

28. FACTORIAL DESIGN is not used

29. The BASIS FUNCTION is the CANONICAL HRF as shown and MODEL DERIVATIVES [NO DERIVATIVES]

30. MODEL INTERACTIONS (VOLTERRA): [DO NOT MODEL INTERACTIONS]

31. GLOBAL NORMALISATION [NONE]

32. EXPLICIT MASK [NONE]

33. SERIAL CORRELATIONS [AR(1)]

Model Estimation

34. Under MODEL ESTIMATION click SELECT SPM.MAT then click the DEPENDENCY button and choose FMRI MODEL SPECIFICATION: SPM.MAT FILE. The METHOD should be left as Classical.

Contrast Manager

35. Under CONTRAST MANAGER click SELECT SPM.MAT then click the DEPENDENCY button and choose MODEL ESTIMATION: SPM.MAT FILE

36. Click CONTRAST SESSIONS then click NEW: F-CONTRAST once, and NEW: T-CONTRAST twice from the CURRENT ITEM box.

37. Click CONTRAST VECTORS and then NEW: F CONTRAST VECTOR.

38. The F contrast vector can be entered as [eye(3), zeros(3,4)], which will produce:

```
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
```

39. For the first T-contrast, NAME is `Attention`, and the T CONTRAST VECTOR is `0 -1 1 0 0 0 0` (Note the order of the conditions in the design matrix is: Stationary, NoAttMot and AttMot).

40. For the second T-contrast NAME is `Motion`, and the T CONTRAST VECTOR is: `-2 1 1 0 0 0 0`.

41. Click the SAVE icon on the toolbar and save the batch file.

Design estimation

42. If everything has been entered correctly the RUN button should now be green. Click RUN to estimate the design.

43. The design matrix should look as shown in Figure 15.7, below.

15.3.2   GLM analysis - Results

1. Click RESULTS and select the `SPM.mat` file.

2. Choose the `Attention` contrast

3. Mask with other contrasts [No]

4. Title for comparison [Attention]

5. p value adjustment to control [None]

6. threshold T or p value [0.0001]

7. & extent threshold voxels [10]
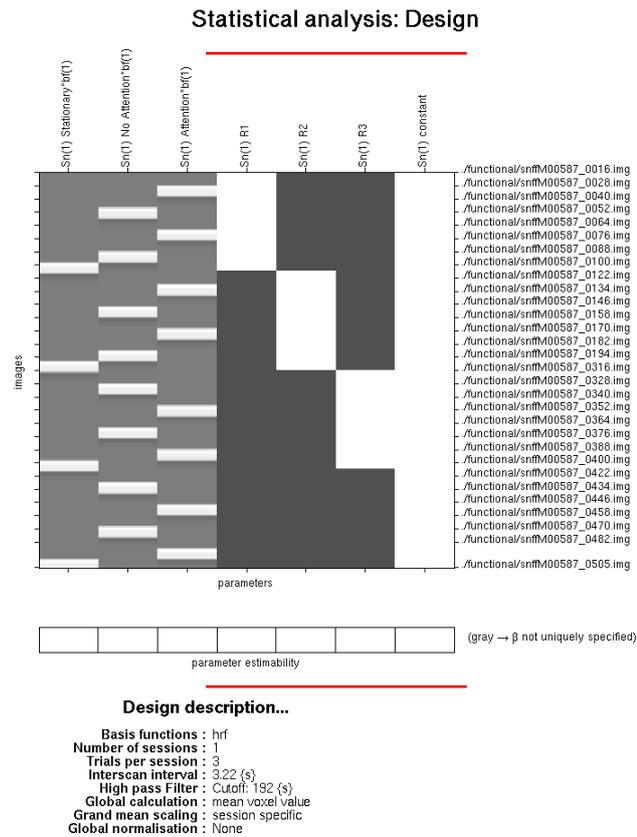
Figure 15.7: *Design matrix*
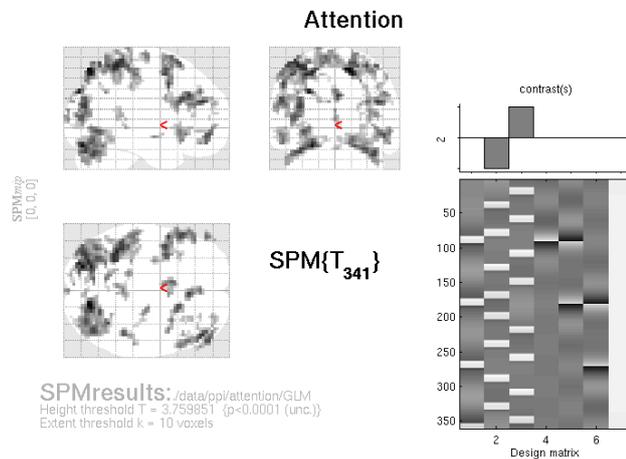
Figure 15.8: *Statistical parametric map for the* `Attention` *contrast*

8. You should see an SPM that looks like the one shown below, Figure 15.8. Note the Superior Parietal and Dorso-Lateral Prefrontal activations, among others. By selecting OVERLAYS → SECTIONS, and selecting the normalised structural image, you should be able to identify the anatomy more accurately.

9. To look at the `Motion` contrast where `Attention` is greater than `No Attention`, click RE-SULTS, choose the `SPM.mat` file and choose the `Motion` contrast.

10. apply masking [Yes]

11. Select contrast for masking: Choose the `Attention` contrast

12. Uncorrected mask p-value [0.01]

13. Nature of Mask: [inclusive]

14. title for comparison: leave as the defaults, which is [Motion (masked [incl.] by Attention at p=0.01)]

15. p value adjustment to control [FWE]

16. threshold T or p value [0.05]

17. & extent threshold voxels [3]

18. The masked `motion` contrast on the glass brain is shown below in Figure 15.9.
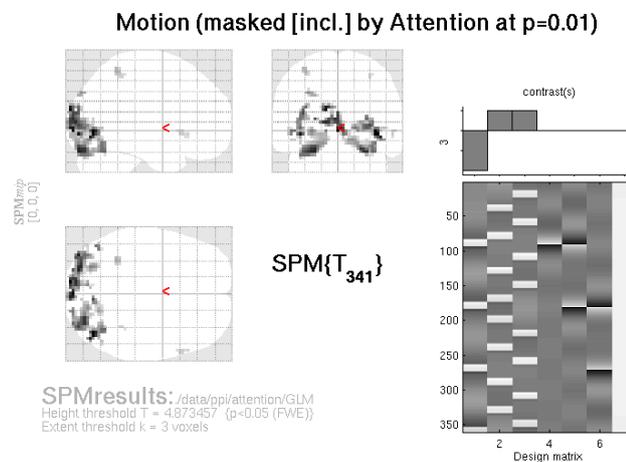
Figure 15.9: *Statistical parametric map for the* `Motion` *contrast inclusively masked by the Attention contrast*

15.4   GLM analysis - Extracting VOIs

1. First select the `Motion` contrast, but do not include masking. Use a p-value adjustment of FWE with height threshold of 0.05 and a cluster threshold of 3.

2. Go to point [15 -78 -9]

3. Press `eigenvariate`

4. Name of region [V2]

5. Adjust data for [effects of interest]

6. VOI definition [sphere]

7. VOI radius(mm) [6]

This saves the extracted VOI data in the file `VOI_V2_1.mat` in the working directory, and displays Figure 15.10, below. The left side shows the location on a standard brain. The right side shows the first eigenvariate of the extracted BOLD signal.
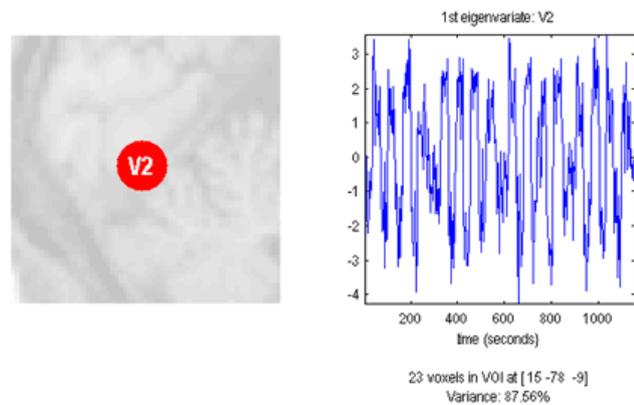
Figure 15.10: *First eigenvariate of the extracted BOLD signal in V2*
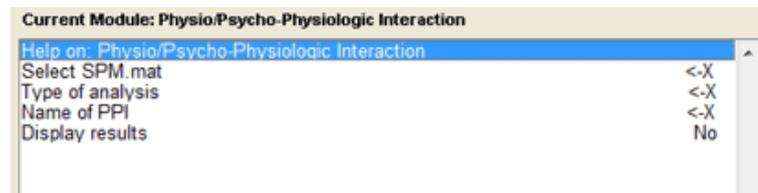
Figure 15.11: *Physio/Psycho-Physiologic module in the Batch Editor*

15.5 PPI analysis - Create PPI variable

1. PPIs can be calculated either by pressing the PPIs button in the SPM MENU window, or by selecting the PHYSIO/PSYCHO-PHYSIOLOGIC menu item from the SPM → Stats menu of the BATCH EDITOR. This example uses the BATCH EDITOR, Figure 15.11.

2. Choose the SPM.MAT file in the GLM directory.

3. Type of analysis: Choose PSYCHO-PHYSIOLOGIC INTERACTION, Figure 15.12.
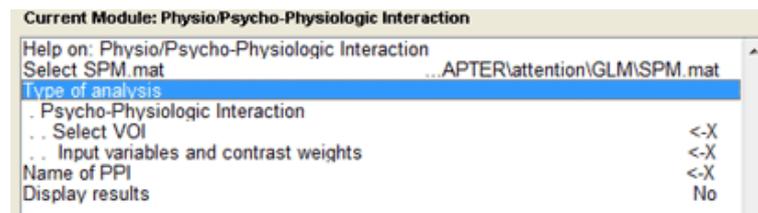
Figure 15.12: *Specifying a Psycho-Physiologic interaction.*

4. Select VOI: Choose VOI_V2_1.mat

5. Input variables and contrast weights: Must be specified as an n x 3 matrix, where n is the number of conditions included in the PPI. The first column of the matrix indexes SPM.Sess.U(i). The second column indexes SPM.Sess.U(i).nameii. It will generally be a 1 unless there are parametric effects. The third column is the contrast weight. In order to include Attention - No-attention in the PPI, recall that the conditions were entered as: Stationary, No-attention, Attention, therefore the matrix should be.

   [2 1 -1; 3 1 1]

6. Name of PPI [V2x(Att-NoAtt)]

7. Display results: Yes

After a few seconds the PPI will be calculated and a graphical window will appear, Figure 15.13. In the upper left, the details of the PPI setup calculation are given including the name of the PPI, the chosen VOI file, and the included conditions and their contrast weights. The main central graph shows the original BOLD signal (actually the eigenvariate) in blue and the neuronal or deconvolved signal in green. These will look quite similar for block design data. The graph in the lower left shows the task condition plot, dotted green line, and the convolved task conditions (psych variable). In the lower right the PPI interaction term is plotted.
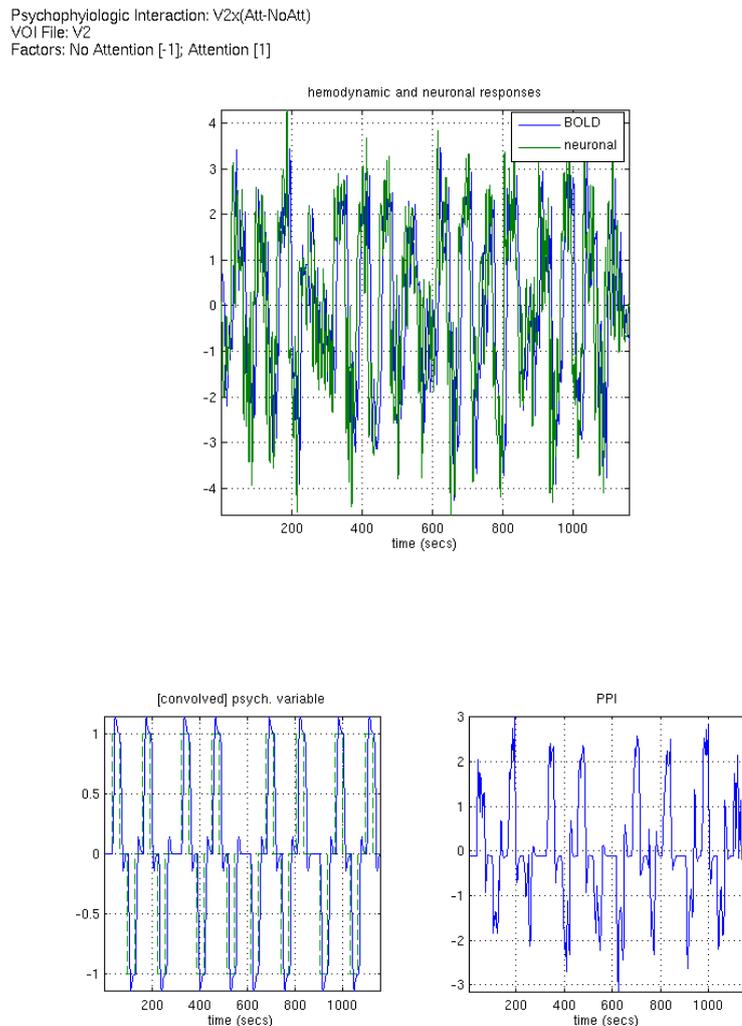
Figure 15.13: *PPI output graphics*

The PPI calculation will create a file `PPI_V2x(Att-NoAtt).mat` in the current working directory. It contains the variable `PPI.ppi` (the interaction term), `PPI.Y` (the original VOI eigenvariate) and `PPI.P` (the `Attention - No Attention` task vector). You will use these vectors in setting up your psychophysiologic interaction GLM analysis. See `spm_peb_ppi` for a full description of the `PPI` data structure.

15.5.1   PPI GLM analysis - Design setup and estimation

1. Copy the file `PPI_V2x(Att-NoAtt)` MAT-file to the `PPI` directory that you created at the start of this exercise.

2. Change directory to the new one, i.e. `cd PPI`

3. At the MATLAB prompt type

```
>> load PPI_V2x(Att-NoAtt)
```

4. In the BATCH EDITOR setup another GLM analysis by choosing the modules FMRI MODEL SPECIFICATION, MODEL ESTIMATION and CONTRAST MANAGER as you did above, and fill it in as follows.

5. Directory: Choose the PPI directory

6. Units for design [scans]

7. Interscan interval [3.22]

8. Add a NEW: SUBJECT/SESSION under DATA & DESIGN

9. Click SCANS and choose all the functional scans snffM00587_00xx.img. There should be 360 *.img files.

10. Click NEW: REGRESSOR and add 6 regressors.

11. Regressor 1: NAME = PPI-interaction, VALUE = PPI.ppi

12. Regressor 2: NAME = V2-BOLD, VALUE = PPI.Y

13. Regressor 3: NAME = Psych_Att-NoAtt, VALUE = PPI.P

14. Regressor 4: NAME = Block 1, VALUE = block1

15. Regressor 5: NAME = Block 2, VALUE = block2

16. Regressor 6: NAME = Block 3, VALUE = block3

17. High Pass Filter [192]

Model Estimation

18. Under MODEL ESTIMATION click SELECT SPM.MAT then click the DEPENDENCY button and choose FMRI MODEL SPECIFICATION: SPM.MAT FILE. The METHOD should be left as Classical.

Contrast Manager

19. Under CONTRAST MANAGER click SELECT SPM.MAT then click the DEPENDENCY button and choose MODEL ESTIMATION: SPM.MAT FILE

20. Click CONTRAST SESSIONS then click NEW: T-CONTRAST

21. T-contrast, NAME: PPI-Interaction, vector: 1 0 0 0 0 0 0

22. Save the batch file.

23. Run

The design matrix is shown below, Figure .
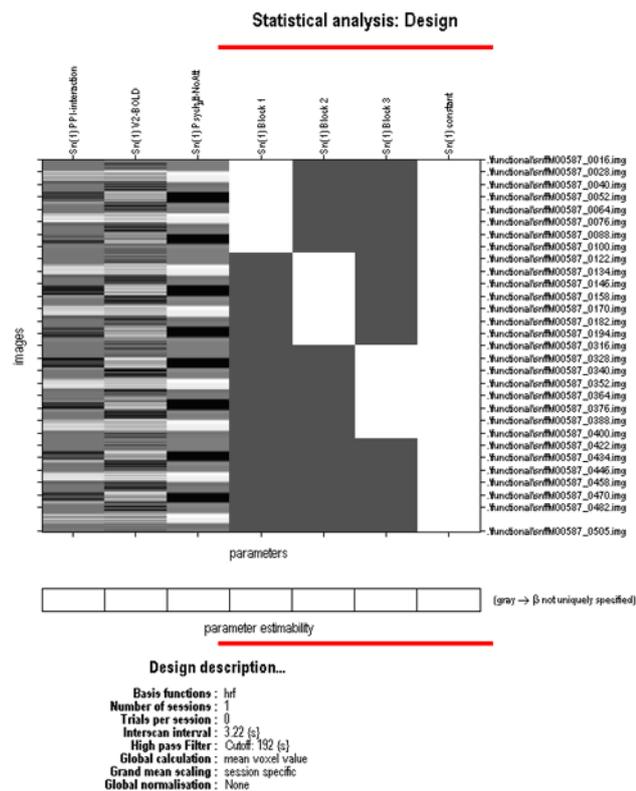
Figure 15.14: *Design matrix for the PPI analysis*

15.5.2   PPI analysis - Results

1. Press the RESULTS button and select the SPM.mat file in the PPI directory.

2. Choose the PPI-Interaction contrast

3. apply masking [No]

4. title for comparison [PPI-Interaction]

5. p value adjustment to control [None]

6. threshold T or p value [0.01]

7. & extent threshold voxels [10]

8. You should see an SPM that looks the same as the one shown below in the top part of Figure 15.15. The resulting SPM shows areas showing differential connectivity to V2 due to the effect of attention vs. no attention conditions. The effect in this subject is weak.

15.5.3   PPI analysis - Plotting

1. One region showing the psychophysiologic interaction is the V5region, which is located at [39 -72 0] in this subject. Move the cursor to this point to view the area of activation, as shown below, in the bottom half of Figure 15.15.

2. In order to plot the PPI graph showing the effect of attention, you need to extract a VOI from the V5 region. To do this, you will return to the original GLM analysis.

3. Click Results, then select the GLM analysis SPM.mat file and the Motion contrast.

4. apply masking [No]
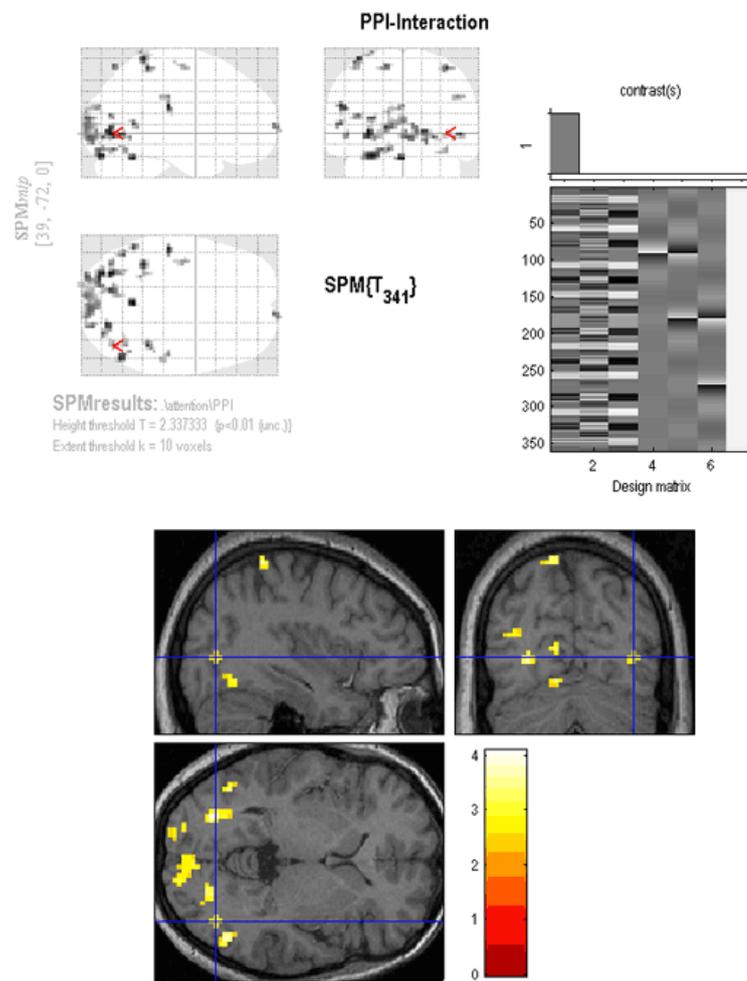
Figure 15.15: *PPI results*

5. title for comparison [Motion]

6. p value adjustment to control [None]

7. threshold T or p value [0.001]

8. & extent threshold voxels [3]

9. Go to point [39 -72 0]

10. Press eigenvariate

11. Name of region [V5]

12. Adjust data for [effects of interest]

13. VOI definition [sphere]

14. VOI radius(mm) [6]

15. Now you will create 4 PPIs (Follow the steps under section 15.5, Create PPI Variable, above). By using the PPI software machinery to create the interaction vectors, rather than just multiplying the extracted V2 and V5 eigenvariates by the behavioral vectors, the PPI vectors will be formed properly.

16. `V2xNoAttention` (Use the V2 VOI and include `No-Attention` with a contrast weight of 1, do not include `Stationary`, `Attention`)

17. **V2xAttention** (Use the V2 VOI and include `Attention` with a contrast weight of 1, do not include `Stationary`, `No-Attention`)

18. **V5xNoAttention** (Use the V5 VOI and include `No-Attention` with a contrast weight of 1, do not include `Stationary`, `Attention`)

19. **V5xAttention** (Use the V5 VOI and include `Attention` with a contrast weight of 1, do not include `Stationary`, `No-Attention`)

20. Load the PPIs you just created with the following commands at the MATLAB prompt:

```
>> v2noatt = load('PPI_V2xNoAttention');
>> v2att   = load('PPI_V2xAttention.mat');
>> v5noatt = load('PPI_V5xNoAttention.mat');
>> v5att   = load('PPI_V5xAttention.mat');
```

21. Plot the PPI datapoints with the following commands at the MATLAB prompt:

```
>> figure
>> plot(v2noatt.PPI.ppi,v5noatt.PPI.ppi,'k.');
>> hold on
>> plot(v2att.PPI.ppi,v5att.PPI.ppi,'r.');
```

22. To plot the best fit lines type the following first for `NoAttention`

```
>> x  = v2noatt.PPI.ppi(:);
>> x  = [x, ones(size(x))];
>> y  = v5noatt.PPI.ppi(:);
>> B  = x\y;
>> y1 = B(1)*x(:,1)+B(2);
>> plot(x(:,1),y1,'k-');
```

23. Then for `Attention`

```
>> x = v2att.PPI.ppi(:);
>> x = [x, ones(size(x))];
>> y = v5att.PPI.ppi(:);
>> B = x\y;
>> y1 = B(1)*x(:,1)+B(2);
>> plot(x(:,1),y1,'r-');
>> legend('No Attention','Attention')
>> xlabel('V2 activity')
>> ylabel('V5 response')
>> title('Psychophysiologic Interaction')
```
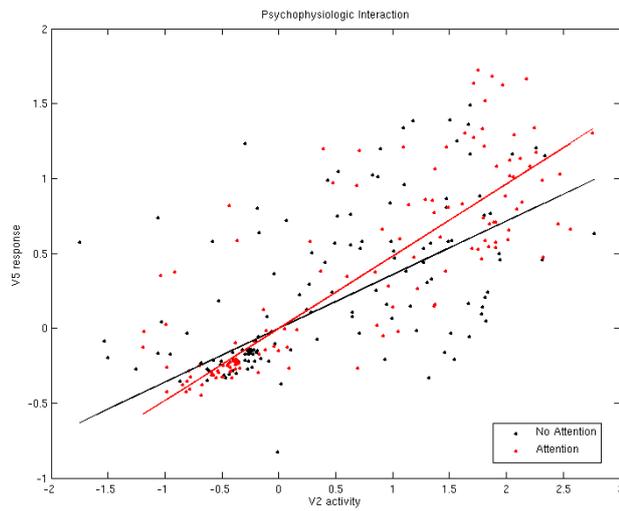
Figure 15.16: *Graph demonstrating PPI interaction.*

Chapter 16

Bayesian Model Inference

This chapter describes the use of SPM's Bayesian Model Inference capabilities. For a fuller background on this topic see [61]. We illustrate the methods using a DCM for fMRI study of the languange system.

16.1 Background

The neuroimaging data derive from an fMRI study on the cortical dynamics of intelligible speech [47]. This study applied dynamic causal modelling of fMRI responses to investigate activity among three key multimodal regions: the left posterior and anterior superior temporal sulcus (subsequently referred to as regions P and A respectively) and pars orbitalis of the inferior frontal gyrus (region F). The aim of the study was to see how connections among regions depended on whether the auditory input was intelligible speech or time-reversed speech.

The basic DCM, from which all other models were derived, is shown in figure 16.1. Auditory input enters region P and the three areas have full intrinsic connectivity. The modulatory input, encoding whether or not the auditory stimulus was speech or reversed speech, was then allowed to modulate a subset of connections in the model. These are the forward and backward connections between P and F, and the forward and backward connections between P and A. As these are either present or absent this results in $2^4 = 16$ different DCMs.

16.2 Data

An archive containing 16 DCMs for each of 12 subjects can be downloaded from the SPM web page. This archive is called `dcm_bms.zip`. When you extract the data onto your computer a number of subdirectories will be created - one for each of the 12 subjects. The 16 DCMs for each subject are then available in these subject-specific directories. You can load one of these into SPM and examine the information contained therein.

These DCM files contain the usual information eg. the original time series from each region of interest are available in `DCM.xY(1)` for region 1, wherein `DCM.xY(1).name='PSTS_6'` indicates this is the posterior temporal region. The estimated parameter values are available in `DCM.Ep`. You should note that these DCMs were specified and estimated using SPM revision 3894 (from May 2010) and that these DCM structures differ from earlier SPM releases.

Also in the Zip archive is a file called `model_space.mat`. If you `load model_space`, you will see that it contains a data structure called `subj` with subfields 'sess' and then 'model'. If you type eg. `subj(1).sess(1).model(1)` you will see four further subfields containing information about the first DCM for subject 1. This comprises the filename (fname), the free energy approximation to the model evidence (F), posterior mean parameters (Ep), and the posterior covariance of parameters (Cp).

The use of a 'model space' file makes use of SPMs Bayesian model comparison (BMC) routines much simpler. If this file is not specified it will be automatically generated (from the DCM files) the first time you use the BMC routines (see below). Alternatively, you can easily create your own model space file. To get the current file to work on your system you will need to change all of
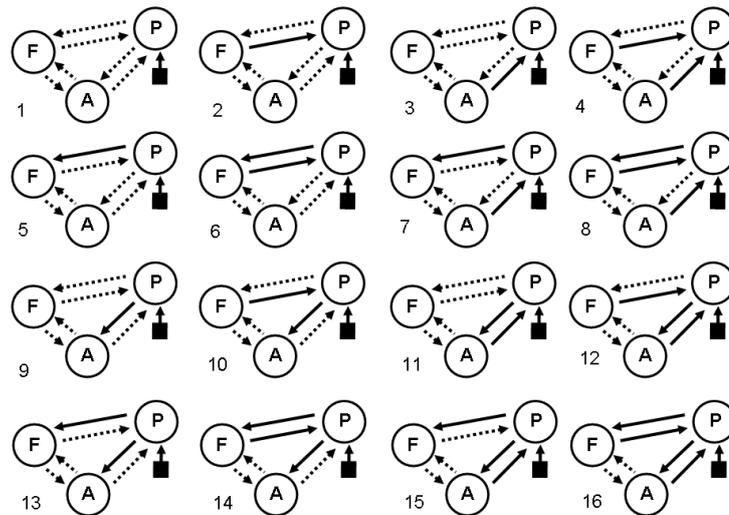
Figure 16.1: All DCMs were fully connected ie. there were endogenous connections between all three regions (dotted lines) (i) left posterior temporal sulcus (region P), (ii) left anterior superior temporal sulcus (region A) and (iii) pars orbitalis of the inferior frontal gyrus (region F). Auditory input enters region P. The sixteen models differ in their modulatory connectivity (solid lines)

the filenames (fname) so that they correspond to the positions of the DCM files in your filespace. You can do this with the `model_space_filenames` function (also provided in the Zip archive).

16.3   Analysis

After unzipping the archive, correct the model space filenames using the command `subj=model_space_filenames(subj,new_base_dir)` where `new_base_dir` is the name of the directory where you have unzipped the archive. This should be something like `'C:\blah\blah\blah\dcm-base-files'`. Then save `subj` back in the model space file using `save model_space subj`.

16.3.1   Single Family

Now open SPM and in the Menu window go to Batch, SPM, Stats, Bayesian Model Selection, BMS:DCM. This will open SPM's batch editor. Select an appropriate directory (eg. where you unzipped the archive), highlight Load model space and select the `model_space.mat` file. For inference method select 'FFX'. Save the batch job as `ffx_all_models.mat`, then press the green play button to run the job. This will produce the figure 16.2, showing that model 6 is the best model.

We can now go back and load the `ffx_all_models.mat` job in the batch editor (press the Batch button) and change the inference methods to RFX. This will produce something like the results in figure 16.3 (note that the RFX approach uses a sampling procedure with a different random initial seed on each run, so the results can vary slighlty from run to run). Again, model 6 is the best model, but not by much. These RFX results will be stored in the same `BMS.mat` file as the FFX results.

16.3.2   Bayesian Model Averaging

Now go back into the batch editor and reload the `ffx_all_models.mat` job. Highlight BMA, and select Choose family (instead of 'Do not compute'). Accept the 'Winning Family' option. The BMA results will be saved in the same BMS.mat file as the previous analyses. Now go ahead
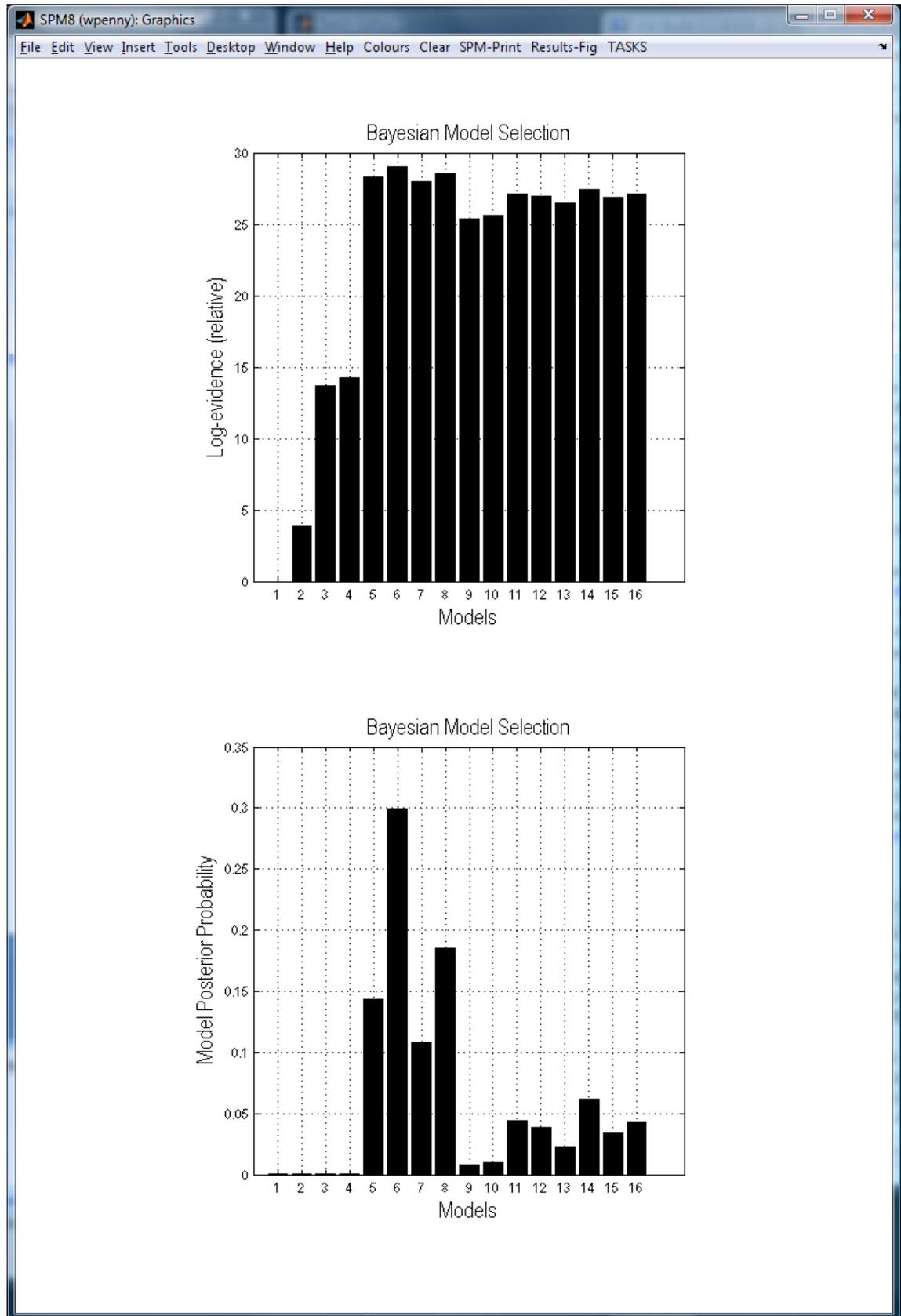
Figure 16.2: Fixed effects model inference
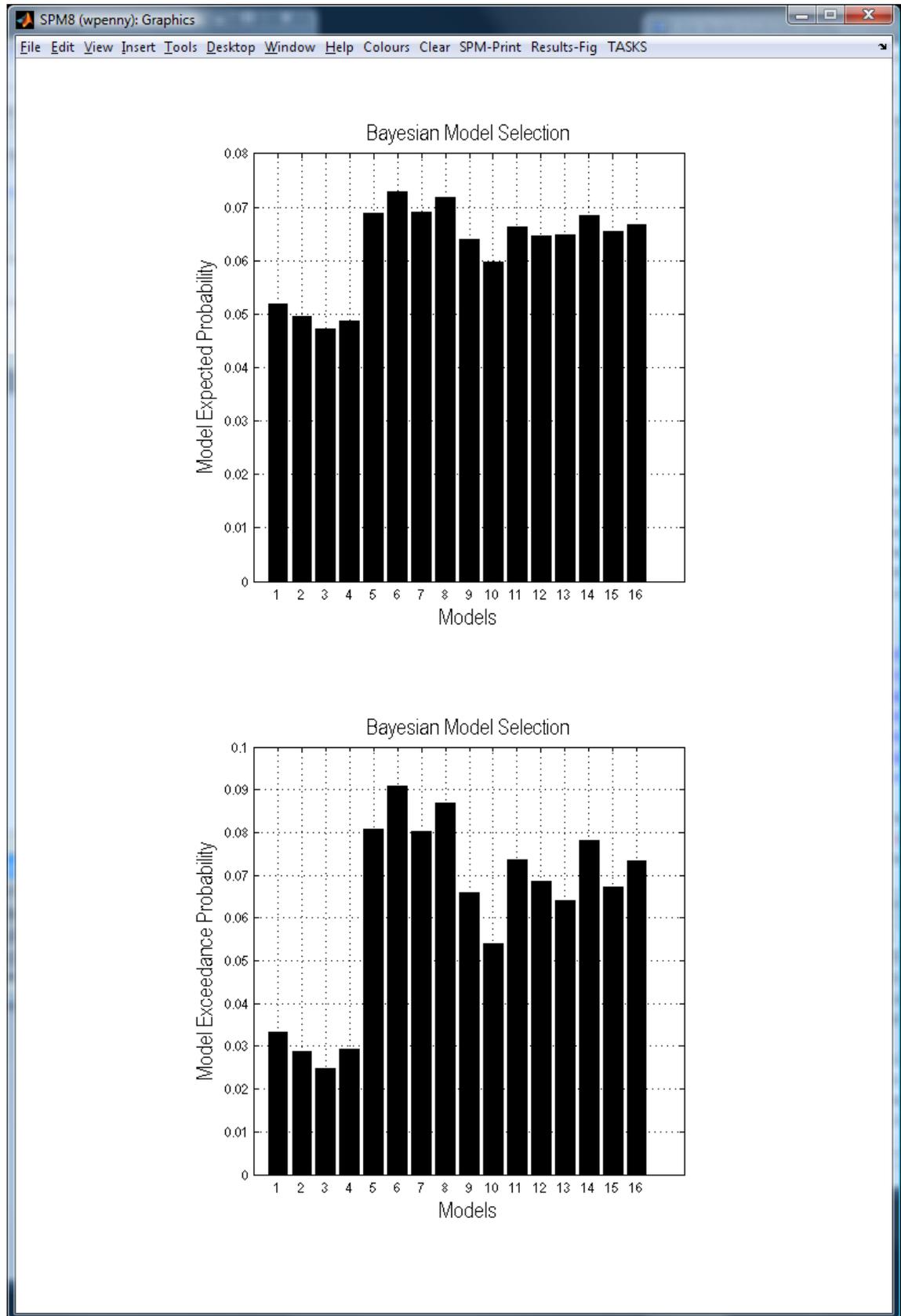
Figure 16.3: Random effects model inference

and press the green play button. SPM will do the FFX model inference (again), but will also implement a weighted average of the model parameters where the weights are given by the evidence for each model, as described in [61]. After the averaging is complete, SPM will report the number of models in Occams window. This should be 10 models (models 5,6,7,8,11,12,13,14,15,16).

To look at the BMA results, go to the Menu window and press the Dynamic Causal Modelling button. Then select Average, select BMA, and then the BMS.mat file just created. Then select FFX for the inference method. If you then highlight the tab (top left) to select the modulatory variables you should get the plot shown in figure 16.4.

16.3.3 Family level inference

The results so far have made no use of SPM's family inference procedure. Or rather, they have, but have assumed that all models belong to the same family.

Open the `ffx_all_models.mat` batch file again, highlight Family inference and select Load family. Highlight Load family and select the `pf_family.mat` file contained in the Zip archive. This comprises two families (i) those with a forward connection from P to F ('PF'), and (ii) those without it ('No PF'). Set the BMA option to Do not Compute. Select a new directory you have created for this analysis (eg pf-family) and run the job. SPM will create the family level inference plot shown in figure 16.5. This gives a 90% posterior probability to models with the P to F connection.

We will now repeat the analysis but with RFX inference. You should see a result similar to that shown in figure 16.6.

16.3.4 Summary Statistics and Group Analyses

The group mean DCM parameters can be easily obtained from the MATLAB command window by loading the `BMS.mat` file and then typing: `BMS.DCM.ffx.bma.Ep`.

The subject specific mean DCM parameters can be obtained as follows: `BMS.DCM.ffx.bma.Eps(n)`, where n is the subject number. For random-effects change `ffx` to `rfx`.

If we are interested in the modulatory connection from region 1 to region 3 (that is modulated by the second input), then the mean value of this for Subject 10 is given by `BMS.DCM.ffx.bma.Eps(10).B(3,1,2)` (which should be 0.7475). The mean connection values for all subjects (12) can be gotten with the MATLAB syntax
`for i=1:12, b(i) = BMS.DCM.ffx.bma.Eps(i).B(3,1,2); end`.

These subject specific mean parameters can then act as summary statistics for a standard group analysis. For example to look for significant differences between eg. a control group and a patient group in a modulatory parameter one would implement a two-sample t-test on data from the appropriate entries in the `mean_bs` matrices. Similarly, if one has 3 groups one would use a 3-level ANOVA.

16.4 BMS.mat file

The BMS structure saved in BMS.mat file contains the following variables[1]:

BMS.DCM.ffx/rfx   (fixed-effects (FFX) / random-effects (RFX) analysis)

.data	path to model_space.mat file (see below).
.F_fname	path to file containing the log evidence matrix, F, (if this option is specified).
.F	matrix of log model evidences for all subjects and models, [nsub × nm].
.SF	vector of summed log evidences over subjects [1 × nm].
.model	results from model level inference (see below).
.family	results from family level inference (see below).
.bma	results from Bayesian model averaging (see below).

---

[1]nm = number of models; nfam = number of families; nsub = number of subjects; nsamp = number of samples; dima/b/c/d = dimensions of a/b/c/d DCM parameters; np = number of model parameters; nsess = number of sessions.
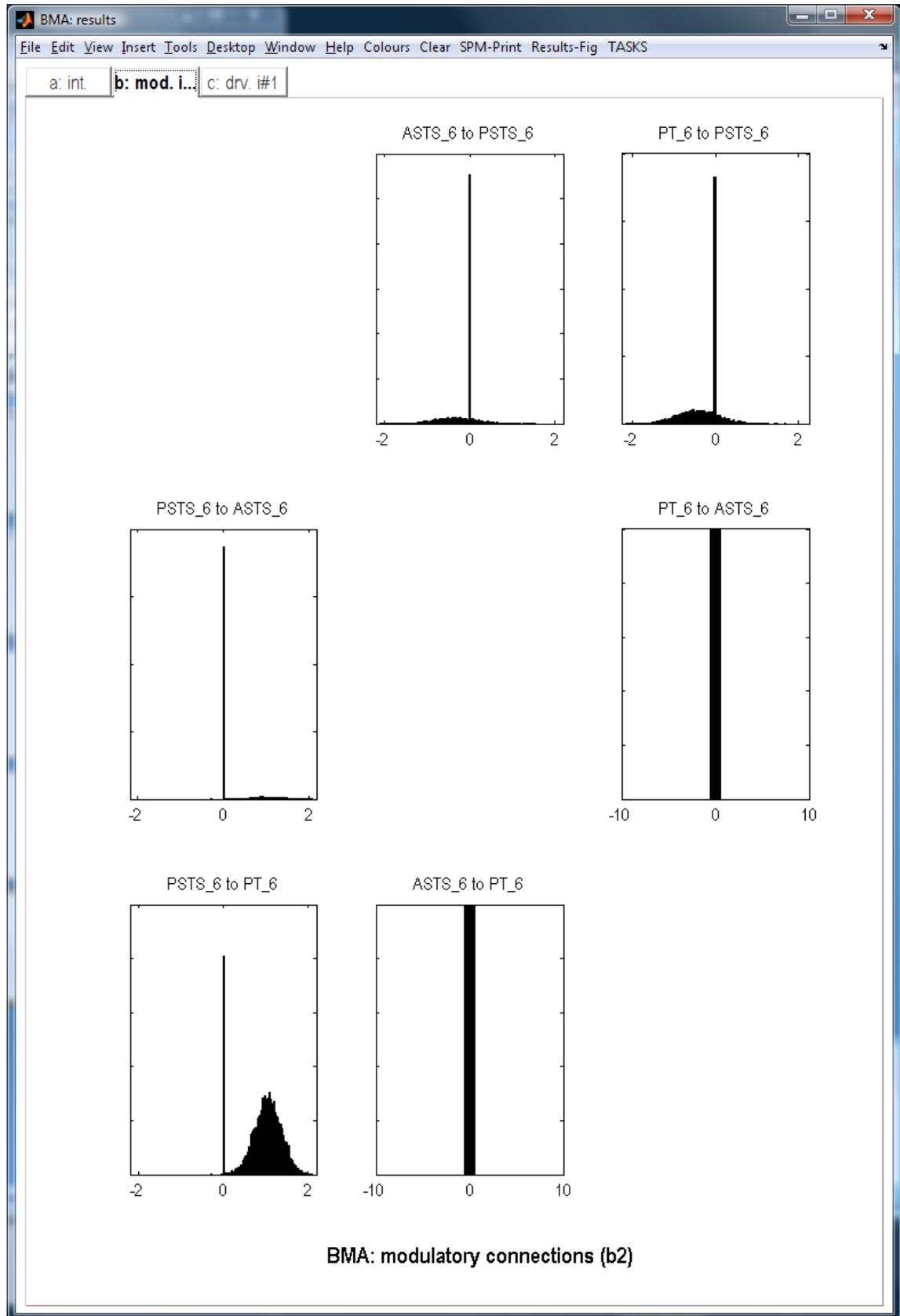
Figure 16.4: Bayesian model averaging over all 16 models
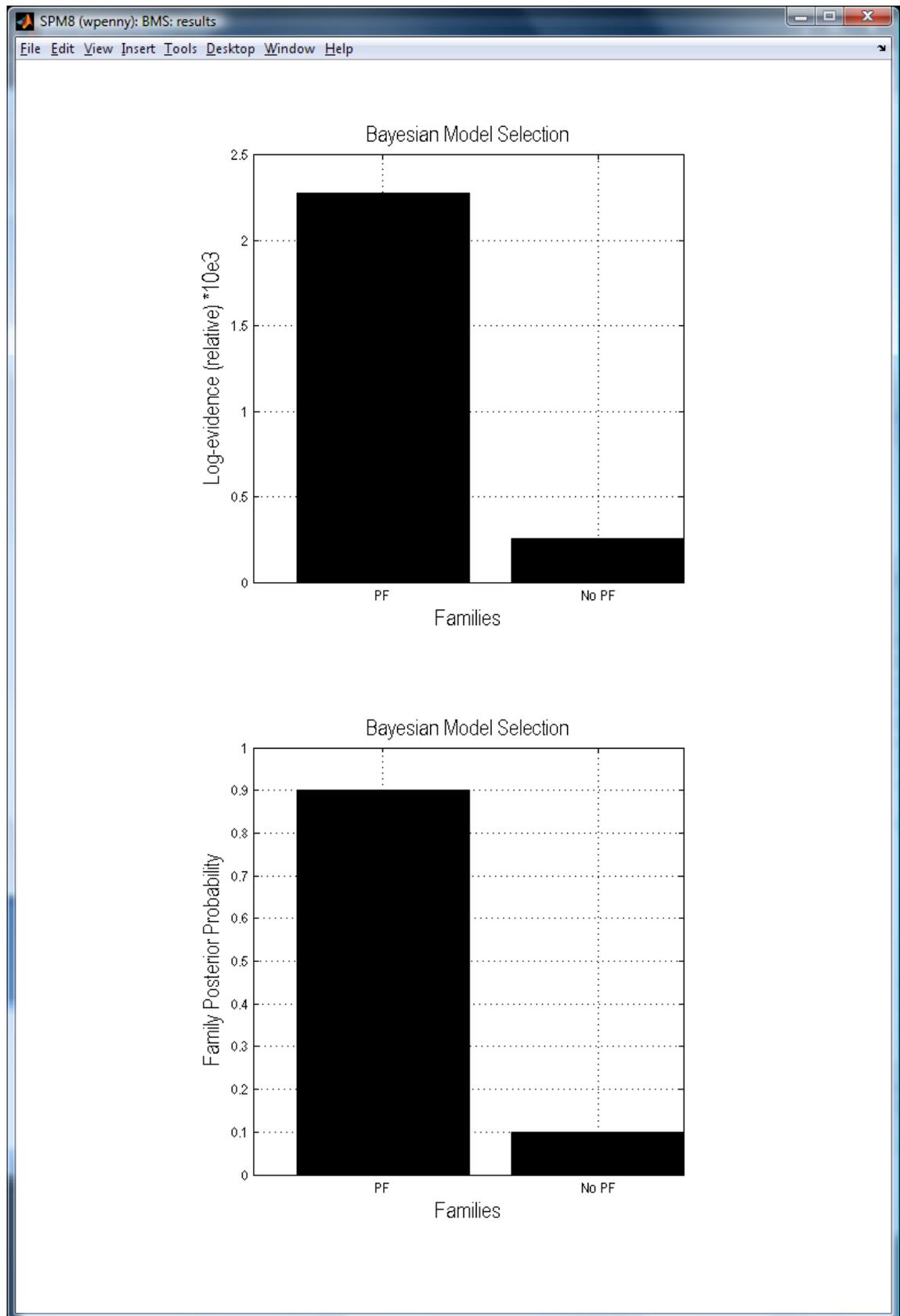
Figure 16.5: FFX family inference
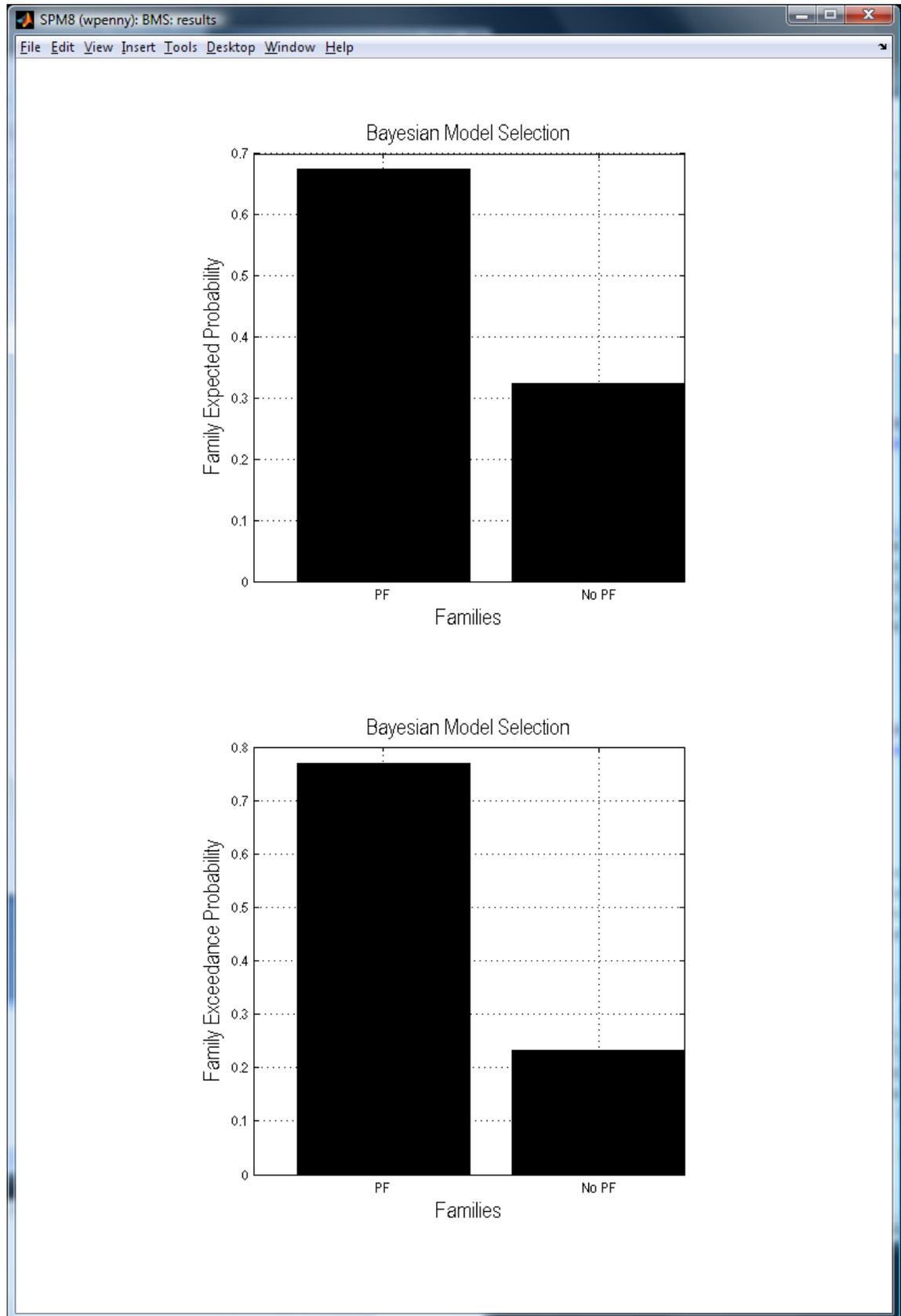
Figure 16.6: RFX family inference

16.4.1 Model level results

Fixed-effects:

model

.prior	model priors, $p(m)$, $[1 \times \text{nm}]$.
.subj_lme	log model evidence matrix, $[\text{nsub} \times \text{nm}]$.
.like	model likelihoods, $p(Y|m)$, $[1 \times \text{nm}]$.
.posts	model posterior probabilities, $p(m|Y)$, $[1 \times \text{nm}]$.

Random-effects (different from fixed-effects):

model

.alpha0	initial Dirichlet parameters (prior counts), α_0, $[1 \times \text{nm}]$.
.exp_r	model posterior means, $<r|Y>$, $[1 \times \text{nm}]$.
.xp	model exceedance probabilities, ψ_m $[1 \times \text{nm}]$.
.r_samp	samples from the model posterior density, $p(r|Y)$, $[\text{nsamp} \times \text{nm}]$.
.g_post	posterior model probabilities for subject n and model m, $p(m_n|Y)$, $[\text{nsub} \times \text{nm}]$.

16.4.2 Family level results

Fixed-effects:

family

.names	family names, ex: {'F1', 'F2', 'F3'}.
.partition	partition vector assigning each model to a family $[1 \times \text{nm}]$.
.infer	inference method ('ffx' or 'rfx').
.prior	family priors, $p(f_k)$, $[1 \times \text{nfam}]$.
.post	family posterior probabilities, $p(f_k|Y)$, $[1 \times \text{nfam}]$.
.like	family likelihoods, $p(Y|f_k)$, $[1 \times \text{nfam}]$.

Random-effects (different from fixed-effects):

family

.Nsamp	number of samples used in Gibbs sampling (default = 20000).
.prior	family type of priors ('F-unity', $\alpha_0 = 1$, for each family, is the default; other option, 'M-unity', $\alpha_0 = 1$, for each model) .
.alpha0	initial values of the Dirichlet parameters (prior counts), $\alpha_{prior}(m)$, $[1 \times \text{nfam}]$.
.s_samp	samples from family posterior density, $p(s|Y)$, $[\text{nsamp} \times \text{nfam}]$.
.exp_r	family posterior means, $<s_k|Y>$, $[1 \times \text{nfam}]$.
.xp	family exceedance probabilities, ψ_k, $[1 \times \text{nfam}]$.

16.4.3 Bayesian model averaging (BMA)

Fixed-effects:

bma

.nsamp	number of samples used to average parameters (default = 10000).	
.oddsr	posterior odds ratio, π_{OCC}, (number of models in Occams window, default = 0).	
.Nocc	number of models in Occam's window.	
.Mocc	index of models in Occam's window, $[1 \times nm]$.	
.indx	index of models in Occam's window (different for each subject in RFX), $[1 \times nm]$.	
.a	samples from posterior density over DCM.a parameters $[dima \times nsamp]$.	
.b	samples from posterior density over DCM.b parameters $[dimb \times nsamp]$.	
.c	samples from posterior density over DCM.c parameters $[dimc \times nsamp]$.	
.d	samples from posterior density over DCM.d parameters $[dimd \times nsamp]$.	
.mEp	mean DCM parameters $[1 \times 1$ struct$]$.	
.sEp	standard deviation of DCM parameters $[1 \times 1$ struct$]$.	
.mEps	mean DCM parameters per subject $[1 \times nsub$ struct$]$.	
.sEps	standard deviation DCM parameters per subject $[1 \times nsub$ struct$]$.	

Random-effects - same variables as in fixed-effects.

16.5   model_space.mat file

This structure is created automatically if it doesn't exist in the chosen directory and can be loaded for subsequent analyses as a faster option to reading the DCM.mat files. The model_space.mat file contains the following structure:

subj(nsub).sess(nsess).model(nm)

.fname	path to DCM.mat file.
.F	log-evidence (free energy).
.Ep	parameter estimates: conditional expectation, $[np \times 1]$.
.Cp	parameter estimates: conditional covariance, $[np \times np]$.

For a detailed description of all the variables and methods please see [61] and [65].

Chapter 17

MEG source localisation

17.1 Overview

In this section we will generate some simulated data to show how the inversion algorithms compare when the ground-truth is known.

17.2 Simulation

The code used to generate the simulated data is called `simmegdip_for_faces.m` and is located in the `man/example_scripts` directory. The simulation code takes the experimental setup: sensor locations, triggers, head model, etc, from the SPM M/EEG dataset[1]:

```
cdbespm8_SPM_CTF_MEG_example_faces1_3D.mat
cdbespm8_SPM_CTF_MEG_example_faces1_3D.dat
```

you created in chapter 19 and replaces the channel data with simulated data. It also forces the head model to be single-sphere. You can manipulate this code to change the number or locations of the simulated dipoles by changing

```
dipolepositions = [ 52, -29, 13; -52, -29, 13]; % in mni space
```

or modify the signal vector in anyway to change the time series being simulated. At the simplest level, the matrices `dipamp` and `dipfreq` define how each source behaves during a given condition.

```
% cond 1 cond 2
dipamp  = [ 1  0;...      % dip 1
            1  0].*1e-1; % dip 2
dipfreq = [10 10;...      % dip 1
           20 20];        % dip 2
```

That is, the two dipoles are currently set to be on (at 10 and 20Hz) during the faces condition and off during the scrambled condition.

You can run the script to create a new or modified dataset each time. If you run it with no changes, it will generate the dataset `simdata_aud1020Hz.mat`. This dataset is also available from the Multimodal face-evoked dataset webpage.

This file has dipoles at [52, -29, 13] and [-52, -29, 13] in MNI space. The dipoles are energized at 10Hz and 20Hz from 0.1 to 0.4 seconds (Figure 17.1). In each epoch the activation profile is identical, the channel data will be slightly different due to the white noise added.

---

[1]Multimodal face-evoked dataset: http://www.fil.ion.ucl.ac.uk/spm/data/mmfaces/
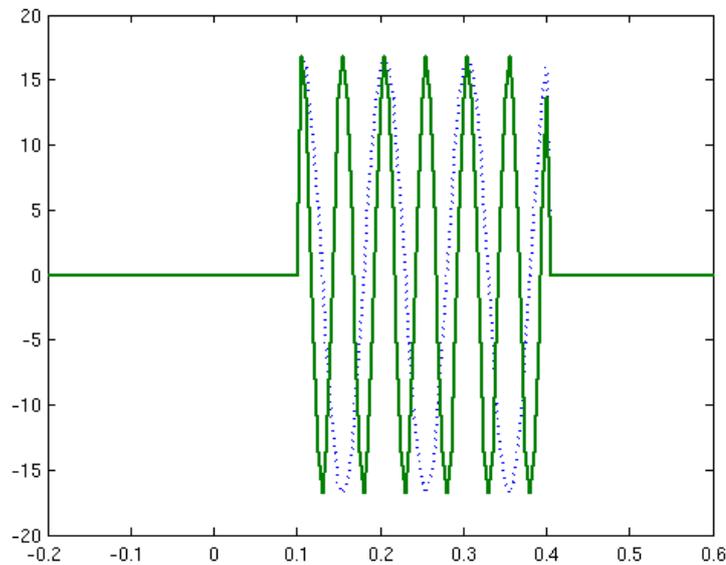
Figure 17.1: *Simulated source data*

17.3   Imaging solutions for evoked or induced responses

On the main menu Click `3D Source Reconstruction`. Press `Load`. Select the unaveraged file `simdata_aud1020Hz.mat`.

Moving left to right along the bottom panel you will notice that all of the buttons (MRI, Co-register, Forward Model) are active. This means that the preprocessing stages have already been carried out on these data (see multi-modal evoked responses chapter).

17.3.1   IID (minimum norm)

We will start off with the traditional minimum norm solution. This starts by assuming that all source elements contribute something to the measured data. The constraint is that the total power (in the sources) should be minimised. Press `Invert`. Under reconstruction method press `Imaging`. For `All conditions or trials` press `Yes`. For model press `Custom`. Model inversion IID. Under Time-window "0 600". For `PST Hanning` select `Yes`. For High-pass (Hz) select 1 for Low-pass (Hz) select 48. For `Source priors`, select No. Under `Restrict solutions` select No.

It appears (17.2) that the minimum norm solution has reconstructed only one of the sources. Note, however, the location of the dotted line in the upper time-series window. The source amplitude reconstruction is based on the location of this line (and conversely the time series shown taken from the maximum image voxel). In the `ms or mm` box fill in "205" then press `mip`. You should now see that the minimum norm solution is indeed bilateral, although rather diffuse (Figure 17.3). Note the log-evidence 4620718 (in the most recent version of SPM the absolute value of the log evidence may have changed, but it is this value relative to those following which is important).

The grey line in the time-series plot shows the amplitude of this voxel in the other ('scrambled') condition. To toggle between the conditions being viewed you can press the `condition 1/2` button. Then press the `mip` button to update the display. Often we will interested in the difference between conditions over a specific time-frequency window, rather than at a single latency. Press the `Window` button. At the `Time window (ms)` prompt type "0 600". For the `Frequency [band] of interest (Hz)` type "1 40". For `Power` select `induced` (although for these data `evoked` will work just as well). You will now see a glass brain showing a power plot for the current condition. You can convert this image and the corresponding image from the other condition into a volume by pressing `Image` (Figure 17.4).
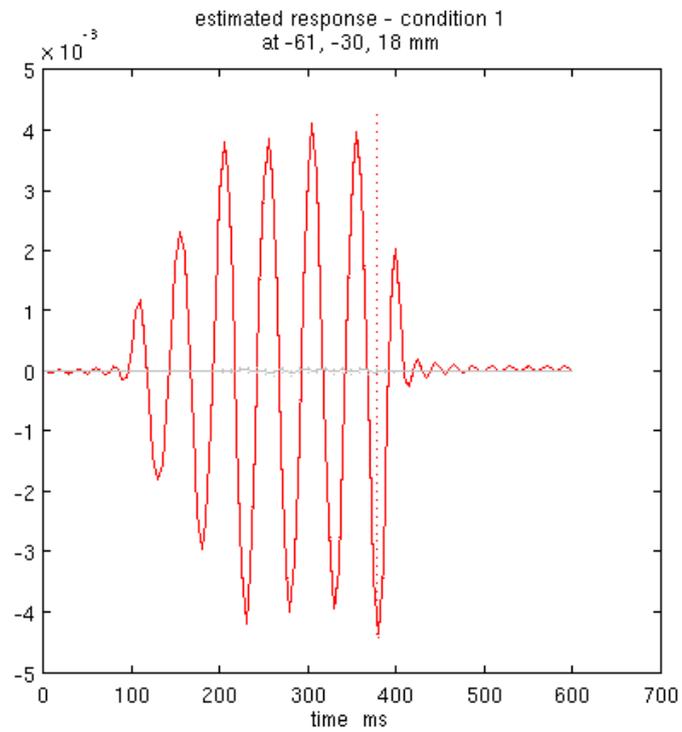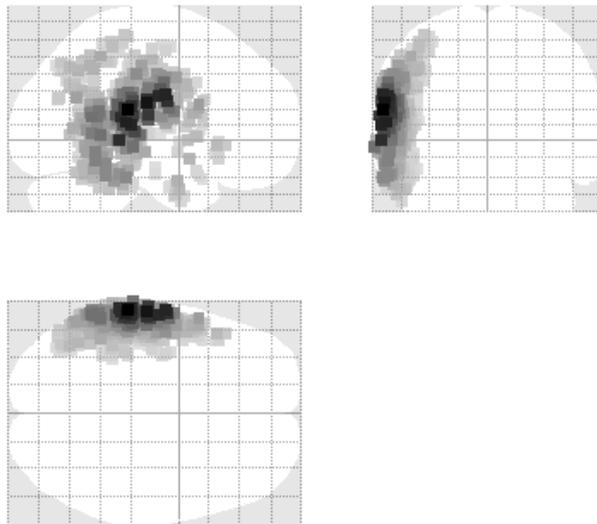
Figure 17.2: *IID imaging source reconstruction*
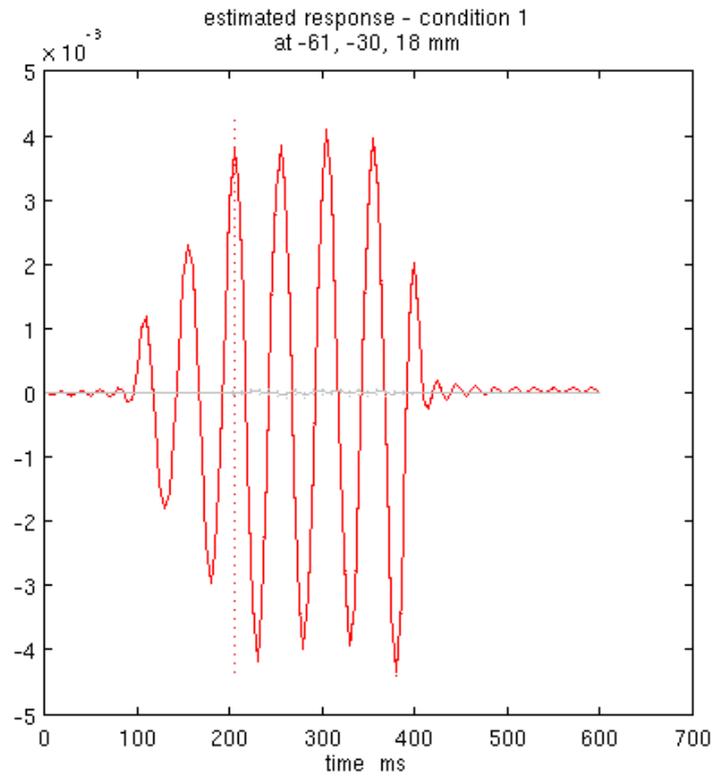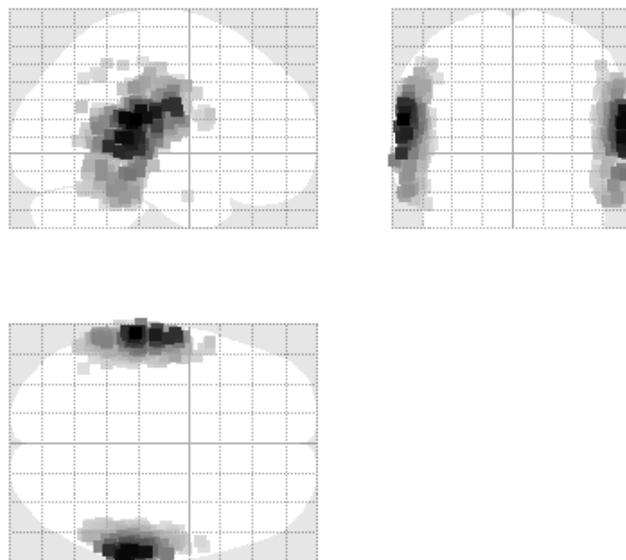
Figure 17.3: *IID imaging source reconstruction at 205ms*
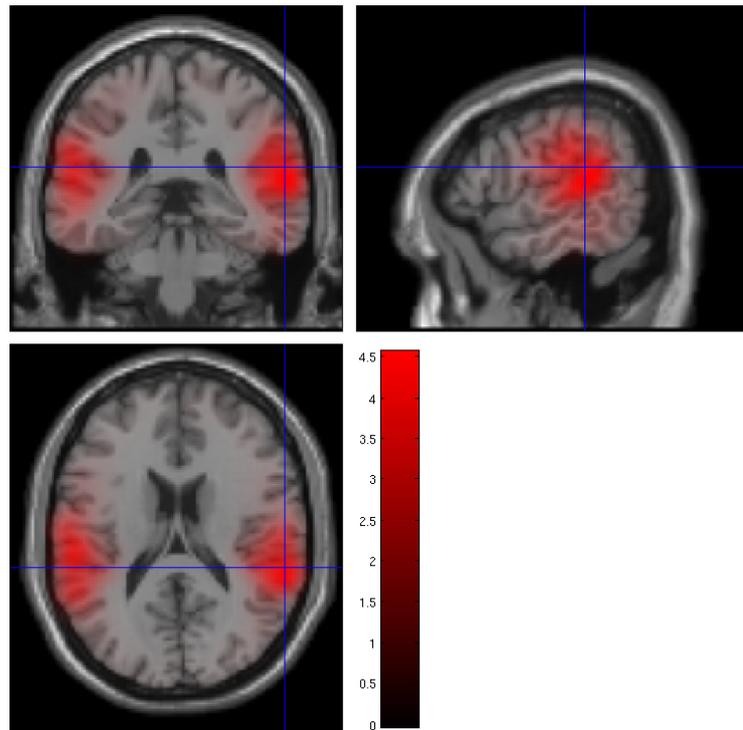
Figure 17.4: *Exported functional image from IID source reconstruction overlayed on the template structural image.*

17.3.2 Smooth priors (COH)

The COH option allows the mixture of two possible source covariance matrices: the minimum norm prior above and a much smoother source covariance matrix in which adjacent sources are correlated (over the scale of a few mm). Press `Invert`. Under reconstruction method press `Imaging`. For `All conditions or trials` press `Yes`. For model press `Custom`. Model inversion `COH`. Under Time-window "0 600". For `PST Hanning` select `Yes`. For High-pass (Hz) select `1` for Low-pass (Hz) select `48`. For `Source priors`, select `No`. Under `Restrict solutions` select `No`. You will see a plot similar to Figure 17.5 appear. The lower panel shows the glass brain in which bilateral sources are apparent. The upper panel shows the time-series of the source with the largest amplitude. In this case the peak activation is identified at location 59,-32 13. The 10Hz time-course (associated with this source) is also clearly visible in the top panel. Log evidence is 5272712 (again this number may be different in your spm version).

17.3.3 MSP

In contrast to IID or COH, the greedy search routine used in MSP builds up successive combinations of source configurations until the model evidence can no longer be improved. Press `Invert`. Under reconstruction method press `Imaging`. For `All conditions or trials` press `Yes`. For model press `Custom`. Model inversion `MSP`. Under Time-window "0-600". For `PST Hanning` select `Yes`. For High-pass (Hz) select `1` for Low-pass (Hz) select "48". For `Source priors`, select `No`. Under `Restrict solutions` select `No`. Note again the more focal sources (Figure 17.6) as compared to the minimum norm solution; although the time-series estimation seems to have suffered. Note the evidence 5464023.

We could now make a volume to compare with the minimum norm solution (Figure 17.4) [2]. Press the `Window` button. At the `Time window (ms)` prompt type "0 600". For the `Frequency`

---

[2]Note that you need to copy the previously generated image files to a different directory to prevent them from being overwritten
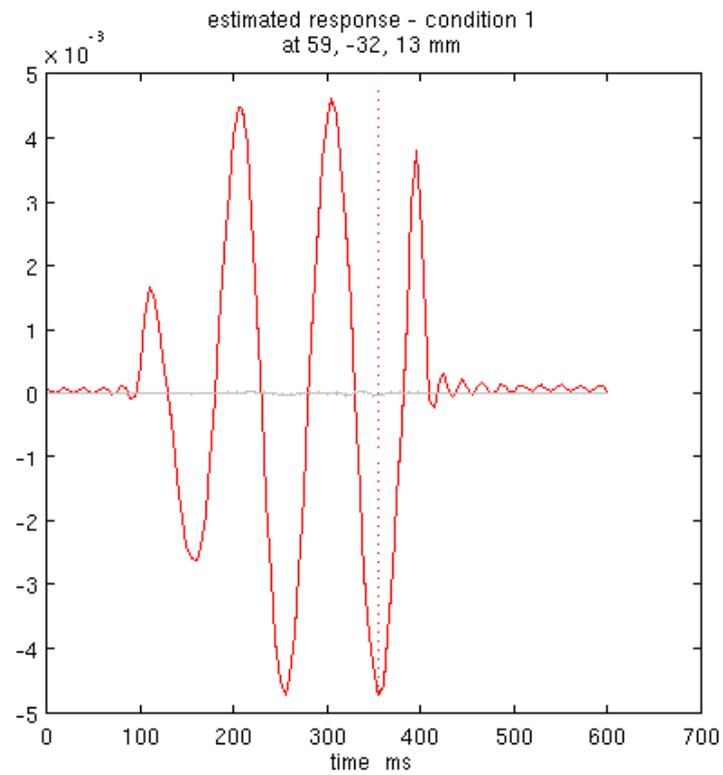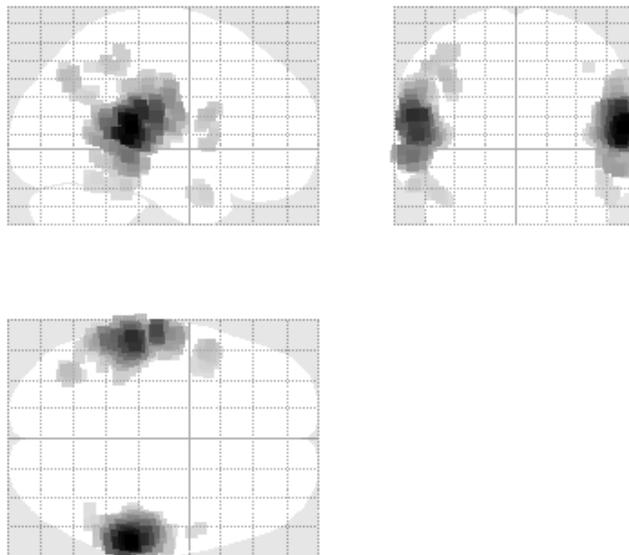
Figure 17.5: *COH imaging source reconstruction.*
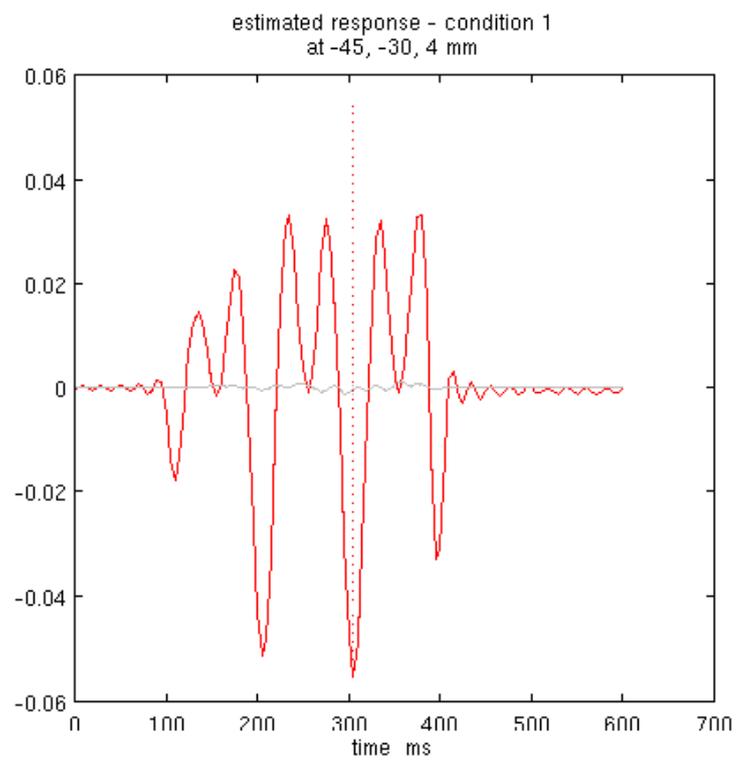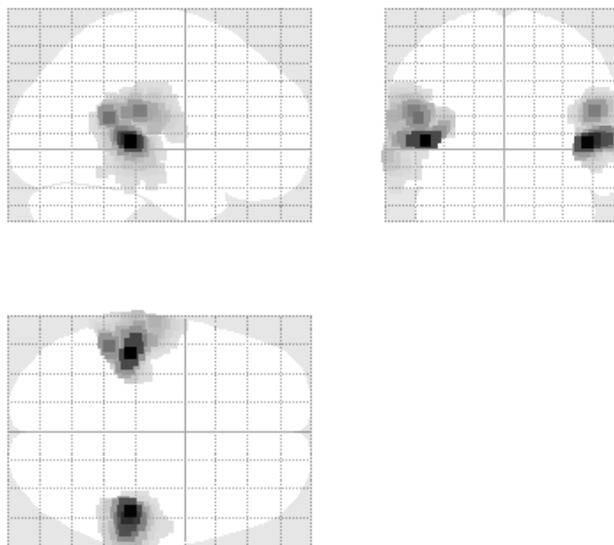
Figure 17.6: *MSP imaging source reconstruction.*
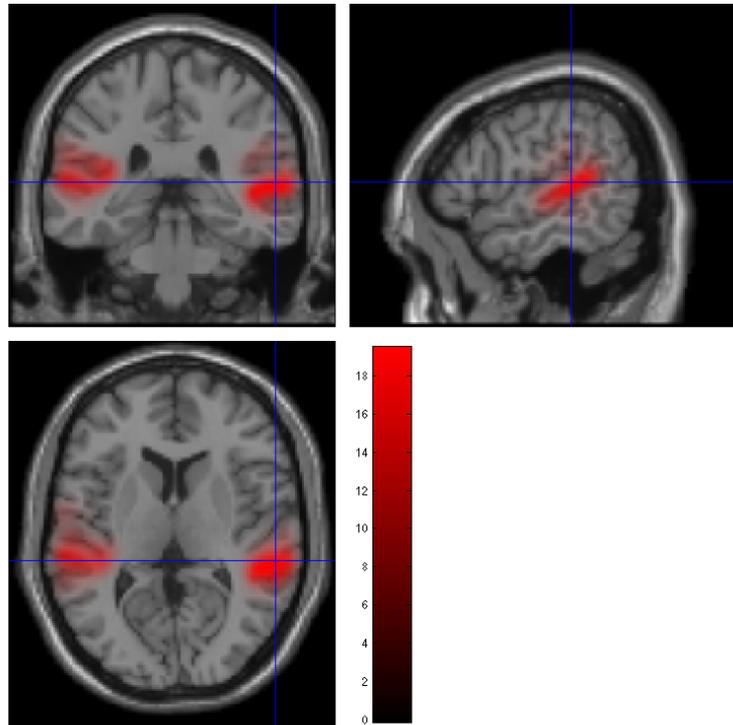
Figure 17.7: *Exported functional image from MSP source reconstruction overlayed on the template structural image.*

[band] of interest (Hz) type "1 40". For Power select induced (although for these data evoked will work just as well). You will now see a glass brain showing a difference in power plot between the two conditions. You can convert this into a volume by pressing Image (Figure 17.7).

17.3.4   Standard

The standard button is simply the use of the MSP greedy search using the whole time window available in a band of 2-128Hz. following default options: MSP, all time, hanning windowed, filter 2-128Hz.

17.3.5   Model comparison

As we tested all the above models on the same data we can make a model comparison. Figure 17.8 shows that the MSP model improves upon IID and COH solutions.
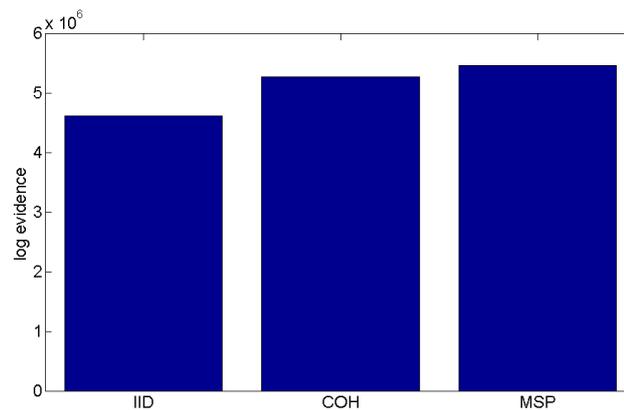
Figure 17.8: *Bayesian model evidence for different prior specifications.*

17.4 Beamformer analysis

In this final stage we will look for differences between the conditions using the linear constrained minimum variance beamformer. The main assumption in beamformer imaging is that there is no covariance between the underlying sources. That is, the prior source covariance matrix is a diagonal, the elements of which can be directly estimated from the sample covariance (Hillebrand et al 2005). The current beamformer implementations differ from the above in that they test points throughout the source space (i.e. not just on the cortical surface) and do not provide model evidence. At present therefore the beamformer routines sit apart from the other imaging reconstruction methods, but rely on the same coregistration and forward model selection.

17.4.1 LCMV beamformer

On the main SPM menu select the Toolbox pull-down menu. Under `Beamforming` select `Volumetric LCMV beamformer`. Select file `sim_data_aud1020Hz.mat`. The first menu will ask you to select the active condition. Select 'faces' and click `OK`. You will then be asked at what time the condition 'faces' begins. At `Offset (ms) from faces`, type 0. At the duration prompt type 600. i.e. we are using the same 0-600ms window as before. At the baseline condition prompt select 'scrambled'. `Offset ms from scrambled` should be set to 0ms also. We will be comparing the same time window 0-600ms across epochs labelled either scrambled, or faces. At the number of frequency bands prompt type "2". The more frequency bands one computes at this stage the smaller the overhead in time. For the first band type "15 25", for the second try "1 48". For `gridstep` type "10" mm (number of voxels hence speed decreases by a factor of 8 each time this is halved). Leave regularization at 0. Select `yes` for preview results. As you selected preview you will see an image similar to Figure 17.9, in which we restricted our analyses to the frequency band around one source. This is a t-statistic image showing the difference between the two conditions. You can navigate through the image using the mouse and need to press any key to continue processing. The second figure produced Figure 17.10 should be comparable with the previous time-window analyses on the 1-48Hz band.

Now to compare the beamformer images with some of our other reconstructions:

You will find the volumetric images within the sub-directory `''tstatBf_images`. You can use the `''checkReg` button to compare these images with other volumes (produced above using the `Window''/``Image` options. In Figure 17.11 the MSP and beamformer 0-600ms, 1-48Hz contrasts are shown alongside an anatomical image from the spm/templates directory. If you would like to observe the case where the beamformer breaks down due to correlated sources you could change the frequency of the two dipoles used in the simulation to be identical (e.g. dipfreq=[20 20;20 20]) and look at how the beamformer images degrade, whereas the MSP schemes (COH,ARD etc) should be relatively unaffected.
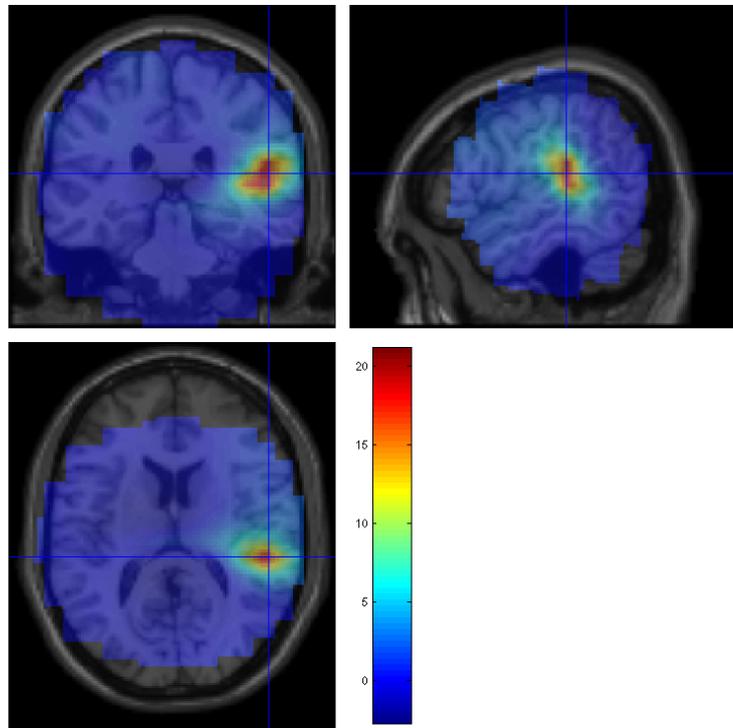
Figure 17.9: *LCMV beamformer results for 15-25Hz.*
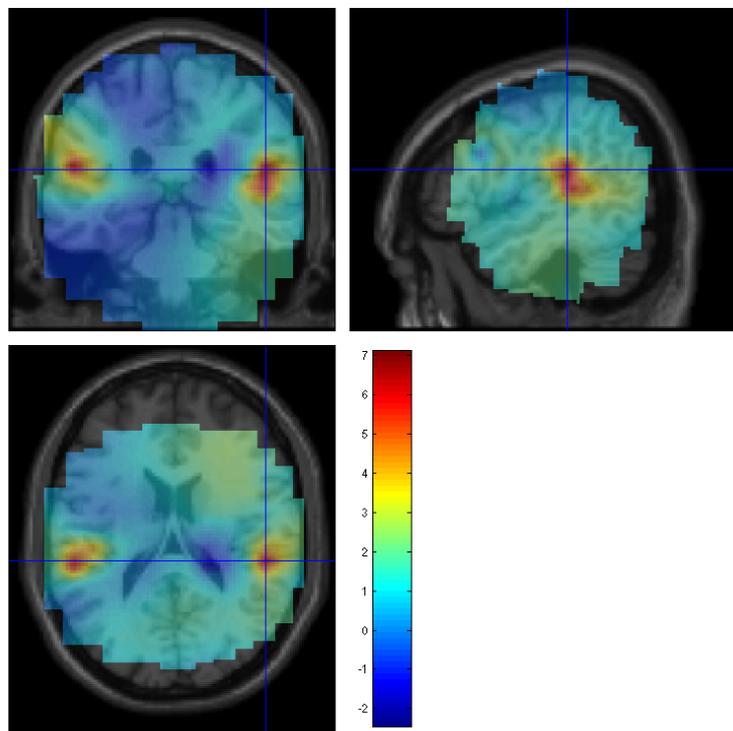
Figure 17.10: *LCMV beamformer results for 1-48Hz.*
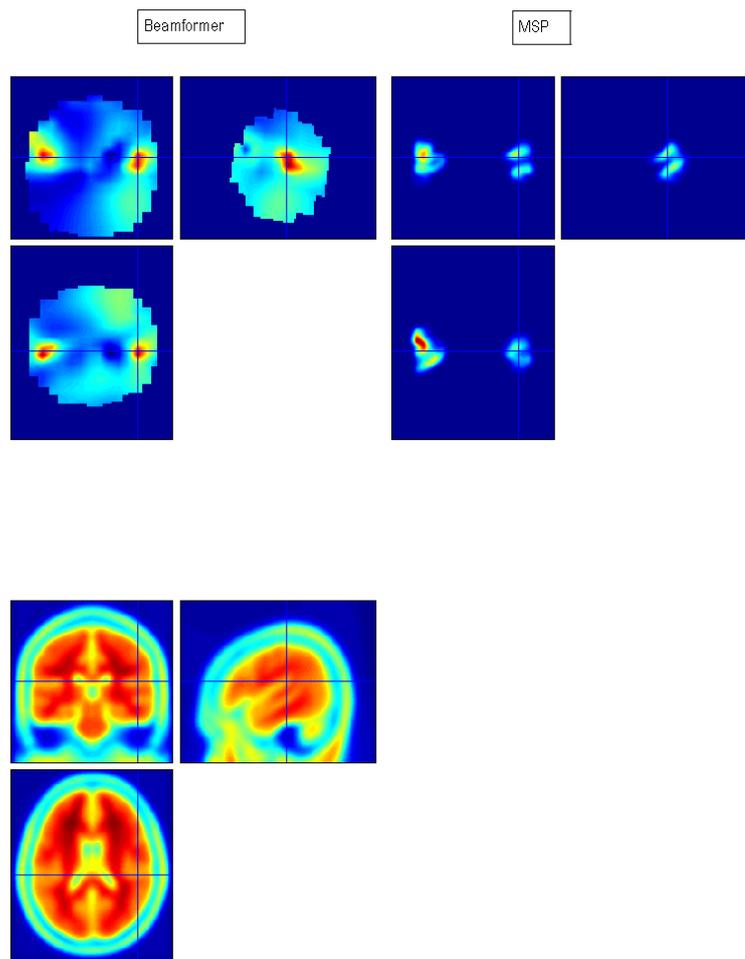
Figure 17.11: *Comparison of MSP and LCMV beamformer results.*
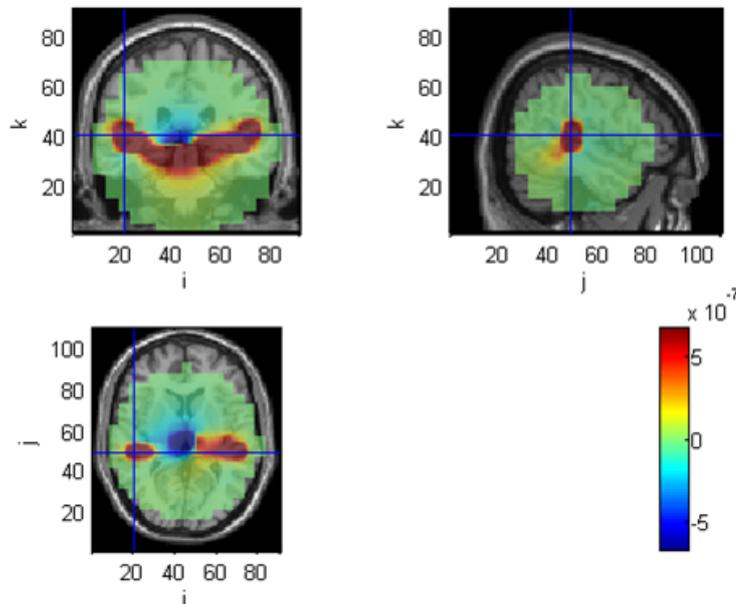
Figure 17.12: *DICS beamformer results.*

17.4.2 DICS beamformer

In `Beamforming` select `Fieldtrip DICS beamformer`. For a change we will simply test the faces condition across two different time windows. Under `Select conditions` select 'faces'. For `Frequency range` choose "1 48". For `Number of time windows` enter "2". For the first time window enter "-0.2 0" (i.e. baseline). For the second enter "0.1 0.3". Change the contrast vector to "-1 1" i.e. it will subtract the first condition (baseline) from the second. For `regularization'` select "0", for `preview results` select `yes`. You will see something like Figure 17.12. Note that this is not a statistical image, but simply the difference of the power in the pre- and post-stimulus faces condition. As source estimates tend to be large near the centre of the head (as the lead fields are small) these large differences remain in the final image. If you do proper statistical test over trials the activations in the centre of the head should not appear because of their high variability.

17.5 Dipole fitting to the average

Up until this point the analysis we have used could have been applied to either induced or evoked changes in electrical activity. The only difference being that it would not have made much sense to look at the MSPs for specific time-instants in the induced case and we would have proceeded directly to look for changes in a time-frequency window. To examine the dipole fit routine we will however concentrate on the averaged data file which will contain only evoked changes.

17.5.1 Load/preview the data

In the main menu click on the drop-down `Display` menu. Select `M/EEG`. For the dipole fitting we are going to use averaged MEG data, this is prefixed with an "m" in SPM. You can generate this file by averaging the epoched file that we have used until now. Select the file:

`msimdata_aud1020Hz.mat`

You will now see the simulated data. You will be looking at the "Trial 1 (..) Faces" trial, but if you switch (top-right menu) to "Trial 2 (..) scrambled" you will see that our simulated
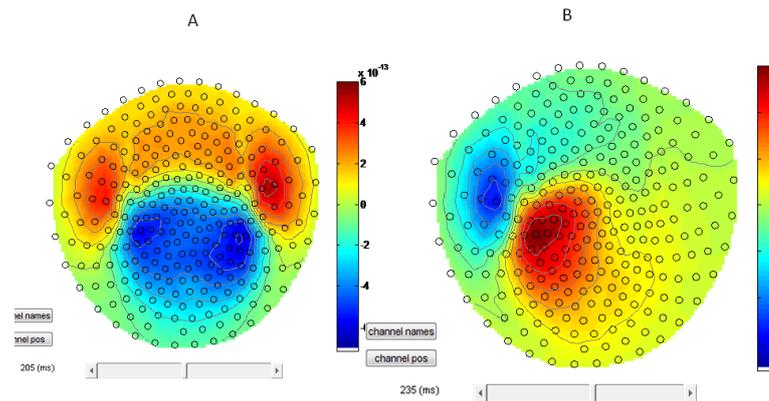
Figure 17.13: *Scalp maps for averaged simulated data.*

sources are not active during this condition. If you now click on the field map button you will see a cursor on the time-series plot and a field map for this time instant as in figure 3. Use the slider at the bottom of the field map to vary the time instant plotted. At 205ms (Figure 17.13A) you will notice that there are clearly two dipolar patterns, whereas at 235ms (Figure 17.13B) there is only one. We will now move on to explore Bayesian dipole fitting to these two time instants.

17.5.2   Inversion

In the main menu window, select `3D Source Reconstruction`. Click `Load` and select the averaged simulated dataset above. Proceed by pressing the `Invert` button. Select the `VB-ECD` button.

Fitting a single dipole with no priors

At the `time_bin or average_win` prompt enter "235". For `Trial type number` choose "1" (we want to model the faces data). At the `Add dipoles to model` click `Single`. For `location prior` click `Non-info`. For `Moment prior` click `Non-info`. At the `Add dipoles to 1 or stop?` prompt click `stop`. At the `Data SNR (amp)` leave as default 5. Leave the default number of iterations at "10". You will see the 10 successive fits of the same data using a random starting location and moment. At each fit maps of the predicted and simulated data along with free-energy values and percent variance explained are shown. The final plot will be similar to Figure 17.14 where the model (i.e. dipole) which maximised the evidence (the best iteration is shown with a red dot) is displayed. Note down the model evidence (in this case -4.3, but the absolute value in your implementation may be different). The Bayesian dipole fit algorithm will be most useful when one has some prior knowledge of the sources (such as location, orientation or symmetry). Typical dipole fit algorithms fit 3 location parameters per dipole and then estimate the moment through a pseudo-inverse. The VB-ECD algorithm however fits 6 parameters per dipole as the moments are also allowed prior values. That is, if you have no prior knowledge then the Bayesian method will be generally less robust than such fitting methods (as more parameters are being fit). However it is when prior knowledge is supplied that the Bayesian methods become optimal.

Fitting a single dipole with reasonable and unreasonable priors

We will now provide some prior knowledge to the dipole fit perhaps led by the literature or a particular hypothesis. In this case we know the answer, but let us specify a location a couple of cm from where we know the source to be (at -52,-29,13mm) and try the fit again. At the `time_bin or average_win` prompt enter "235". For `Trial type number` choose "1" (we want to model the faces data). At the `Add dipoles to model` click `Single`. For `location prior` click `Informative`. For the location enter "-62 -20 10". For prior location variance leave at "100 100 100" mm^2. This means that we are not sure about the source location to better than 10mm in each dimension. For `Moment prior` click `Non-info`. At the `Add dipoles to 1 or`
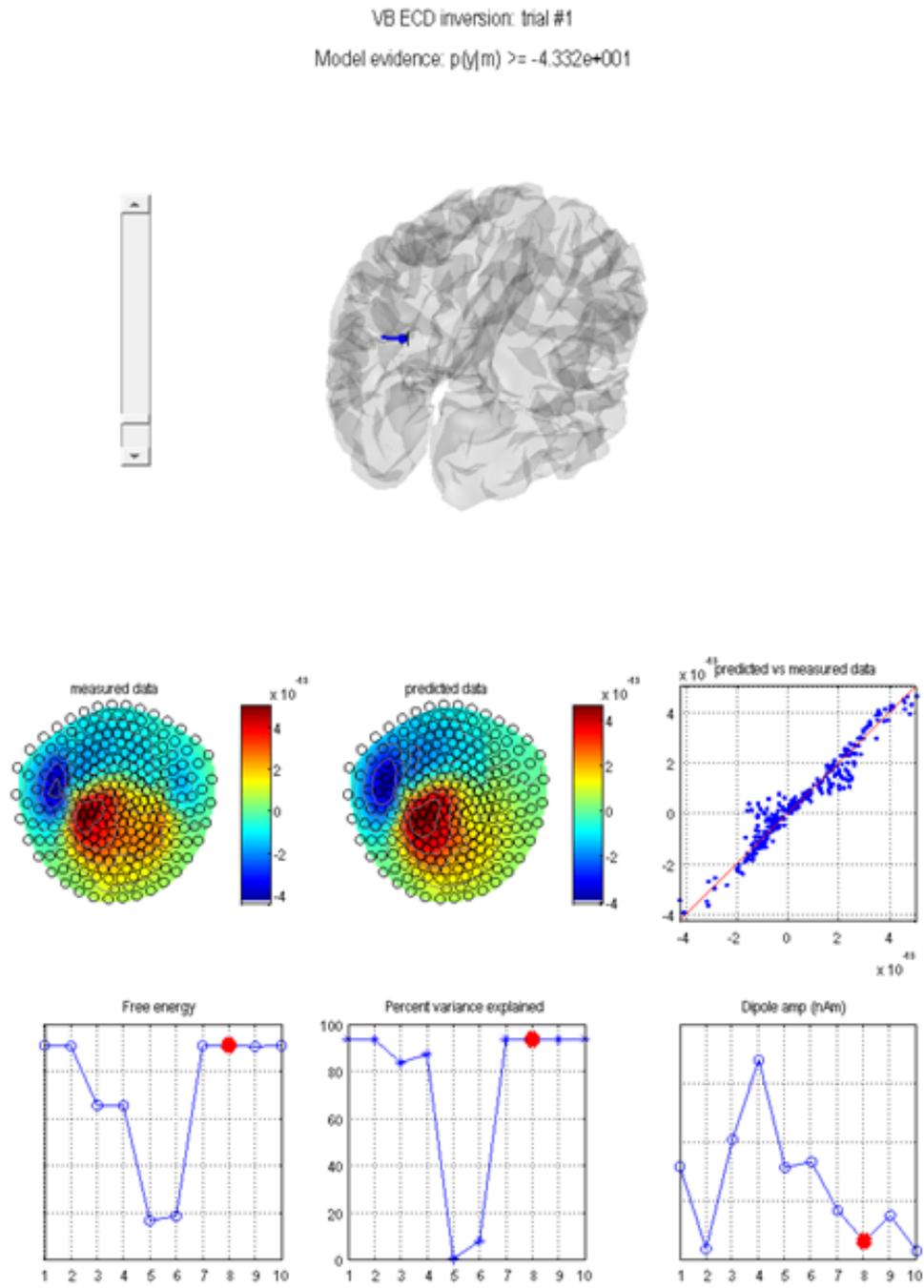
Figure 17.14: *Results of fitting a single dipole with noninformative priors.*

`stop?` prompt click `stop`. At the `Data SNR (amp)` leave as default `5`. Leave the default number of iterations at "10". Again you will get a final fit location and model evidence (-3.8), which should have improved (be more positive) on the evidence above. Now go through exactly the same procedure as above but for the prior location enter "+62 -20 10", i.e. on the wrong side of the head. You will note that the algorithm finds the correct location but the evidence for this model (with the incorrect prior) is much lower (-8.4). You could also try to fit a symmetric pair to this latency dataset (see below), model evidence -150.

Fitting more dipoles

We will start by examining the time instant at which we can clearly see a two-dipolar field pattern. At the `time_bin or average_win` prompt enter "205". For `Trial type number` choose "1". At the `Add dipoles to model` click `Single`. For `location prior` click `Informative`. For the location enter "62 -20 10". For prior location variance enter "400 400 400" mm^2, that is, the prior standard deviation on the dipole location is 20mm in each direction. For `Moment prior` click `Non-info`. At the `Add dipoles to 1 or stop?` prompt click `Single`. For `location prior` click `Informative`. For the location enter "-62 -20 10". For prior location variance enter "400 400 400" mm^2. At the `Add dipoles to 1 or stop?` prompt click `stop`. At the `Data SNR (amp)` leave as default `5`. Leave the default number of iterations at "10". Note down the final model evidence (570).

Alternatively we can exploit the fact that we have prior knowledge that the dipoles will be approximately left-right symmetric in location and orientation. At the `time_bin or average_win` prompt enter "205". For `Trial type number` choose "1". At the `Add dipoles to model` click `Symmetric Pair`. For `location prior` click `Informative`. For the location enter 62 -20 10. For prior location variance enter "400 400 400" mm^2. For `Moment prior` click `Non-info`. At the `Add dipoles to 2 or stop?` prompt click `stop`. At the `Data SNR (amp)` leave as default `5`. Leave the default number of iterations at "10". Note that the final locations are approximately correct, but importantly the model evidence (-9.6) is much lower than previously. Given this information one would accept the two distinct dipole model over the symmetric pair.

Finally we can take our best model so far, and try adding in an extra source. At the `time_bin or average_win` prompt enter "205". For `Trial type number` choose "1". At the `Add dipoles to model` click `Single`. For `location prior` click `Informative`. For the location enter "62 -20 10". For prior location variance enter "400 400 400" mm^2, that is, the prior standard deviation on the dipole location is 20mm in each direction. For `Moment prior` click `Non-info`. At the `Add dipoles to 1 or stop?` prompt click `Single`. For `location prior` click `Informative`. For the location enter "-62 -20 10". For prior location variance enter "400 400 400" mm^2. At the `Add dipoles to 2 or stop?` prompt click `Single`. We will add an extra dipole that can be anywhere in the head, so for `Location Prior` click `Non-info`. For `Moment Prior` click `non-info`. At the `Add dipoles to 3 or stop?` prompt click `Stop`. At the `Data SNR (amp)` leave as default `5`. Leave the default number of iterations at "10". Note down the final model evidence (550). That is, the evidence has now begun to decrease once more suggesting that the two distinct dipole model is the best for these data.

Chapter 18

EEG Mismatch negativity data

This chapter describes the analysis of a 128-channel single subject EEG data set acquired from a study of mismatch negativity in the auditory system [26]. We thank Marta Garrido for providing us with these data. The experiment comprised an auditory oddball paradigm in which subjects heard standard (500Hz) and deviant (550Hz) tones, occuring 80% (480 trials) and 20% (120 trials) of the time, respectively, in a pseudo-random sequence subject to the constraint that two deviant tones did not occur together.

EEG data were recorded with a Biosemi[1] system at 128 scalp electrodes and a sampling rate of 512Hz. Vertical and horizontal eye movements were monitored using EOG electrodes. See [26] for full details of experimental stimuli and recording. To proceed with the data analysis, first download the data set from the SPM website[2]. The data comprises a file called `subject1.bdf` whose size is roughly 200MB. We will refer to the directory in which you have placed it as `DATA_DIR`. This chapter takes you through different stages of analysis:

- Preprocessing

- Sensor space analysis

- Source reconstruction

- Dynamic Causal Modelling

18.1 Preprocessing

There is also an example MATLAB script under `man\example_scripts\history_subject1.m` in the SPM distribution which repeats the preprocessing route we take here.

18.1.1 Convert

At the MATLAB prompt type `spm eeg`, press the CONVERT button and select the `subject1.bdf` file. At the prompt "Define settings ?" select "just read".

SPM will now read the original Biosemi format file and create an SPM compatible data file, called `spm8_subject1.mat` and `spm8_subject1.dat` in the directory containing the original data file (`DATA_DIR`).

18.1.2 Montage

In this step, we will identify the VEOG and HEOG channels, and also remove several channels that don't carry EEG data and are of no importance to the following. In this case we apply montage as the first processing step to set all the channel types correctly for subsequent processing. This is especially important for the EOG channels, which are derived from channels that would not normally be filtered because SPM does not recognize them as containing M/EEG data. We

---

[1]BioSemi: http://www.biosemi.com/
[2]EEG MMN dataset: http://www.fil.ion.ucl.ac.uk/spm/data/eeg_mmn/

generally recommend to remove all data channels that are no longer needed because it will reduce the total file size. We will also convert the data to average reference montage by subtracting from each channel the mean of all EEG channels. To do so, we use the MONTAGE tool in SPM, which is a general approach for pre-multiplying the data matrix (channels × time) by another matrix that linearly weights all channel data. This provides a very general method for data transformation in M/EEG analysis.

The appropriate montage-matrix can be derived as follows. In our case, we would like to only keep EEG channels 1 to 128 and subtract from each channel the average of all EEG channels. In addition, there were three EOG channels (129, 130, 131), where the HEOG is computed as the difference between channels 131 and 130, and the VEOG by the difference between channels 130 and 129. This matrix can be specified in SPM by either using a graphical interface, or by supplying the matrix saved in a file. We will do the latter. The script to generate this file can be found in the `example_scripts` folder: `montage_subject1.m`. Copy this script into `DATA_DIR` and run it. This will generate a file named `MONT_EXP.mat`.

You now call the montage function by choosing MONTAGE in the "Other" drop-down menu and:

- Select the M/EEG-file `spm8_subject1.mat`

- 'How to specify the montage ?' Answer "file".

- Then select the generated `MONT_EXP.mat` file

- "Keep the other channels?" : "No"

This will remove the uninteresting channels from the data. The progress bar appears and SPM will generate two new files `Mspm8_subject1.mat` and `Mspm8_subject1.dat`.

This step will also assign default locations to the sensors, as this information is not contained in the original Biosemi `*.bdf` file. It is usually the responsibility of the user to link the data to sensors which are located in a coordinate system. In our experience this is a critical step. SPM provide tools (PREPARE) for linking data and location information, leaving it the responsibility of the user to verify the success of this process. Chapter 4 describes in detail how you can use the PREPARE tool from the "Other" drop-down menu to use digitized sensor location data.

18.1.3   Filter

Filtering the data in time removes unwanted frequency bands from the data. Usually, for evoked response analysis, the low frequencies are kept, while the high frequencies are assumed to carry noise. Here, we will use a highpass filter to remove ultra-low frequencies close to DC, and a lowpass filter to remove high frequencies. We filter prior to downsampling because otherwise high-amplitude baseline shifts present in the data will generate filtering artefacts at the edges of the file.

- Click on FILTER and select the `Mspm8_subject1.mat` file.

- Select a "highpass" filter with cutoff of 0.5 (Hz).

The progress bar will appear and the resulting filtered data will be saved in files `fMspm8_subject1.mat` and `fMspm8_subject1.dat`.

- Click on FILTER and select the `fMspm8_subject1.mat` file.

- Select a "lowpass" filter with cutoff of 30 (Hz).

The progress bar will appear and the resulting filtered data will be saved in files `ffMspm8_subject1.mat` and `ffMspm8_subject1.dat`.

18.1.4 Downsample

Here, we will downsample the data in time. This is useful when the data were acquired like ours with a high sampling rate of 512 Hz. This is an unnecessarily high sampling rate for a simple evoked response analysis, and we will now decrease the sampling rate to 200 Hz, thereby reducing the file size by more than half. Select DOWNSAMPLE from the "Other" drop-down menu and select the `ffMspm8_subject1.mat` file. Choose a new sampling rate of 200 (Hz). The progress bar will appear and the resulting data will be saved to files `dffMspm8_subject1.mat` and `dffMspm8_subject1.dat`. This step requires the Signal Processing Toolbox.

18.1.5 Epoch

To epoch the data click on EPOCHING. Select the `dffMspm8_subject1.mat` file. Choose the peri-stimulus time window, first the start `-100`, then the end `400` ms. Choose 2 conditions. You can call the first condition "standard". A GUI pops up which gives you a complete list of all events in the EEG file. The standard condition had 480 trials, so select the type with value 1 and press OK. The second condition can be called "rare". The rare stimulus was given 120 times and has value 3 in the list. Select this trial type and press OK. Answer two times "no" to the questions "review individual trials", and "save trial definitions". The progress bar will appear and the epoched data will be saved to files `edffMspm8_subject1.mat` and `edffMspm8_subject1.dat`.

18.1.6 Artefacts

A number of different methods of artefact removal are implemented in SPM8. Here, we will demonstrate a simple thresholding method. However, before doing so, we will look at the data in the display:

- Choose "M/EEG" from the "Display" dropdown menu.

- Select the `edffMspm8_subject1.mat` file.

- Click on the "EEG" tab.

- Press the "scalp" radio button.

The time-series for the first trial will then appear in the topographical layout shown in Figure 18.1.

You will see that Channel 14, second-row from bottom, left hemisphere, contains (slightly) higher variability data than the others . Right-click on the channel; this tells you that this channel is "A14". You will also see as an entry in this menu "bad: 0". Select this entry, and click the left button. This will make the menu disappear, but the channel now has a grey background. You have marked this channel as bad. Click on "save" in the top-right corner. This channel will then be ignored in subsequent processing. In fact this channel probably doesn't need removing, but we do so for teaching purposes only.

Now, click on ARTEFACTS. A window of SPM8 batch interface will open. You might be already familiar with this interface from other SPM8 functions. It is also possible to use the batch interface to run the preprocessing steps that we have performed until now, but for artefact detection this is the only graphical interface. Click on "File name" and select the `edffMspm8_subject1.mat` file. Double click "How to look for artefacts" and a new branch will appear. It is possible to define several sets of channels to scan and several different methods for artefact detection. We will use simple thresholding applied to all channels. Click on "Detection algorithm" and select "Threshold channels" in the small window below. Double click on "Threshold" and enter 80 (in this case μV). The batch is now fully configured. Run it by pressing the green button at the top of the batch window.

This will detect trials in which the signal recorded at any of the channels exceeds 80 microvolts (relative to pre-stimulus baseline). These trials will be marked as artefacts. Most of these artefacts occur on the VEOG channel, and reflect blinks during the critical time window. The procedure will also detect channels in which there are a large number of artefacts (which may reflect problems specific to those electrodes, allowing them to be removed from subsequent analyses).

In this case, the Matlab window will show:
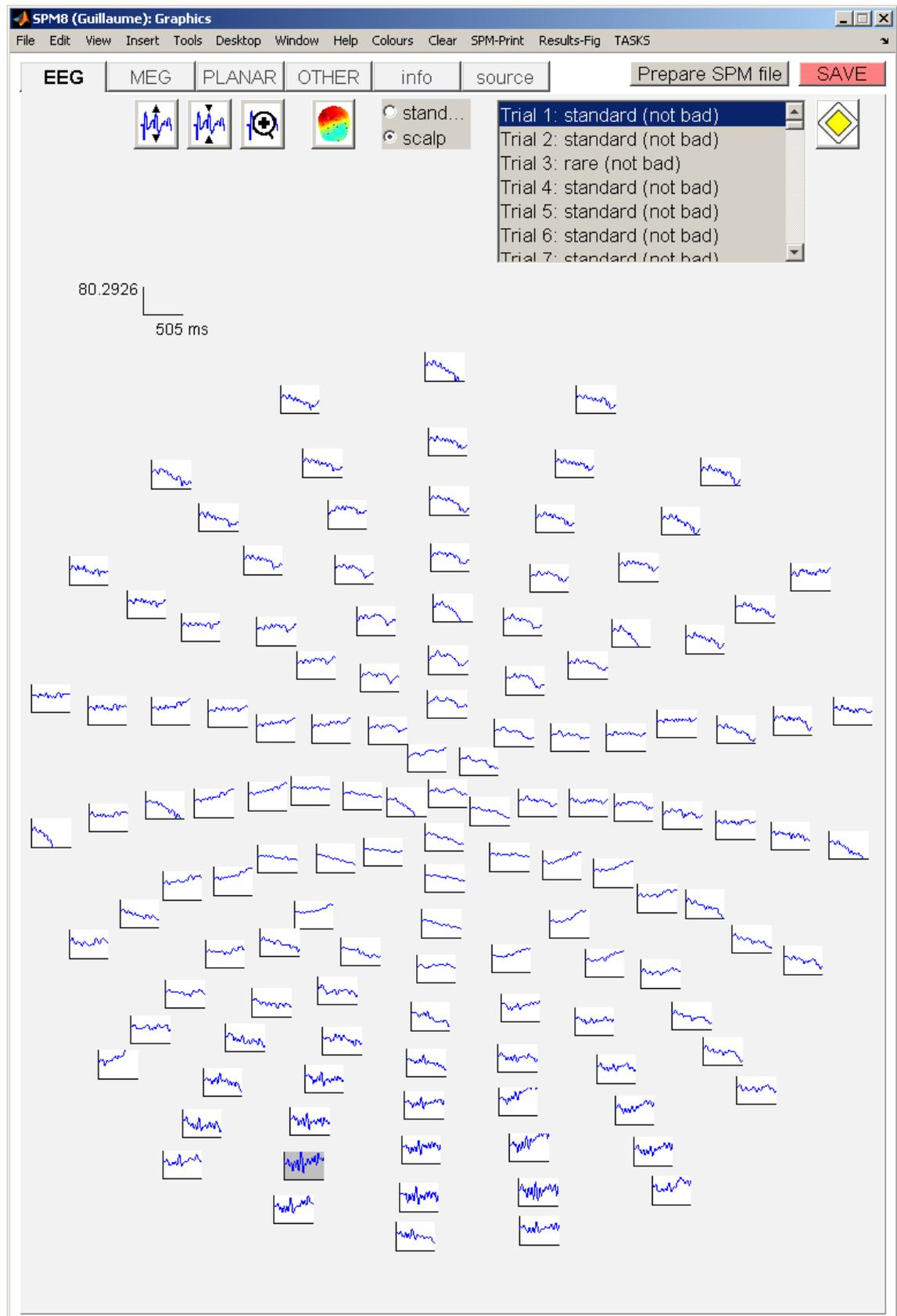
Figure 18.1: *Scalp topography of single trial MMN EEG data. Channel 14, second-row from bottom, left hemisphere contains (slightly) higher variability data than the others. This channel is to be marked as artefactual (ie. 'bad').*

```
1 bad channels: A14
77 rejected trials:  3    4    5    7    8    9   10   12   28   29   88  [...]
Done    'M/EEG Artefact detection'
Done
```

A new file will also be created, `aedffMspm8_subject1.mat`.

There are also interactive artefact removal routines available from `Toolbox` → `MEEG tools` → `Fieldtrip visual artifact rejection`.

18.1.7 Averaging

To produce an ERP click on AVERAGING and select the `aedffMspm8_subject1.mat` file. At this point you can perform either ordinary averaging or "robust averaging". Robust averaging makes it possible to supress artefacts automatically without rejecting trials or channels compltely, but just the contaminated parts. For robust averaging answer "yes" to 'Use robust averaging?". Answer "yes" to "Save weights", and "yes" to "Compute weights by condition" [3] and press "Enter" to accept the default "Offset of the weighting function". A new dataset will be generated `maedffMspm8_subject1` and automatically opened in the reviewing tool so that you can examine the ERP. There will also be an additional dataset named `WaedffMspm8_subject1` this dataset will contain instead of EEG data the weights used by robust averaging. This is useful to see what was suppressed and whether there might be some condition-specific bias that could affect the results.

The Graphics window will pop up and allow you to look at the averaged data. To look at the ERP, click on the EEG tab, and press the "scalp" radio button. Now hold the Shift button down on the keyboard whilst selecting trial 2 with the left mouse button in the upper right corner of the graphics window. This will overlay responses to standard and rare trials on the same figure axes.

Now press the "plus" icon at the top of this graphics window and select channel C23 (seventh central channel down from the top) with a left mouse click. This will plot the ERPs shown in Figure 18.2. This completes the preprocessing step.

18.1.8 History

The evoked response file (and every other SPM MEEG data file) contains a history-entry which stores all of the above preprocessing steps. You can take this history and produce a script that will re-run the same analysis which you entered using the GUI. See the "history" tab in the "info" section when displaying the data. Chapter 4 provides more details on this.

18.2 Sensor space analysis

A useful feature of SPM is the ability to use Random Field Theory to correct for multiple statistical comparisons across N-dimensional spaces. For example, a 2D space representing the scalp data can be constructed by flattening the sensor locations and interpolating between them to create an image of MxM pixels (when M is user-specified, eg M=32). This would allow one to identify locations where, for example, the ERP amplitude in two conditions at a given timepoint differed reliably across subjects, having corrected for the multiple t-tests performed across pixels. That correction uses Random Field Theory, which takes into account the spatial correlation across pixels (i.e, that the tests are not independent). Here, we will consider a 3D example, where the third dimension is time, and test across trials within this single subject. We first create a 3D image for each trial of the two types, with dimensions M×M×S, where S=101 is the number of samples (time points). We then take these images into an unpaired t-test across trials (in a 2nd-level model) to compare "standard" and "rare" events. We can then use classical SPM to identify locations in space and time in which a reliable difference occurs, correcting across the multiple comparisons entailed. This would be appropriate if, for example, we had no a priori

---

[3]In this case we do not want to pool both conditions together because the number of standard and rare trials are quite different.
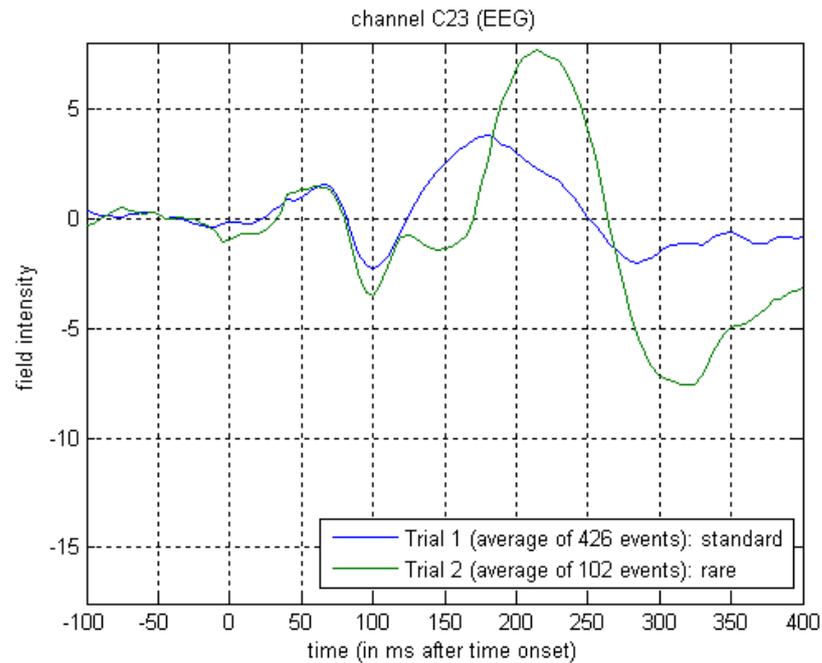
Figure 18.2: *ERPs at channel C23 (CZ) for standard and rare tones. The ERP curve for rare tones lies underneath that for standard tones between 100 and 180ms. This corresponds to the mismatch negativity signal in this subject.*

knowledge where or when the difference between standard and rare trials would emerge. The appropriate images are created as follows

- Select the "Convert to image" option from the "Other" pulldown menu.

- Select the `aedffMspm8_subject1.mat` file.

- For "output image dimensions" accept the default of 32 (leading to a 32x32 pixel space).

- For "interpolate' or "mask out" bad channels, select "interpolate".

SPM will take some time as it writes out a NIfTI image for each trial (except rejected trials), in a new directory called `aedffMspm8_subject1`, which will itself contain two subdirectories, one for each trialtype, called `type_rare` and `type_standard`. In each trialtype subdirectory there will be image and header files for each non-rejected trial of that type, e.g, trial0001.img/hdr. You can press "Display: images" to view one of these images - it will have dimensions $32 \times 32 \times 101$.

To perform statistics on these images:

- Create a new directory, eg. `mkdir XYTstats`.

- Press the "Specify 2nd level" button.

- Select "two-sample t-test" (unpaired t-test)

- Define the images for "Group 1" as all those in the subdirectory `type_standard` (using right mouse, and "select all") and the images for "Group 2" as all those in the subdirectory `type_rare`.

- Finally, specify the new `XYTstats` directory as the output directory.

- Press the "save" icon, top left, and save this design specification as `mmn_design.mat` and press "save".

- Press the green "Run" button to execute the job[4] This will produce the design matrix for a two-sample t-test.

- Now press "Estimate" in SPMs main window, and select the `SPM.mat` file from the `XYTstats` directory.

Now press "Results" and define a new F-contrast as [1 -1] (for help with these basic SPM functions, see eg. chapter 10). Keep the default contrast options, but threshold at $p < .05$ FWE corrected for the whole search volume and select "Scalp-Time" for the "Data Type". Then press "whole brain", and the Graphics window should now look like that in Figure 18.3. This reveals a large fronto-central region within the 2D sensor space and within the time epoch in which standard and rare trials differ reliably, having corrected for multiple F-tests across pixels/time. An F-test is used because the sign of the difference reflects the polarity of the ERP difference, which is not of primary interest.

The cursor in Figure 18.3 has been positioned by selecting the second cluster in the results table. This occurs at time point 160ms post stimulus.

Now:

- Press the right mouse button in the MIP

- Select "display/hide channels"

- Select the `maedffMspm8_subject1.mat` file.

This links the `SPM.mat` file with the M/EEG file from which the EEG images were created. It is now possible to superimpose the channel labels onto the spatial SPM, and also to "goto the nearest channel" (using options provided after a right mouse click, when navigating the MIP).

We have demonstrated sensor space analysis for single-subject data. More frequently, one would compute ERP images for each subject, smooth them, and then perform paired t-tests over subjects to look for condition differences. See [26] for a group analysis of MMN data.

Finally, if one had more constrained a priori knowledge about where and when the differences would appear, one could perform a Small Volume Correction (SVC) based on, for example, a box around fronto-central channels and between 100 and 200ms poststimulus. We also refer the reader to chapter 5 for further details on sensor space analysis.

18.3 Source reconstruction

Source reconstruction comprises forward modeling and inverse modeling steps and is implemented by pressing the 3D source reconstruction button in SPM's top-left window. This brings up the source localisation GUI shown in Figure 18.4. The following subsections detail each of the steps in a source reconstruction analysis. We also advise the reader to consult the reference material in chapter 6.

18.3.1 Mesh

The first step is to load the data and create a cortical mesh upon which M/EEG data will be projected:

- Press the "Load" button in the souce localisation GUI and select the file `maedffMspm8_subject1.mat`.

- Enter "Standard" under "Comment/Label for this analysis" and press OK.

- Now press the "template" button.

- For "Cortical mesh", select "normal".

SPM will then form the "standard" or "canonical" cortical mesh shown in the Graphics window which, after rotation, should look like Figure 18.5

---

[4]Note that we can use the default "nonsphericity" selections, i.e, that the two trial-types may have different variances, but are uncorrelated.

Statistics: *p-values adjusted for search volume*

set-level		cluster-level				peak-level					mm mm ms
p	c	$p_{FWE-corr}$	$q_{FDR-corr}$	k_E	p_{uncorr}	$p_{FWE-corr}$	$q_{FDR-corr}$	F	$(Z_\equiv)$	p_{uncorr}	
0.000	8			3155		0.000	0.000	126.60	Inf	0.000	4 -25 300
						0.000	0.000	122.59	Inf	0.000	17 -30 300
						0.000	0.000	119.29	Inf	0.000	-4 -25 285
				666		0.000	0.000	70.29	Inf	0.000	4 2 160
						0.000	0.000	57.00	7.82	0.000	4 -9 160
						0.003	0.061	29.34	5.21	0.000	30 -19 145
				943		0.000	0.000	53.79	7.05	0.000	-34 -89 250
						0.000	0.000	52.40	6.96	0.000	-21 -95 250
						0.000	0.000	52.10	6.94	0.000	47 -79 265
				1031		0.000	0.000	42.68	6.29	0.000	26 45 305
						0.000	0.000	42.55	6.28	0.000	0 55 305
						0.000	0.000	42.18	6.26	0.000	17 45 305
				30		0.001	0.027	31.45	5.40	0.000	-51 -62 150
						0.025	0.509	24.10	4.71	0.000	-60 -35 155
				10		0.011	0.250	25.96	4.90	0.000	30 -9 100
				4		0.022	0.481	24.39	4.74	0.000	-21 -36 395
				3		0.029	0.577	23.70	4.67	0.000	-55 40 300

table shows 3 local maxima more than 8.0mm apart

Height threshold: F = 22.44, p = 0.000 (0.050)   Degrees of freedom = [1.0, 512.0]
Extent threshold: k = 0 voxels, p = 1.000 (0.050)   FWHM = 22.6 27.6 27.7 mm mm ms; 5.3 5.1 5.5 {voxels}
Expected voxels per cluster, <k> = 4.878   Volume: 9528813 = 83426 voxels = 505.4 resels
Expected number of clusters, <c> = 0.05   Voxel size: 4.3 5.4 5.0 mm mm ms; (resel = 150.97 voxels)
FWEp: 22.440, FDRp: 30.562

Figure 18.3: *In this SPM the time axis is reflected in the two MIP windows in the top row, with time proceeding from the bottom to the top of the page. The cursor has been positioned by selecting the third cluster in the results table. This occurs at time point 160ms post stimulus. The design matrix on the right hand side comprises two columns, the first for standard trials and the second for rare ones.*
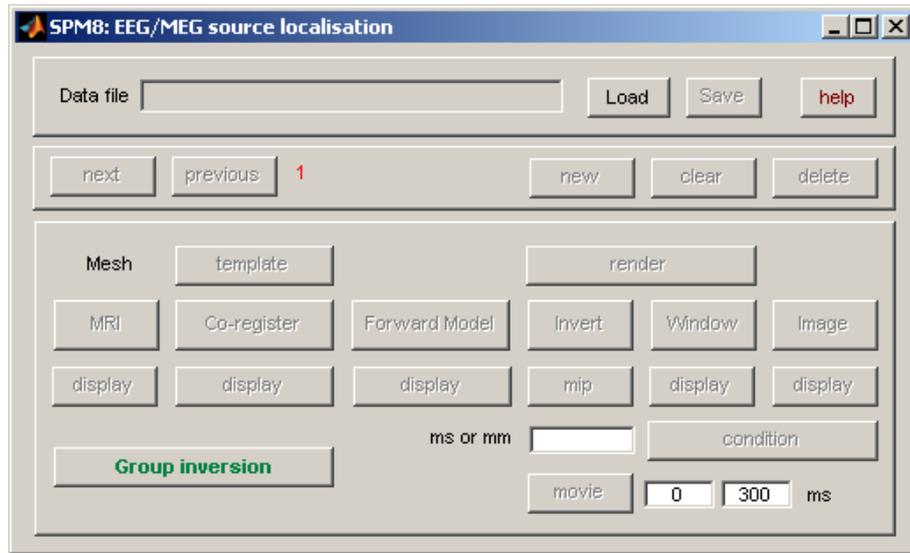
Figure 18.4: *Graphical user interface for 3D source localisation. A complete localisation comprises the following steps (i) creation of a cortical mesh, (ii) co-registration of the mesh with M/EEG data, (iii) creation of a forward model, and (iv) results interrogation. As each of these steps is completed the relevant part of the GUI becomes highlighted (text appears more solid).*
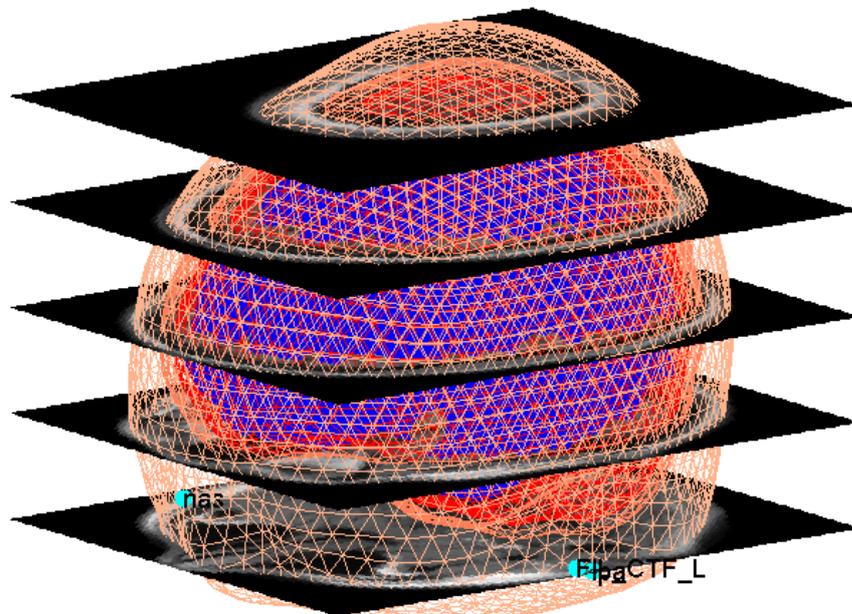
Figure 18.5: *The figure shows the canonical cortical mesh (blue), inner skull surface (red) and scalp surface (light brown). The hardwired fiducials are shown in light blue. Transverse slices of canonical MRI are also shown in black, with gray scale inlays showing anatomical detail.*
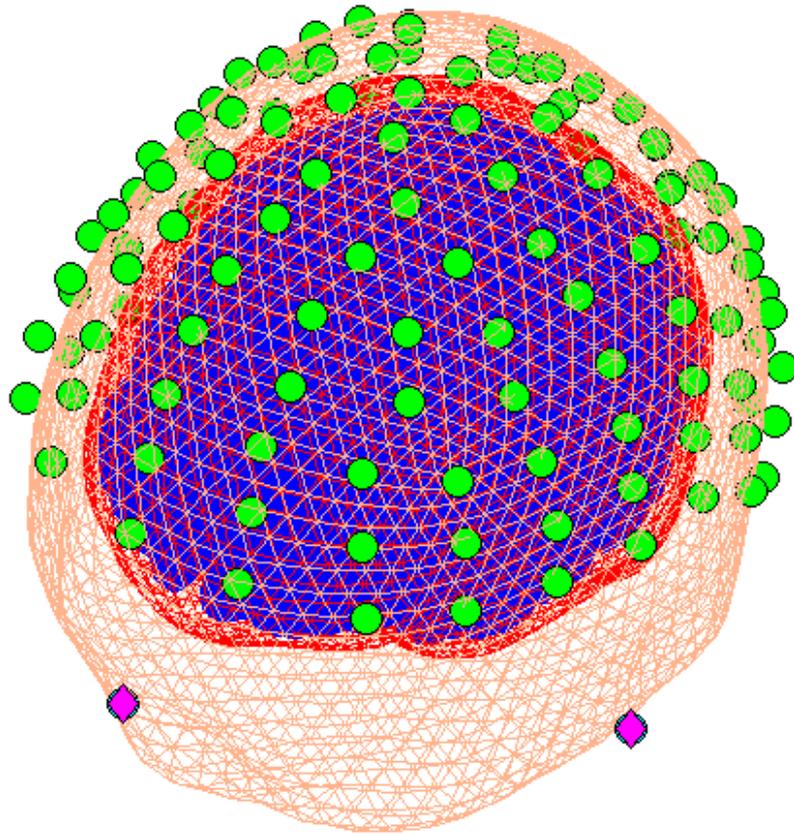
Figure 18.6: *The figure shows the MRI fiducials (pink), the sensor fiducials (blue) and the locations of sensors (green) in addition the the canonical cortical mesh (blue), inner skull surface (red) and scalp surface (light brown).*

18.3.2   Coregister

Now press the "Co-register" button. This will create further output in the Graphics window, the upper panel of which should like like Figure 18.6.

In this coregister step we were not required to enter any further parameters. However, if you are not using the template (or "canonical" mesh) or if at the "prepare" stage above you loaded your own (non-standard) sensor positions then you will be asked for the locations in MNI coordinates of the fiducial positions.

18.3.3   Forward model

Now press the "Forward model" button. Then select "EEG-BEM" in response to the "Which EEG head model?" question. SPM will then use a Boundary Element Method (BEM) which will take approximately 10 minutes to run. Upon completion SPM will write the `single_subj_T1_EEG_BEM.mat` file into the canonical subdirectory of your SPM distribution. The Graphics window should now appear as in Figure 18.7. The next time you wish to use an EEG-BEM solution based on the template mesh, SPM will simply use the date from the `single_subj_T1_EEG_BEM.mat` file (so this step will be much quicker the next time you do it). The same principle applies to EEG-BEM solutions computed from meshes based on subjects individual MRIs.

18.3.4   Invert

Now press the Invert button and
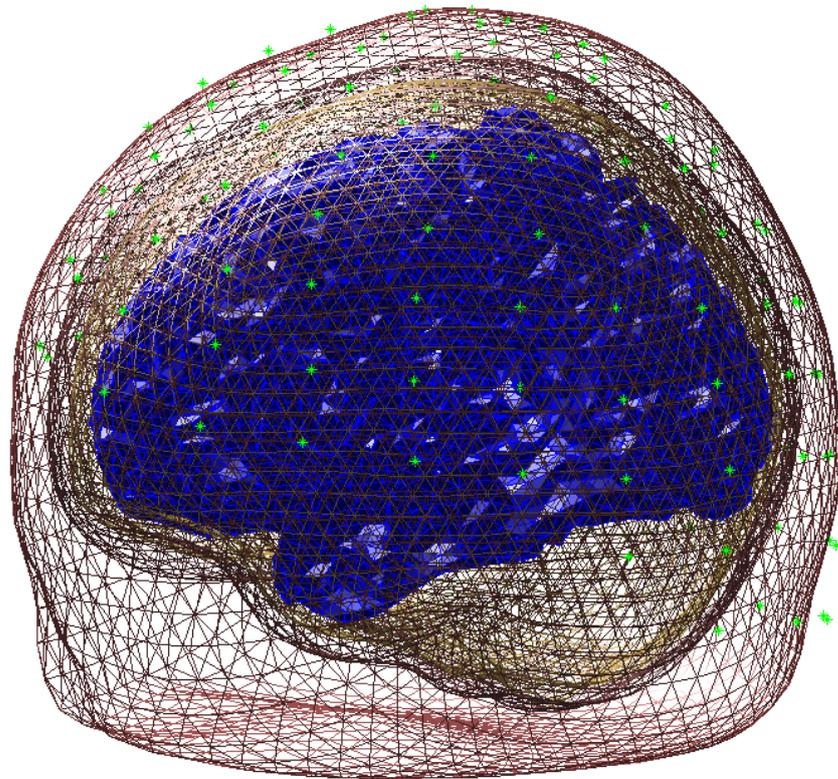
Figure 18.7: *The figure shows the cortical mesh (blue), brain, skull and scalp surfaces. Electrode positions are marked with asterisks.*

- Select an "Imaging" reconstruction.

- Select "Yes" for "All conditions or trials".

- Select "Standard" for Model.

SPM will now compute a leadfield matrix and save it in the file `SPMgainmatrix_maedffMspm8_subject1_1.mat` placed in `DATA_DIR`. This file can be replaced with one computed using other methods for computing the lead field (eg methods external to SPM). The forward model will then be inverted using the Multiple Sparse Priors (MSP) algorithm (the progress of which is outputted to the MATLAB command window). SPM will produce, in the Graphics window, (i) a Maximum Intensity Projection (MIP) of activity in source space (lower panel) and (ii) a time series of activity for (upper panel) each condition.

The "ms or mm" window has three functionalities (i) if you enter a single number this will be interpreted as ms, (ii) if you enter two numbers this will be interpreted as a time window for plotting movies (see below), (iii) if you enter 3 numbers this will be interpreted as MNI coordinates for a time series plot.

Now enter "160" for "ms or mm" and press the MIP button, to see a MIP of activity in source space at 160ms post-stimulus, and the time series of activities (top panel) at the position with largest magnitude signal. The corresponding graphic is shown in Figure 18.8. By toggling the "Condition" button, and pressing MIP each time, you can view the spatial distribution of activity for the different conditions (at the selected time point).

18.4   Dynamic Causal Modeling

Many of the functionalities of DCM for M/EEG are described in more detail in the reference chapter 8. In this chapter we demonstrate only the "DCM for ERP" model. Users are strongly encouraged to read the accompanying theoretical papers [13, 43]. Briefly, DCM for ERP fits a neural network model to M/EEG data, in which activity in source regions are described using differential equations based on neural mass models. Activity in each region comprises three populations of cells; pyramidal, local excitatory and local inhibitory. Fitting the model will then allow you to plot estimated activity in each cell population in each region. It will also provide estimates of the long range connections between regions, and show how these values are changed by experimental manipulation (eg. rare versus standard trial types).

In the `example_scripts` folder of the SPM distribution, we also provide an example script that will run a DCM-for-ERP analysis of this data. This can be edited to implement your own analysis.

Pressing the "DCM" button will open up the DCM GUI shown in Figure 18.9. We will now complete the three model specification entries shown in Figure 18.10:

- Press the "new data" button and select the `maedffMspm8_subject1.mat` file.

- Enter the "between-trial effects" and design matrix information shown in Figure 18.10(a).

- Press the "Display" button.

This completes the data specification stage. Now:

- Press the right hand arrow to move on to the specification of the electromagnetic model.

- Instead of "IMG" select "ECD" for the spatial characteristics of the sources.

- Now enter the names and (prior mean) locations of the sources shown in Figure 18.10(b).

- Pressing the "dipoles" button will create an interactive display in the graphics window showing the prior source positions.

This completes the specification of the electromagnetic model. Now:

- Press the right hand arrow (next to the dipoles button) to move on to specification of the neuronal model.
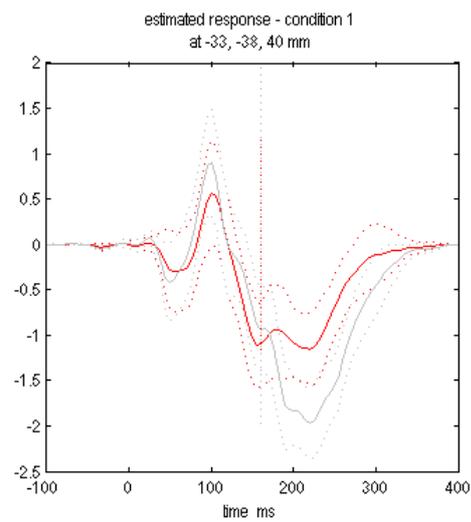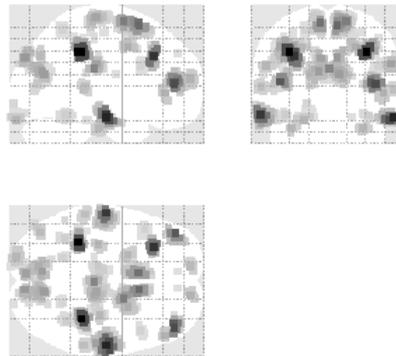
Figure 18.8: *Source reconstructed activity at 160ms post-stimulus. The upper trace shows responses to Condition 1 (Standards) with the red curve, and to Condition 2 (Rare) in gray.*
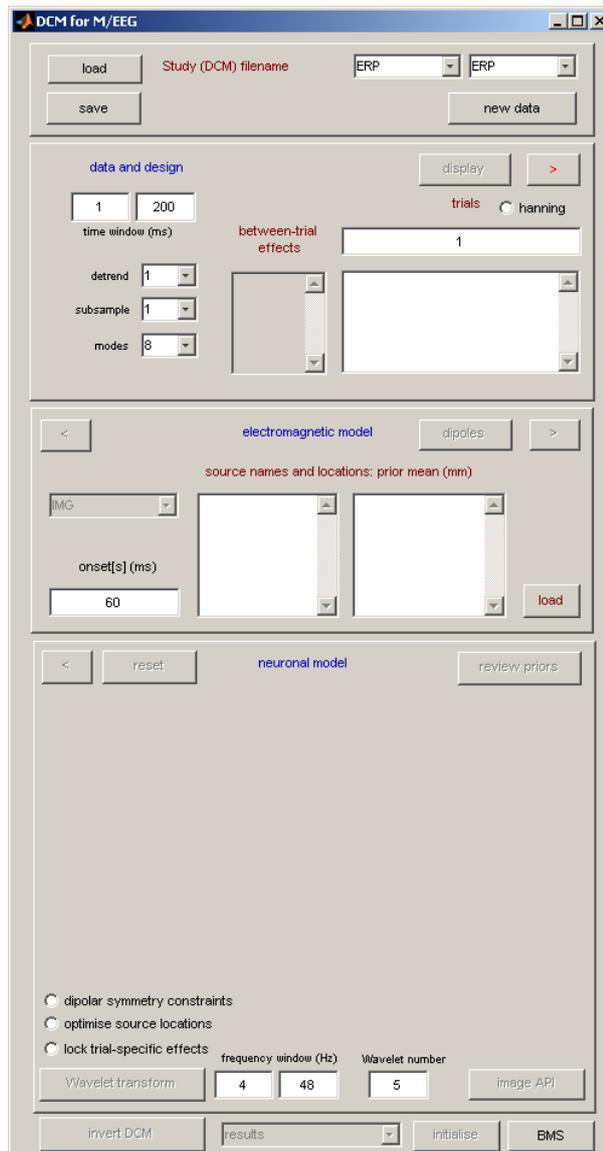
Figure 18.9: *The Dynamic Causal Modeling GUI splits model specification into three reversible phases (i) data and design, (ii) electromagnetic model and (iii) neuronal model. One can move forwards and backwards in the model specification using the left and right arrow buttons (these become highlighted when sufficient information has been entered to proceed to the next step).*
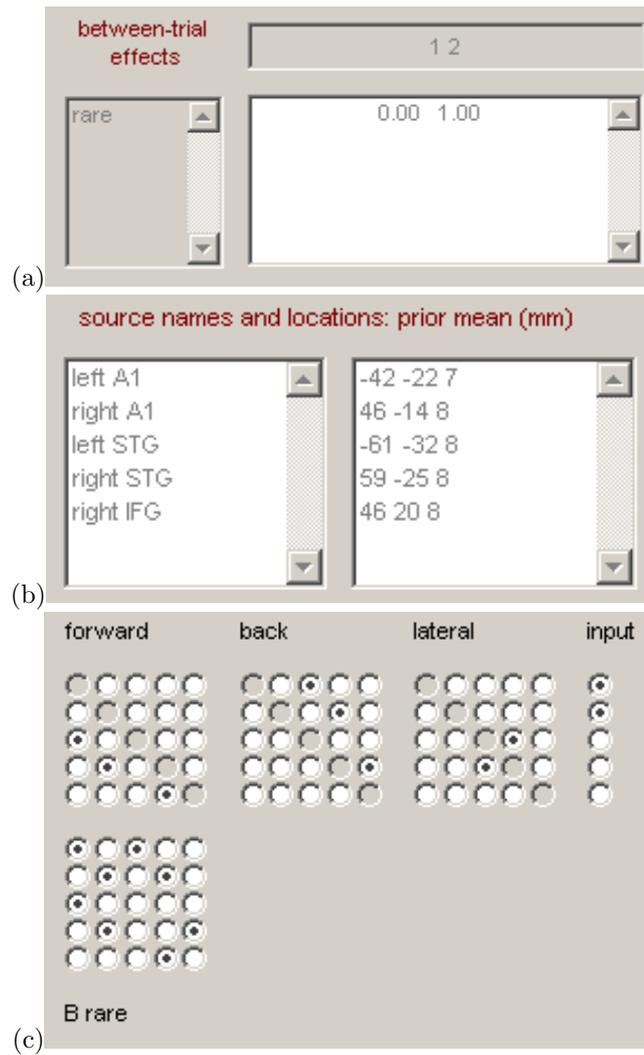
Figure 18.10: *Specification of DCM for ERP model (a) Data and design, (b) electromagnetic model and (c) neuronal model.*
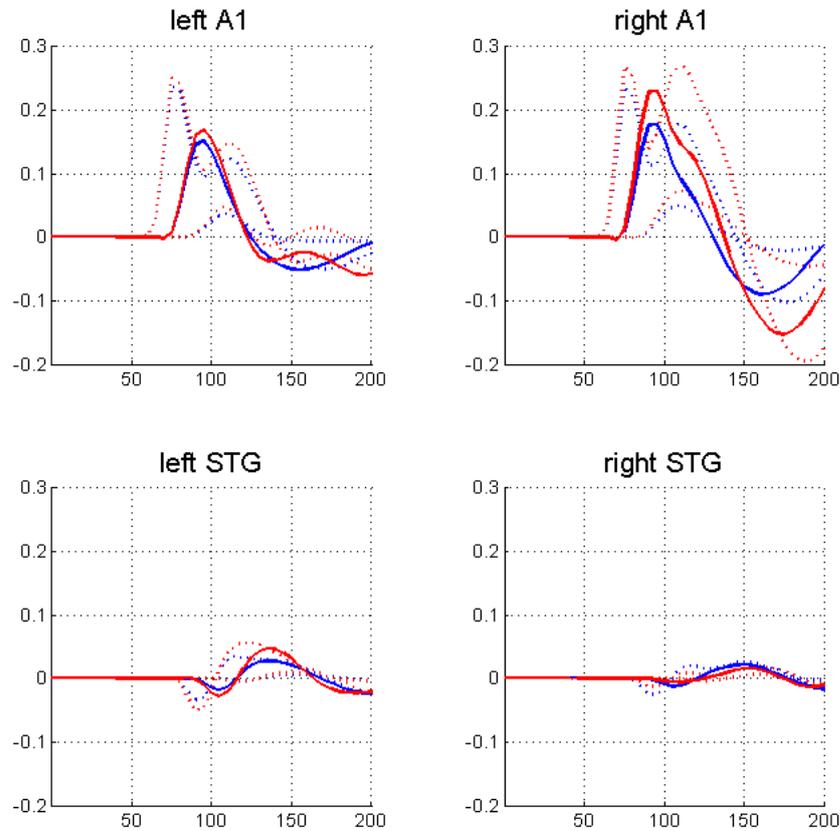
Figure 18.11: *Activity plots for three neuronal populations (solid lines for pyramidal cells, dotted lines for others), in four areas (fifth not shown in this figure), for standard (blue) and rare (red) trial types.*

- Highlight the connectivity matrix radio buttons so that they correspond to those shown in Figure 18.10(c).

- Press the (top left) 'save' button and accept the default file name.

- Press 'Invert DCM'

SPM will plot the progess of the model estimation in the MATLAB command window. Plots of data and the progressing model fit will be shown in SPM's graphics window. The algorithm should converge after five to ten minutes (in 64 iterations). Now select the "ERPs (sources)" option from the pull down menu to the right of the "Estimated" button. This will produce the plot shown in Figure 18.11. The values of the connections between areas can be outputted by selecting eg "Coupling(A)" from the pull-down menu in the DCM GUI. This will allow you to interrogate the posterior distribution of model parameters. It is also possible to fit multiple models, eg. with different numbers of regions and different structures, and to compare them using Bayesian Model Comparison. This is implemented by pressing the BMS button (bottom right hand corner of the DCM window).

Chapter 19

Multimodal face-evoked responses

19.1 Overview

This dataset contains EEG, MEG, functional MRI and structural MRI data on the same subject with 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). 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 275 channel CTF/VSM 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, 2009a, 2009b, in press), Kiebel and Friston (2004) and Friston et al (2006). To proceed with the data analysis, first download the data set from the SPM website[1]. Most of the analysis below can be implemented in Matlab 7.1 (R14SP3) and above. However, recoding condition labels using the GUI requires features of SPM8 only available in Matlab 7.4 (R2007a) and above. The Signal Processing toolbox is required for the filtering and downsampling steps.

19.2 Paradigm and Data

The basic paradigm involves randomised presentation of at least 86 faces and 86 scrambled faces (Figure 19.1), 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 content. 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.

---

[1]Multimodal face-evoked dataset: http://www.fil.ion.ucl.ac.uk/spm/data/mmfaces/
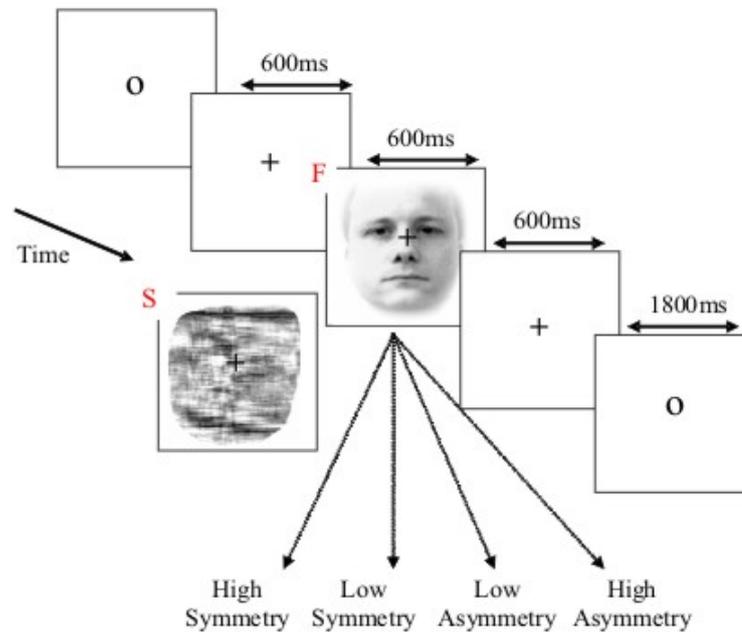
Figure 19.1: *One trial in the experiment. Trials involved either a Face (F) or Scrambled face (S).*

19.2.1   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 $1{\times}1{\times}1mm^3$ voxels, using a whole-body coil for RF transmission and an 8-element phased array head coil for signal reception.

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

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

The structural was manually positioned to roughly match Talairach space, with the origin close to the Anterior Commissure. The approximate position of 3 fiducials within this MRI space - the nasion, and the left and right peri-aricular points - are stored in the file:

```
sMRI/smri_fid.txt
```

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 implemented via digitised surfaces of the scalp ("head shape functions") that were created using the Polhemus 3D digitizer.

19.2.2   EEG data

The EEG data were acquired on a 128-channel ActiveTwo system, sampled at 2048 Hz, plus electrodes on left earlobe, right earlobe, and two bipolar channels to measure HEOG and VEOG. The 128 scalp channels are named: 32 A (Back), 32 B (Right), 32 C (Front) and 32 D (Left). The data acquired in two runs of the protocol are contained in two Biosemi raw data files:

```
EEG/faces_run1.bdf
EEG/faces_run2.bdf
```

The EEG directory also contains the following files:

```
EEG/condition_labels.txt
```

This text file contains a list of condition labels in the same order as the trials appear in the two files - "faces" for presentation of faces and "scrambled" for presentation of scrambled faces. The EEG directory also contains the following files:

```
EEG/electrode_locations_and_headshape.sfp
```

This ASCII file contains electrode locations, fiducials and headshape points measured with Polhemus digitizer.

The 3 fiducial markers were placed approximately on the nasion and preauricular points and digitised by the Polhemus digitizer. Later, we will coregister the fiducial points and the head shape to map the electrode positions in the "Polhemus space" to the "MRI space". Also included as reference are some SPM batch files and SPM scripts (though these are recreated as part of the demo):

```
EEG/batch_eeg_XYTstats.mat
EEG/batch_eeg_artefact.mat
EEG/eeg_preprocess.m
EEG/faces_eeg_montage.m
```

19.2.3   MEG data

The MEG data were acquired on a 275 channel CTF/VSM system, using second-order axial gradiometers and synthetic third gradient for denoising and sampled at 480 Hz. There are actually 274 MEG channels in this dataset since the system it was recorded on had one faulty sensor. Two runs (sessions) of the protocol have been saved in two CTF datasets (each one is a directory with multiple files)

```
MEG/SPM_CTF_MEG_example_faces1_3D.ds
MEG/SPM_CTF_MEG_example_faces2_3D.ds
```

The MEG data also contains a `headshape.mat` file, containing the headshape recorded during the MEG experiment with a Polhemus digitizer.

The locations of the 3 fiducials in the `headshape.mat` file are the same as the positions of 3 "locator coils" the locations of which are measured by the CTF machine, and used to define the coordinates (in "CTF space") for the location of the 274 sensors.

Also included as reference are two SPM batch files and two trial definition files (though these are recreated as part of the demo):

```
MEG/batch_meg_preprocess.mat
MEG/batch_meg_TFstats.mat
MEG/trials_run1.mat
MEG/trials_run2.mat
```

19.2.4   fMRI data

The fMRI data were acquired using a gradient-echo EPI sequence on a 3T Siemens TIM Trio, with 32, 3mm slices (skip 0.75mm) of $3 \times 3mm^2$ pixels, acquired in a sequential descending order with a TR of 2s. There are 390 images in each of the two "Session" sub-directories (5 initial dummy scans have been removed), each consisting of a NIfTI image and header file:

```
fMRI/Session1/fM*.{hdr,img}
fMRI/Session2/fM*.{hdr,img}
```

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

```
fMRI/trials_ses1.mat
fMRI/trials_ses2.mat
```

and two example SPM batch files (see Section 19.6):

```
fMRI/batch_fmri_preproc.mat
fMRI/batch_fmri_stats.mat
```

19.3   Getting Started

You need to start SPM8 and toggle "EEG" as the modality (bottom-right of SPM main window), or start SPM8 with `spm eeg`. In order for this to work you need to ensure that the main SPM directory is on your MATLAB path.

19.4   EEG analysis

First change directory to the EEG subdirectory (either in MATLAB or via the "CD" option in the SPM "Utils" menu).

19.4.1   Convert

Press the CONVERT button and select the `faces_run1.bdf` file. At the prompt "Define settings?" select "just read". SPM will now read the original Biosemi format file and create an SPM compatible data file, called `spm8_faces_run1.mat` and `spm8_faces_run1.dat` in the current MATLAB directory. After the conversion is complete the data file will be automatically opened in SPM8 reviewing tool. By default you will see the "info" tab. At the top of the window there is some basic information about the file. Below it you will see several clickable tabs with additional information. The "history" tab lists the processing steps that have been applied to the file. At this stage there is only one such step - conversion. The "channels" tab lists the channels in the file and their properties, the "trial" tab lists the trials or in the case of a continuous file all the triggers (events) that have been recorded. The "inv" tab is used for reviewing the inverse solutions and is not relevant for the time being. Note that the detailed information in the tabs will not be available for MATLAB versions older than 7.4. At the top of the window there is another set of tabs. If you click on the "EEG" tab you will see the raw EEG traces. They all look unusually flat because the continuous data we have just converted contains very low frequencies and baseline shifts. Therefore, if we try to view all the channels together, this can only be done with very low gain. If you press the "intensity rescaling" button (with arrows pointing up and down) several times you will start seeing EEG activity in a few channels but the other channels will not be visible as they will go out of range. You can also use the controls at the bottom of the window to scroll through the recording. If you press the icon to the right of the mini-topography icon, with the rightwards pointing arrow, the display will move to the next trigger, shown as a vertical line through the display. (New triggers/events can be added by the rightmost icon). At the bottom of the display is a plot of the global field power across the session, with the black line indicating the current timewindow displayed (the width of this timewindow can be controlled by the two leftmost top icons).

19.4.2   Downsample

Here, we will downsample the data in time. This is useful when the data were acquired like ours with a high sampling rate of 2048 Hz. This is an unnecessarily high sampling rate for a simple evoked response analysis, and we will now decrease the sampling rate to 200 Hz, thereby reducing the file size by more than ten fold and greatly speeding up the subsequent processing steps. This step requires the Signal Processing toolbox. Select DOWNSAMPLE from the "Other" drop-down menu and select the `spm8_faces_run1.mat` file. Choose a new sampling rate of 200 (Hz). The progress bar will appear and the resulting data will be saved to files `dspm8_faces_run1.mat` and `dspm8_faces_run1.dat`. Note that this dataset and other intermediate datasets created during preprocessing will not be automatically opened in the reviewing tool, but you can always review them by selecting `M/EEG` from the "Display" drop down menu and choosing the corresponding `.mat` file.

19.4.3   Montage

In this step, we will identify the VEOG and HEOG channels, remove several channels that don't carry EEG data and are of no importance to the following and convert the 128 EEG channels

to "average reference" by subtracting the mean of all the channels from each channel[2]. We generally recommend removal of data channels that are no longer needed because this will reduce the total file size and conversion to average reference is necessary at present for source modelling to work correctly. To do so, we use the MONTAGE tool in SPM, which is a general approach for pre-multiplying the data matrix (channels × time) by another matrix that linearly weights all channel data. This provides a very general method for data transformation in M/EEG analysis.

The appropriate montage-matrix can be specified in SPM by either using a graphical interface, or by supplying the matrix saved in a file. We will do the latter. The script to generate this file is `faces_eeg_montage.m`. Running this script will produce a file named `faces_eeg_montage.mat`. In our case, we would like to keep only channels 1 to 128. To re-reference each of these to their average, the script uses MATLAB "detrend" to remove the mean of each column (of an identity matrix). In addition, there were four EOG channels (131, 132, 135, 136), where the HEOG is computed as the difference between channels 131 and 132, and the VEOG by the difference between channels 135 and 136.

You now call the montage function by choosing MONTAGE in the "Other" drop-down menu and:

- Select the M/EEG-file `dspm8_faces_run1.mat`.

- "How to specify the montage ?" Answer "file".

- Then select the generated `faces_eeg_montage.mat` file.

- "Keep the other channels?" : "No".

This will remove the uninteresting channels from the data. The progress bar appears and SPM will generate two new files `Mdspm8_faces_run1.mat` and `Mdspm8_faces_run1.dat`.

19.4.4 Epoch

To epoch the data click on EPOCHING. Select the `Mdspm8_faces_run1.mat` file. Choose the peri-stimulus time window, first the start `-200`, then the end `600` ms. Choose 1 condition. There is no information in the file at this stage to distinguish between faces and scrambled faces. We will add this information at a later stage. You can give this condition any label, for instance "stim". A GUI pops up which gives you a complete list of all events in the EEG file. Each event has type and value which might mean different things for different EEG and MEG systems. So you should be familiar with your particular system to find the right trigger for epoching. In our case it is not very difficult as all the events but one appear only once in the recording, whereas the event with type "STATUS" and value 1 appears 172 times which is exactly the number of times a visual stimulus was presented. Select this event and press OK. Answer two times "no" to the questions "review individual trials", and "save trial definitions". The progress bar will appear and the epoched data will be saved to files `eMdspm8_faces_run1.mat` and `eMdspm8_faces_run1.dat`. The epoching function also performs baseline correction by default (with baseline -200 to 0ms). Therefore, in the epoched data the large channel-specific baseline shifts are removed and it is finally possible to see the EEG data clearly in the reviewing tool.

19.4.5 Reassignment of trial labels

Open the file `eMdspm8_faces_run1.mat` in the reviewing tool (under "Display" button). The first thing you will see is that in the history tab there are now 4 processing steps. Now switch to the "trials" tab. You will see a table with 172 rows - exactly the number of events we selected before. In the first column the label "stim" appears in every row. What we would like to do now is change this label to "faces" or "scrambled" where appropriate. We should first open the file `condition_labels.txt` (in the EEG directory) with any text editor, such as MATLAB editor or Windows notepad. In this file there are exactly 172 rows with either "faces" or "scrambled" in each row. Select and copy all the rows (Ctrl-A, Ctrl-C on Windows). Then go back to SPM

---

[2]Re-referencing EEG to the mean over EEG channels is important for source localisation. Note also that if some channels are subsequently marked "bad" (see later), one should re-reference again, because bad channels are ignored in any localisation.

and the trials tab. Place the cursor in the first row and first column cell with the "stim" label and paste the copied labels (Ctrl-V). The new labels should now appear for all rows. Press the "update" button above the table and then the "SAVE" button at the top right corner of the window. The new labels are now saved in the dataset.

19.4.6 Using the history and object methods to preprocess the second file

At this stage we need to repeat the preprocessing steps for the second file `faces_run2.bdf`. You can do it by going back to the "Convert" section and repeating all the steps for this file, but there is a more efficient way. If you have been following the instructions until now the file `eMdspm8_faces_run1.mat` should be open in the reviewing tool. If it is not the case, open it. Go to the "history" tab and press the "Save as script" button. A dialog will appear asking for the name of the MATLAB script to save. Let's call it `eeg_preprocess.m`. Then there will be another dialogue suggesting to select the steps to save in the script. Just press "OK" to save all the steps. Now open the script in the MATLAB editor. You will now need to make some changes to make it work for the second file. Here we suggest the simplest way to do it that does not require familiarity with MATLAB programming. But if you are more familar with MATLAB you'll definitely be able to do a much better job. First, replace all the occurences of "run1" in the file with "run2". You can use the "Find & Replace" functionalty (Ctrl-F) to do it. Secondly, erase the line starting with `S.timewindow` (line 5). This line defines the time window to read, in this case from the first to the last sample of the first file. The second file is slightly longer than the first so we should let SPM determine the right time window automatically. Save the changes and run the script by pressing the "Run" button or writing `eeg_preprocess` in the command line. SPM will now automatically perform all the steps we have done before using the GUI. This is a very easy way for you to start processing your data automatically once you come up with the right sequence of steps for one file. After the script finishes running there will be a new set of files in the current directory including `eMdspm8_faces_run2.mat` and `eMdspm8_faces_run2.dat`. If you open these files in the reviewing tool and go to the "trials" tab you will see that the trial labels are still "stim". The reason for this is that updates done using the reviewing tool are not presently recorded in the history (with the exception of the "Prepare" interface, see below). You can still do this update automatically and add it to your script. If you write `D` in the command line just after running the script and press "Enter" you will see some information about the dataset `eMdspm8_faces_run2`. `D` is an object, this is a special kind of data structure that makes it possible to keep different kinds of related information (in our case all the properties of our dataset) and define generic ways of manipulating these properties. For instance we can use the command:

```
D = conditions(D, [], importdata('condition_labels.txt')); D.save;
```

to update the trial labels using information imported from the `condition_labels.txt`[3] (the two runs had identical trials). Now, `conditions`' is a "method", a special function that knows where to store the labels in the object. All the methods take the M/EEG object (usually called `D` in SPM by convention) as the first argument. The second argument is a list of indices of trials for which we want to change the label. We specify an empty matrix which is interpreted as "all". The third argument is the new labels which are imported from the text file using a MATLAB built-in function. We then save the updated dataset on disk using the `save` method. If you now write `D.conditions` or `conditions(D)` (which are two equivalent ways of calling the `conditions` method with just `D` as an argument), you should see a list of 172 labels, either "faces" or "scrambled". If you add the commands above at the end of your automatically generated script, you can run it again and this time the output will have the right labels.

19.4.7 Merge

We will now merge the two epoched files we have generated until now and continue working on the merged file. Select the "Merge" command from the "Other" drop-down menu. In the selection

---

[3]You might need to change the full path to this text file inside the single quotes, depending on your current directory and the directory of the original data.

window that comes up click on `eMdspm8_faces_run1.mat` and `eMdspm8_faces_run2.mat`. Press "done". Answer "Leave as they are" to "What to do with condition labels?". This means that the trial labels we have just specified will be copied as they are to the merged file. A new dataset will be generated called `ceMdspm8_faces_run1.{mat,dat}`.

19.4.8   Prepare

In this section we will add the separately measured electrode locations and headshape points to our merged dataset. In principle, this step is not essential for further analysis because SPM8 automatically assigns electrode locations for commonly used EEG caps and the Biosemi 128 cap is one of these. Thus, default electrode locations are present in the dataset already after conversion. But since these locations are based on channel labels they may not be precise enough and in some cases may be completely wrong because sometimes electrodes are not placed in the correct locations for the corresponding channel labels. This can be corrected by importing individually measured electrode locations. Select PREPARE from the "Other" menu and in the file selection window select `ceMdspm8_faces_run1.mat`. A menu will appear at the top of SPM interactive window (bottom left window). In the "Sensors" submenu choose "Load EEG sensors"/"Convert locations file". In the file selection window choose the `electrode_locations_and_headshape.sfp` file (in the original EEG directory). Then from the "2D projection" submenu select "Project 3D (EEG)". A 2D channel layout will appear in the Graphics window. Select "Apply" from "2D Projection" and "Save" from "File" submenu. Note that the same functionality can also be accessed from the reviewing tool by pressing the "Prepare SPM file" button.

19.4.9   Artefact rejection

Here we will use SPM8 artefact detection functionality to exclude from analysis trials contaminated with large artefacts. Press the ARTEFACTS button. A window of the SPM8 batch interface will open. You might already be familiar with this interface from other SPM8 functions. It is also possible to use the batch interface to run the preprocessing steps that we have performed until now, but for artefact detection this is the only graphical interface and there is no way to configure it with the usual GUI buttons. Click on "File name" and select the `ceMdspm8_faces_run1.mat` file. Double click "How to look for artefacts" and a new branch will appear. It is possible to define several sets of channels to scan and several different methods for artefact detection. We will use simple thresholding applied to all channels. Click on "Detection algorithm" and select "Threshold channels" in the small window below. Double click on "Threshold" and enter 200 (in this case μV). The batch is now fully configured. Run it by pressing the green button at the top of the batch window.

   This will detect trials in which the signal recorded at any of the channels exceeds 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. The procedure will also detect channels in which there is 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.
39 rejected trials: 38   76   82   83   86   88   89   90   92   [...]
```

(leaving 305 valid trials). A new file will also be created, `aceMdspm8_faces_run1.{mat,dat}`.

19.4.10   Exploring the M/EEG object

We can now review the preprocessed dataset from the MATLAB command line by typing:

```
D = spm_eeg_load
```

and selecting the `aceMdspm8_faces_run1.mat` file. This will print out some basic information about the M/EEG object `D` that has been loaded into MATLAB workspace.

```
SPM M/EEG data object
Type: single
Transform: time
2 conditions
130 channels
161 samples/trial
344 trials
Sampling frequency: 200 Hz
Loaded from file  ...\EEG\aceMdspm8_faces_run1.mat
Use the syntax D(channels, samples, trials) to access the data.
```

Note that the data values themselves are memory-mapped from `aceMdspm8_faces_run1.dat` and can be accessed by indexing the D object (e.g, `D(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 trials (`D.ntrials`). Typing `D.conditions` will show the list of condition labels consisting of 172 faces ("faces") and 172 scrambled faces ("scrambled"). `D.reject` will return a 1×344 vector of ones (for rejected trials) and zeros (for retained trials). `D.condlist` will display a list of unique condition labels. The order of this list is important because every time SPM needs to process the conditions in some order, this will be the order. If you type `D.chanlabels`, you will see the order and the names of the channels. `D.chantype` will display the type for each channel (in this case either "EEG" or "EOG"). `D.size` will show the size of the data matrix, [130 161 344] (for channels, samples and trials respectively). The size of each dimension separately can be accessed by `D.nchannels`, `D.nsamples` and `D.ntrials`. Note that although the syntax of these commands is similar to those used for accessing the fields of a struct data type in MATLAB what's actually happening here is that these commands evoke special functions called "methods" and these methods collect and return the requested information from the internal data structure of the D object. The internal structure is not accessible directly when working with the object. This mechanism greatly enhances the robustness of SPM code. For instance you don't need to check whether some field is present in the internal structure. The methods will always do it automatically or return some default result if the information is missing without causing an error.

Type `methods('meeg')` for the full list of methods performing operations with the object. Type `help meeg/method name` to get help about a method.

19.4.11   Basic ERPs

Press the AVERAGING button and select the `aceMdspm8 faces run1.mat` file. At this point you can perform either ordinary averaging or "robust averaging" (Wager et al., 2005). Robust averaging makes it possible to suppress artefacts automatically without rejecting trials or channels completely, but just the contaminated parts. Thus, in principle we could do robust averaging without rejecting trials with eye blinks and this is something you can do as an exercise and see how much difference the artefact rejection makes with ordinary averaging vs. robust averaging. For robust averaging answer "yes" to "Use robust averaging?". Answer "yes" to "Save weights", and "no" to "Compute weights by condition"[4].

Finally, press "Enter" to accept the default "Offset of the weighting function". A new dataset will be generated `maceMdspm8 faces run1.{mat,dat}` ("m" for "mean") and automatically opened in the reviewing tool so that you can examine the ERP. There will also be an additional dataset named `WaceMdspm8 faces run1.{mat,dat}`. This dataset will contain the weights used by robust averaging. This is useful to see what was suppressed and whether there might be some condition-specific bias that could affect the results.

Select "Contrast" from the "Other" pulldown menu on the SPM window. This function creates linear contrasts of ERPs/ERFs. Select the `maceMdspm8 faces run1.mat` file, enter $[1 - 1]$ as the first contrast and label it "Difference", answer "yes" to "Add another", enter $[1/2 \, 1/2]$ as the second contrast and label it "Mean". Press "no" to the question "Add another" and not to

---

[4]When there are approximately equal numbers of trials in each condition, as here, it is probably safer to compute weights across all conditions, so as not to introduce artifactual differences between conditions. However, if one condition has fewer trials than the others, it is likely to be safer to estimate the weights separately for each condition, otherwise evoked responses in the rarer condition will be downweighted so as to become more similar to the more common condition(s).
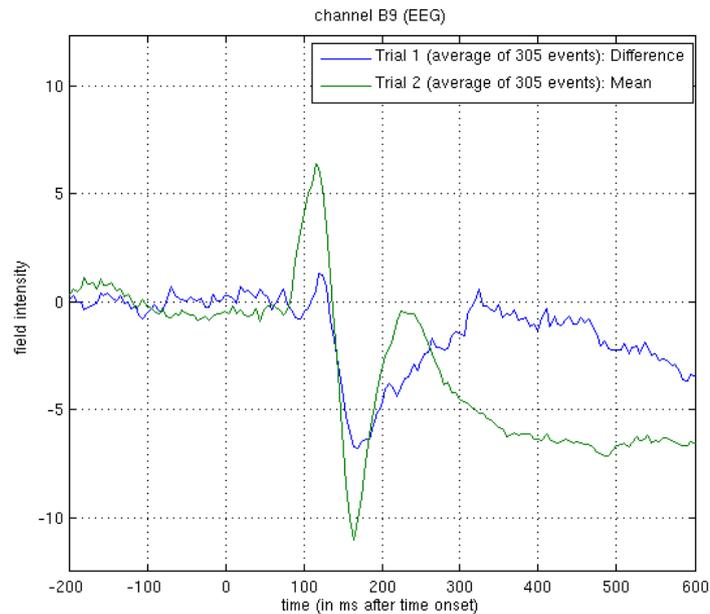
Figure 19.2: *Average (green) and differential (blue) ERPs for faces and scrambled faces at channel B9 in `wmaceMdspm8_faces_run1.mat`.*

"weight by num replications". This will create new file `wmaceMdspm8_faces_run1.{mat,dat}`, 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 `wmaceMdspm8_faces_run1.mat` file. Switch to the "EEG" tab and to "scalp" display by toggling a radio button at the top of the tab. The Graphics window should then show the ERP for each channel (for Trial 1 the "Difference" condition). Hold SHIFT and select Trial 2 to see both conditions superimposed. Then click on the zoom button and then 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 19.2.

The green 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. The difference is small around the P1 latency, but large and 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 19.4.13.

To see the topography of the differential ERP, click on Trial 1 again, press the "topography" icon button at the top of the window and scroll the latency from baseline to the end of the epoch. You should see a maximal difference around 180ms as in Figure 19.3 (possibly including a small delay of about 8ms for the CRT display to scan to the centre of the screen).

19.4.12   3D SPMs (Sensor Maps over Time)

A 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 (using the 2D layout we created earlier) and interpolating between them to create an image of $M \times M$ pixels (when M is user-specified, eg $M = 32$). This would allow one to identify locations where, for example, the ERP amplitude in two conditions at a given timepoint differed reliably across subjects, having corrected for the multiple t-tests performed across pixels. That correction uses Random Field Theory, which takes into account the spatial correlation across pixels (i.e, that the tests are not independent). 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
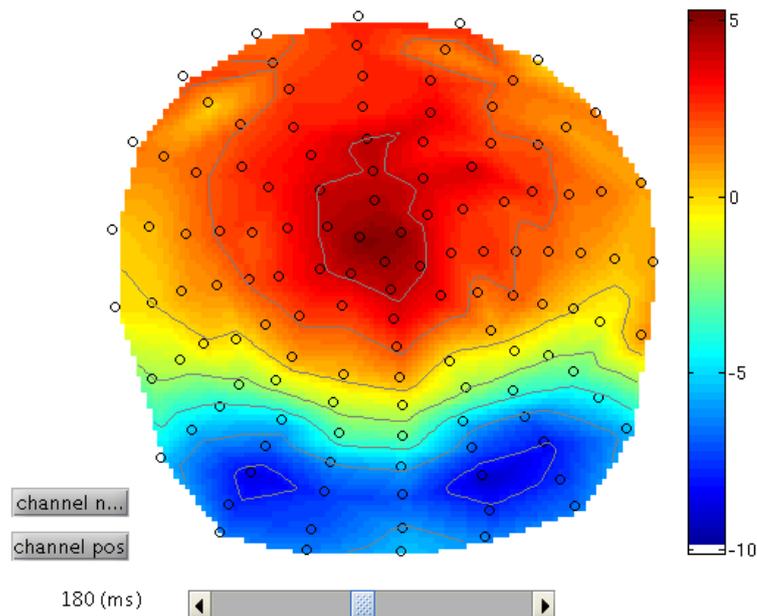
Figure 19.3: *2D topography for faces minus scrambled faces at 180ms.*

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 $M \times M \times S$, 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[5].

Select the "Convert to images" option in the "Other" menu in the SPM main window, and select the `aceMdspm8_faces_run1.mat` file. You will then be prompted for "output image dimensions", for which you can accept the default of 32 (leading to a 32×32 pixel space). It will then ask whether you want to interpolate or mask out bad channels, for which you 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 `aceMdspm8_faces_run1`, which will itself contain two subdirectories, one for each trialtype. In each trialtype subdirectory there will be image and header files for each non-rejected trial of that type, e.g, `trial0002.{hdr,img}`. You can press "Display: images" to view one of these images - it will have dimensions $32 \times 32 \times 161$, with the origin set at [16 18.6 41] (where 41 samples is 0ms), as in Figure 19.4.

Smoothing

Note that you can also smooth these images in 3D (i.e, in space and time) by pressing "Smooth Images" from the "Others" pulldown menu. When you get the Batch Editor window, you can enter a smoothness of your choice (eg [9 9 20], or 9mm by 9mm by 20ms). Note that you should also change the default "Implicit masking" from "No" to "Yes"; this is to ensure that the smoothing does not extend beyond the edges of the topography.

As with fMRI, smoothing can improve statistics if the underlying signal has a smoothness close to the smoothing kernel (and the noise does not; the matched filter theorem). Smoothing

---

[5]Note that the 2D location in sensor space for EEG will depend on the choice of montage.
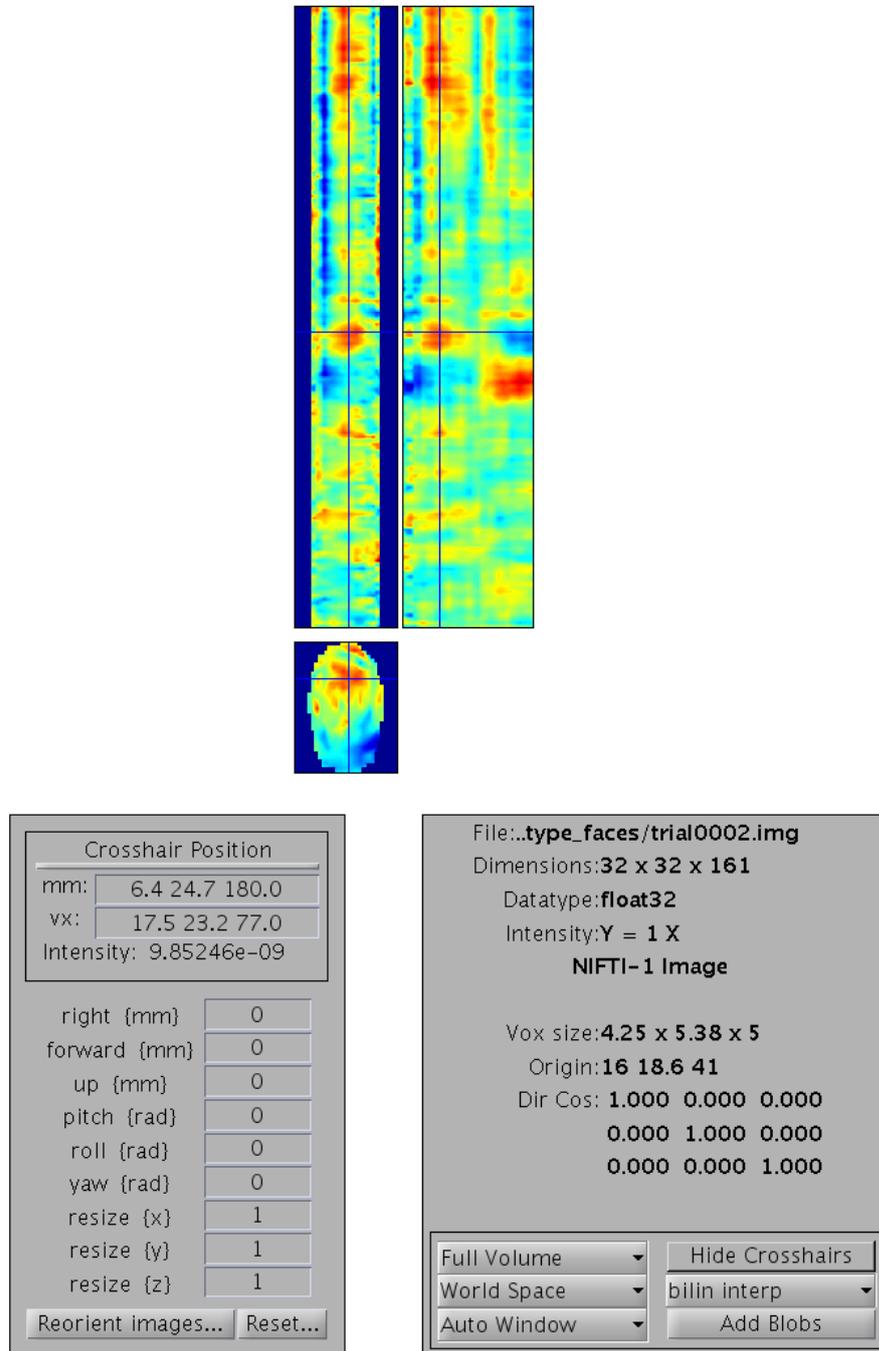
Figure 19.4: *3D image for trial 2 of `aceMdspm8_faces_run1.mat`. The bottom image is a 2D x-y space interpolated from the flattened electrode locations (at one point in time). The two top images are sections through x and y respectively, now expressed over time (vertical (z) dimension).*

may also be necessary if the final estimated smoothness of the SPMs (below) is not at least three times the voxel size; an assumption of Random Field Theory. In the present case, the data are already smooth enough (as you can check below), so we do not have to smooth further.

Stats

To perform statistics on these images, first create a new directory, eg. `mkdir XYTstats`.

Then press the "Specify 2nd level" button, to produce the batch editor window again. Select the new `XYTstats` as the "Directory", and "two-sample t-test" (unpaired t-test) as the "Design". Then select the images for "group 1 scans" as all those in the subdirectory "type_faces" (using right click, and "select all") and the images for "group 2 scans" as all those in the subdirectory "type_scrambled". You might want to save this batch specification, but then press "run"[6].

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]. Keep the default contrast options, but threshold at $p < .05$ FWE corrected for the whole search volume and select "Scalp-Time" for the "Data Type". Then press "whole brain", and the Graphics window should now look like that in Figure 19.5.

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 and time. There are a number of such regions, but the largest has maxima at [-13 -78 180] and [21 -68 180], corresponding to left and right posterior sites at 180ms.

To relate these coordinates back to the original sensors, right-click in some white space in the top half of the Graphics window, to get a menu with various options. First select "goto global maxima". The red cursor should move to coordinates [-13, -78, 180]. (Note that you can also overlap the sensor names on the MIP by selecting "display/hide channels" - though it can get a bit crowded!). Then right-click again to get the same menu, but this time select "go to nearest suprathreshold channel". You will be asked to select the original EEG/MEG file used to create the SPM, which in this case is the `aceMdspm8_faces_run1.mat` file. This should output in the Matlab window:

```
spm_mip_ui: Jumped 4.25mm from [-13, -78, 180],
to nearest suprathreshold channel (A15) at [ -8, -78, 180]
```

In other words, it is EEG channel "A15" that shows the greatest face/scrambled difference over the epoch (itself maximal at 180ms).

Note that 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). Indeed, if you plot the contrast of interest from the cluster maxima, you will see that the difference is negative for the first posterior, cluster but positive for the second, central cluster. This is consistent with the polarity of the differences in Figure 19.3[7].

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. See http://imaging.mrc-cbu.cam.ac.uk/meg/SensorSpm for more details.

If you go to the global maximum, then press "overlays", "sections" and select the "mask.img" in the stats directory, you will get sections through the space-time image. A right click will reveal the current scalp location and time point. By moving the cursor around, you can see that the N170/VPP effects start to be significant (after whole-image correction) around 150ms (and may also notice a smaller but earlier effect around 100ms).

19.4.13   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

---

[6]Note that we can use the default "nonsphericity" selections, i.e, that the two trial-types may have different variances, but are uncorrelated.

[7]The former likely corresponds to the "N170", while the latter likely corresponds to the "VPP", which may be two signs of the same effect, though of course these effects depend on the choice of reference.

Faces vs Scrambled

SPM{F$_{1,303}$}

SPMresults: ./multimodal/EEG/XYTstats
Height threshold F = 26.067509 {p<0.05 (FWE)}
Extent threshold k = 0 voxels

contrast(s)

Design matrix

Statistics: *p-values adjusted for search volume*

set-level		cluster-level				peak-level					mm mm ms
p	c	$p_{FWE-corr}$	$q_{FDR-corr}$	k_E	p_{uncorr}	$p_{FWE-corr}$	$q_{FDR-corr}$	F	$(Z_\equiv)$	p_{uncorr}	
0.000	**18**			**4025**		**0.000**	**0.000**	**233.47**	**Inf**	**0.000**	**−13 −78 180**
						0.000	0.000	232.38	Inf	0.000	21 −68 180
						0.000	0.000	217.35	Inf	0.000	26 −57 185
				1458		**0.000**	**0.000**	**102.32**	**Inf**	**0.000**	**0 8 180**
						0.000	0.000	98.19	Inf	0.000	0 29 185
						0.000	0.000	92.29	Inf	0.000	−13 8 185
				32		**0.000**	**0.000**	**57.89**	**7.18**	**0.000**	**4 −89 125**
				544		**0.000**	**0.000**	**51.37**	**6.78**	**0.000**	**34 −57 590**
						0.000	0.000	48.66	6.61	0.000	34 −68 585
						0.000	0.000	47.36	6.52	0.000	34 −57 505
				150		**0.000**	**0.000**	**49.74**	**6.68**	**0.000**	**−13 −78 100**
						0.000	0.001	45.23	6.38	0.000	−13 −57 100
						0.000	0.002	43.15	6.24	0.000	−8 −78 90
				130		**0.000**	**0.002**	**43.09**	**6.23**	**0.000**	**−26 −78 590**
						0.000	0.002	42.79	6.21	0.000	−26 −78 600
						0.000	0.005	39.63	5.99	0.000	−26 −78 575
				55		**0.001**	**0.028**	**35.32**	**5.66**	**0.000**	**−21 −68 335**
						0.001	0.037	34.48	5.59	0.000	−21 −68 320
						0.002	0.050	33.64	5.52	0.000	−21 −73 305
				7		**0.004**	**0.112**	**31.69**	**5.36**	**0.000**	**−26 −73 505**
				9		**0.006**	**0.154**	**30.88**	**5.29**	**0.000**	**30 −57 335**

table shows 3 local maxima more than 8.0mm apart

Height threshold: F = 26.07, p = 0.000 (0.050)   Degrees of freedom = [1.0, 303.0]
Extent threshold: k = 0 voxels, p = 1.000 (0.050)   FWHM = 15.4 20.0 16.8 mm mm ms; 3.6 3.7 3.4 {voxels}
Expected voxels per cluster, <k> = 1.169   Volume: 11603597 = 101591 voxels = 2034.2 resels
Expected number of clusters, <c> = 0.05   Voxel size: 4.2 5.4 5.0 mm mm ms; (resel = 45.31 voxels)
FWEp: 26.068, FDRp: 33.737   *Page 1*

Figure 19.5: *3D sensor-time SPMF at $p < .05$ FWE corrected for the amplitude difference between face and scrambled face trials. The x, y coordinates refer to position in the 32x32 electrode plane (with units of mm); the z coordinate refers to peristimulus time in ms (to the nearest sampling of 5ms).*
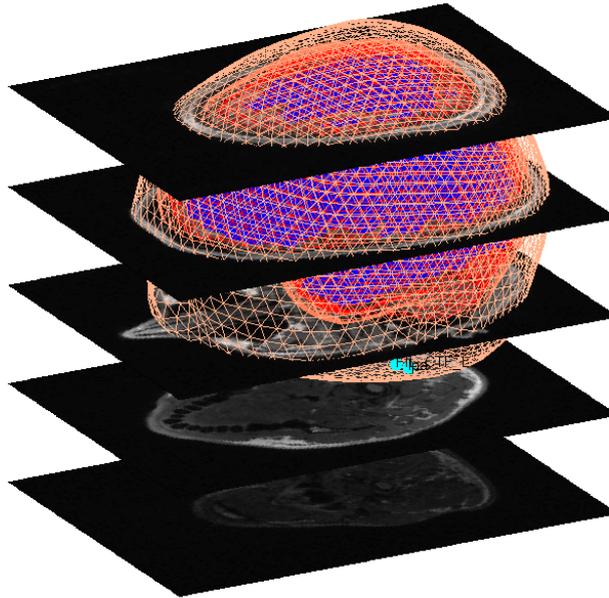
Figure 19.6: *Cortex (blue), inner skull (red), outer skull (orange) and scalp (pink) meshes with transverse slices of the subject's MRI.*

MRI, and the Multiple Sparse Priors (MSP) method in which multiple constraints on the solution can be imposed (Friston et al, 2008, Henson et al, 2009a).

Press the "3D source reconstruction" button, and press the "load" button at the top of the new window. Select the `wmaceMdspm8_faces_run1.mat` file and type a label (eg "N170 MSP") for this analysis[8].

Press the "MRI" button, select the `smri.img` file within the `sMRI` sub-directory, and select "normal" for the cortical mesh.

The "imaging" option corresponds to a distributed source localisation, where current sources are estimated at a large number of fixed points (8196 for a "normal" mesh here) within a cortical mesh, rather than approximated by a small number of equivalent dipoles (the ECD option). The imaging approach is better suited for group analyses and (probably) for later-occuring ERP components. The ECD approach may be better suited for very early sensory components (when only small parts of the brain are active), or for DCMs using a small number of regions (Kiebel et al, 2006).

The first time you use a particular structural image for 3D source reconstruction, it will take some time while the MRI is segmented (and normalisation parameters determined). This will create in the `sMRI` directory the files `y_smri.nii` and `smri_seg8.mat` for normalisation parameters and 4 GIfTI (`.gii`) files defining the cortical mesh, inner skull, outer skull and scalp surface.

When meshing has finished, the cortex (blue), inner skull (red), outer skull (orange) and scalp (pink) meshes will be shown in the Graphics window with slices from the sMRI image, as shown in Figure 19.6. This makes it possible to visually verify that the meshes fit the original image well. The field `D.inv{1}`.mesh field will be updated in MATLAB . Press "save" in top right of window to update the corresponding `mat` file on disk.

Both the cortical mesh and the skull and scalp meshes are not created directly from the segmented MRI, but rather are determined from template meshes in MNI space via inverse spatial normalisation (Mattout et al, 2007).

Press the "Co-register" button. You will first be asked to select at least 3 fiducials from a

---

[8]Note that no new M/EEG files are created during 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).
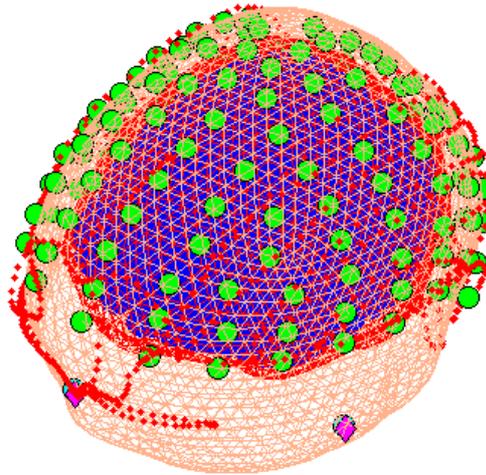
Figure 19.7: *Graphical output of Co-registration of EEG data, showing (upper panel) cortex (blue), inner skull (red) and scalp (black) meshes, electrode locations (green), MRI/Polhemus fiducials (cyan/magneta), and headshape (red dots).*

list of points in the EEG dataset (from Polhemus file): by default, SPM has already highlighted what it thinks are the fiducials, i.e, points labelled "nas" (nasion), "lpa" (left preauricular) and "rpa" (right preauricular). So just press "ok".

You will then be asked for each of the 3 fiducial points to specify its location on the MRI images. This can be done by selecting a corresponding point from a hard-coded list ("select"). These points are inverse transformed for each individual image using the same deformation field that is used to create the meshes. The other two options are typing the MNI coordinates for each point ("type") or clicking on the corresponding point in the image ("click"). Here, we will type coordinates based on where the experimenter defined the fiducials on the `smri.img`. These coordinates can be found in the `smri_fid.txt` file also provided. So press "type" and for "nas", enter [0 91 -28]; for "lpa" press "type" and enter [-72 4 -59]; for "rpa" press "type" and enter [71 -6 -62]. Finally, answer "no" to "Use headshape points?" (in theory, these headshape points could offer better coregistration, but in this dataset, the digitised headshape points do not match the warped scalp surface very well, as noted below, so just the fiducials are used here).

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, a figure like that in Figure 19.7 will appear in the top of the Graphics window, which you can rotate with the mouse (using the Rotate3D MATLAB Menu option) to check all sensors. Finally, press "save" in top right of window to update the corresponding mat file on disk.

Note that for these data, the coregistration is not optimal, with several EEG electrodes appearing inside the scalp. This may be inaccurate Polhemus recording of the headshape or inaccurate surface matching for the scalp mesh, or "slippage" of headpoints across the top of the scalp (which might be reduced in future by digitising features like the nose and ears, and including them in the scalp mesh). This is not actually a problem for the BEM calculated below, however, because the electrodes are re-projected to the scalp surface (as a precaution).

Press "Forward Model", and select "EEG BEM". The first time you do this, there will be a lengthy computation and a large file `smri_EEG_BEM.mat` will be saved in the `sMRI` directory containing the parameters of the boundary element model (BEM). In the Graphics window the BEM meshes will be displayed with the EEG sensors marked with green asterisks as shown (after rotating to a "Y-Z" view using MATLAB rotate tool) in Figure 19.8. This display is the final quality control before the model is used for lead field computation.

Press "Invert", select "Imaging" (i.e, a distributed solution rather than DCM; Kiebel et al
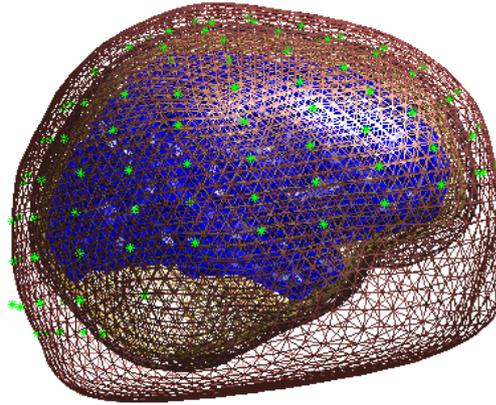
Figure 19.8: *BEM meshes with the EEG sensors marked as asterisks.*

(2006)), select "yes" to include all conditions (i.e, both the differential and common effects of faces and scrambled faces) and then "Standard" to use the default settings.

By default the MSP method will be used. MSP stands for "Multiple Sparse Priors" (Friston et al. 2008a), and has been shown to be superior to standard minimum norm (the alternative IID option) or a maximal smoothness solution (like LORETA; the COH option) - see Henson et al (2009a). Note that by default, MSP uses a "Greedy Search" (GS) (Friston et al, 2008b), though the standard ReML (as used in Henson et al, 2007) can also be selected via the batch tool (this uses Automatic Relevance Determination - ARD).

The "Standard" option uses default values for the MSP approach (to customise some of these parameters, press "Custom" instead).

At the first stage of the inversion lead fields will be computed for all the mesh vertices and saved in the file SPMgainmatrix_wmaceMdspm8_faces_run1_1.mat. Then the actual MSP algorithm will run and the summary of the solution will be displayed in the Graphics window.

Press "save" to save the results. You can now explore the results via the 3D reconstruction window. If you type 180 into the box in the bottom right (corresponding to the time in ms) and press "mip", you should see an output similar to Figure 19.9. This fit explains approx 96% of the data.

Note the hot-spots in bilateral posterior occipitotemporal cortex, bilateral mid-fusiform, and right lateral ventral temporal. The timecourses come from the peak voxel. The red curve 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), type "100"ms for the P100, 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 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).

You can also explore other inversion options, such as COH and IID (available for the "custom" inversion), which you will notice give more superficial solutions (a known problem with standard minimum norm approaches). 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.

In this final section we will concentrate on how to prepare source data for subsequent statistical analysis (eg with data from a group of subjects).

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
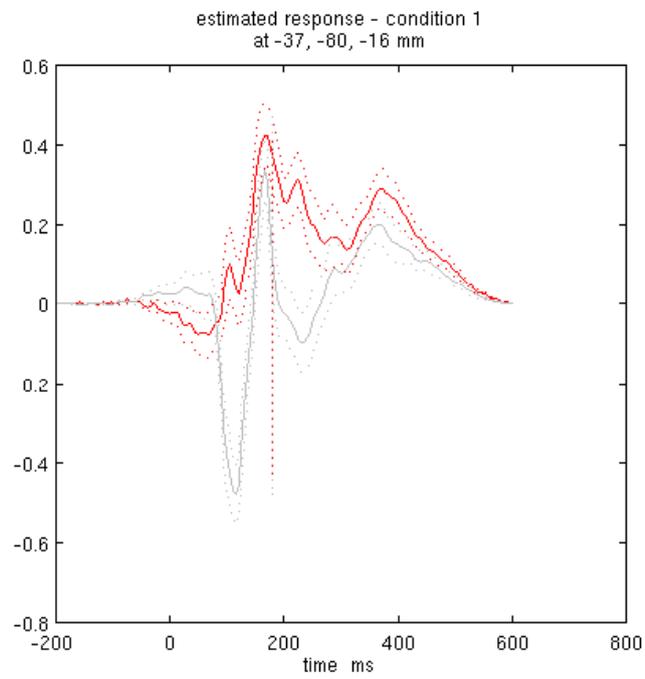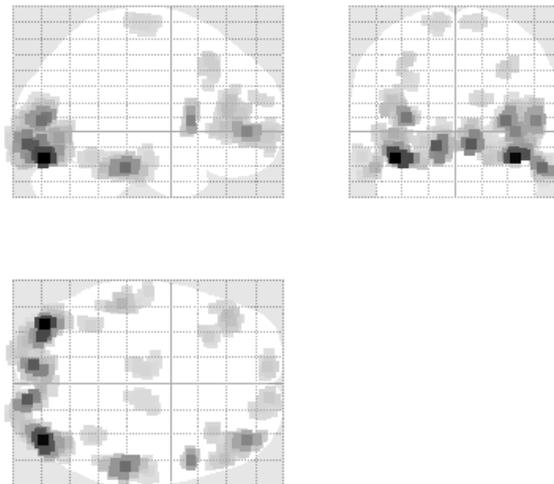
Figure 19.9: *Graphical output of an MSP estimation of the differential ERP between faces and scrambled faces at 180ms.*

window will then show the mean activity for this time/frequency contrast (top plot) and the contrast itself (bottom plot; note additional use of a Hanning window).

Then press "Image" and SPM will write 3D NIfTI images corresponding to the above contrast for each condition:

```
wmaceMdspm8_faces_run1_1_t150_200_f_1.nii
wmaceMdspm8_faces_run1_1_t150_200_f_2.nii
```

The last number in the file name refers to the condition number; the number following the dataset name refers to the reconstruction number (i.e. the number in red in the reconstruction window, i.e, `D.val`, here 1). The reconstruction number will increase if you create a new inversion by pressing "new".

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, see Figure 19.10 (after moving the cursor to the right posterior hotspot), 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 (just the functional image with no structural will be shown), and locate the coordinates of the "hotspots" in MNI space. Note that these images contain RMS (unsigned) source estimates (see Henson et al, 2007). Given that one has data from multiple subjects, one can create a NIFTI file for each. Group statistical analysis can the be implemented with eg. second level t-tests as described earlier in the chapter.
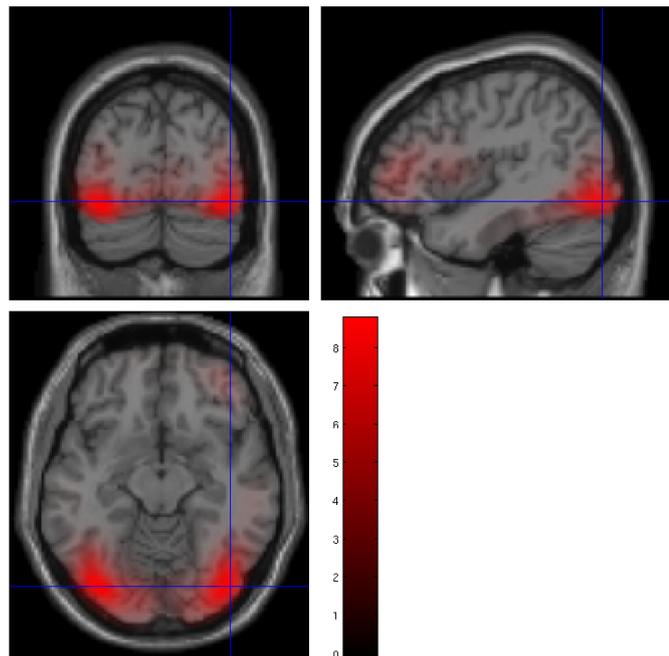
Figure 19.10: *3D reconstruction saved as a smoothed NIfTI image of the differential evoked response for faces vs scrambled faces around the N170.*

19.5   MEG analysis

19.5.1   Preprocessing the MEG data

First change directory to the MEG subdirectory (either in MATLAB or via the "CD" option in the SPM "Utils" menu)

19.5.2 Adjust trigger latency

For the EEG data, the faces were displayed directly via a CRT monitor. For the MEG data on the other hand, the faces were displayed inside the MSR via a projector. This projector produces a delay of 1.5 screen refreshes, which at 60Hz, is 25ms. This means that the subject actually saw the stimuli 25ms after the trigger was sent to the MEG acquisition machine. To correct for this visual delay, we will illustrate how to manipulate "trial" structures[9]. First, we need to read in the triggers from the MEG data (unlike the EEG dataset, the MEG dataset contains information about trial type so we can define the correct condition labels already at this stage). To do this, type the following in the MATLAB window:

```
[trl, conditionlabels, S] = spm_eeg_definetrial;
```

and follow these steps:

- Select the SPM_CTF_MEG_example_faces1_3D.ds/SPM_CTF_MEG_example_faces1_3D.meg4 file.

- Enter -200 for "Start of trial in PST [ms]" and 600 to "End of trial in PST [ms]".

- Enter 2 for "How many conditions?".

  - Enter "faces" for "Label of condition 1". A dialog with a list of events will come up and Select the event with type UPPT001_up and Value 1.

  - Enter "scrambled" for "Label of condition 2". Select the even with type UPPT001_up and Value 2.

- Answer "no" to the question about reviewing trials.

- Answer "yes" to the prompt to save the trial definition.

- Enter a filename like trials_run1.mat and save in the MEG directory.

Then type load trials_run1.mat in MATLAB to see the contents of the file you just saved. It contains two variables, trl and conditionlabels. The trl variable contains as many rows as triggers were found (across all conditions) and three columns: the initial sample of the epoch, the final sample of the epoch and the offset in samples corresponding to a peristimulus time of 0. The sampling rate for the MEG data was 480Hz (as can be found in the S.fsample field of the structure S returned by the spm_eeg_definetrial call above). Thus the figure of -96 samples in the third column corresponds to the 200ms baseline period that you specified. Now we need to shift the initial and final samples of the epochs by 25ms. You can do this by typing:

```
trl(:,1:2) = trl(:,1:2) + round(25*S.fsample/1000);
save trials_run1 trl conditionlabels
```

The new trial definition is thus resaved, and we can use this file when next converting the data.

19.5.3 Convert

Press the CONVERT button, and in the file selection window again select the SPM_CTF_MEG_example_faces1_3D.ds subdirectory and the SPM_CTF_MEG_example_faces1_3D.meg4 file. At the prompt "Define settings?" select "yes". Here we will use the option to define more precisely the part of data that should be read during conversion. Answer "trials" to "How to read?", and "file" to "Where to look for trials?". Then in the file selector window, select the new trials_run1.mat file. Press "no" to "Read across trials?" and select "meg" for "What channels?". Press "no" to avoid saving the channel selection. Press "Enter" to accept the default suggestion for the name of the output dataset. Two files will be generated espm8_SPM_CTF_MEG_example_faces1_3D.mat and espm8_SPM_CTF_MEG_example_faces1_3D.dat. After the conversion is complete the data file will be automatically opened in the SPM8 reviewing tool. If you click on the "MEG" tab you will see the MEG data which is already epoched. By pressing the "intensity rescaling" button (with arrows pointing up and down) several times you will start seeing MEG activity.

---

[9]Alternatively, you could correct by 1 refresh, to match the delay in the EEG data.

19.5.4   Baseline correction

We need to perform baseline correction as it is not done automatically during conversion. This will prevent excessive edge artefacts from appearing after subsequent filtering and downsampling. Select BASELINE CORRECTION from the "Other" drop-down menu and select the espm8_SPM_CTF_MEG_example_face file. Enter [−200 0] for "Start and stop of baseline [ms]". The progress bar will appear and the resulting data will be saved to dataset bespm8_SPM_CTF_MEG_example_faces1_3D.{mat,dat}.

19.5.5   Downsample

Select DOWNSAMPLE from the "Other" drop-down menu and select the bespm8_SPM_CTF_MEG_-example_faces1_3D.mat file. Choose a new sampling rate of 200 (Hz). The progress bar will appear and the resulting data will be saved to dataset dbespm8_SPM_CTF_MEG_example_faces1_3D.{mat,dat}.

19.5.6   Batch preprocessing

Here we will preprocess the second half of the MEG data using using the SPM8 batch facility to demonstrate this third (after interactive GUI and Matlab script) possibility. First though, we have to correct the visual onset latency for the second run, repeating the above steps that you did for the first run:

```
[trl, conditionlabels, S] = spm_eeg_definetrial;
```

and select the SPM_CTF_MEG_example_faces2_3D.ds subdirectory and the SPM_CTF_MEG_example_-faces2_3D.meg4 file. Then enter -200 for "Start of trial in PST [ms]" and 600 to "End of trial in PST [ms]". Enter 2 for "How many conditions?". Enter "faces" for "Label of condition 1". A dialog with a list of events will come up. Select the event with type "UPPT001_up" and Value 1. Enter scrambled for "Label of condition 2". Select the event with type "UPPT001_up" and Value 2. Answer "no" to the question about reviewing trials, but "yes" to the prompt to save the trial definition. Enter a filename like trials_run2.mat and save in the MEG directory. Then type:

```
load trials_run2
trl(:,1:2) = trl(:,1:2)+round(25*S.fsample/1000);
save trials_run2 trl conditionlabels
```

Now press the BATCH button (lower right corner of the SPM8 menu window). The batch tool window will appear. We will define exactly the same settings as we have just done using the interactive GUI. From the "SPM" menu, "M/EEG" submenu select "M/EEG Conversion". Click on "File name" and select the SPM_CTF_MEG_example_faces2_3D.meg4 file from SPM_CTF_-MEG_example_faces2_3D.ds subdirectory. Click on "Reading mode" and switch to "Epoched". Click on "Epoched" and choose "Trial file", double-click on the new "Trial file" branch and then select the trials_run2.mat file. Then click on "Channel selection" and select MEG from the menu below. Finally enter espm8_SPM_CTF_MEG_example_faces2_3D for "Output filename" to be consistent with the file preprocessed interactively.

Now select "M/EEG Baseline correction" from the "SPM" menu, "M/EEG" submenu. Another line will appear in the Module list on the left. Click on it. The baseline correction configuration branch will appear. Select "File name" with a single click. The file that we need to downsample has not been generated yet but we can use the "Dependency" button. A dialog will appear with a list of previous steps (in this case just the conversion) and we can set the output of one of these steps as the input to the present step. Now just enter enter −200 0 for "Baseline". Similarly we can now add "M/EEG Downsampling" to the module list, define the output of baseline correction step for "File name" and 200 for the "New sampling rate". This completes our batch. We can now save it for future use (e.g, as batch_meg_preprocess and run it by pressing the green "Run" button. This will generate all the intermediate datasets and finally dbespm8_SPM_CTF_MEG_example_faces2_3D.{mat,dat}.

19.5.7 Merge

We will now merge the two epoched files we have generated until now and continue working on the merged file. Select the MERGE command from the "Other" drop-down menu. In the selection window that comes up click on `dbespm8_SPM_CTF_MEG_example_faces1_3D.mat` and `dbespm8_SPM_CTF_MEG_example_faces2_3D.mat`. Press "done". Answer "Leave as they are" to "What to do with condition labels?". A new dataset will be generated called `cdbespm8_SPM_CTF_MEG_example_faces1_3D.{mat,dat}`.

19.5.8 Reading and preprocessing data using Fieldtrip

Yet another even more flexible way to pre-process data in SPM8 is to use the Fieldtrip toolbox (http://www.ru.nl/neuroimaging/fieldtrip/) that is distributed with SPM. All the pre-processing steps we have done until now can also be done in Fieldtrip and the result can then be converted to SPM8 dataset. An example script for doing so can be found in the `man\example_scripts\spm_ft_multimodal_preprocessing.m`. The script will generate a merged dataset and save it under the name `ft_SPM_CTF_MEG_example_faces1_3D.{mat,dat}`. The rest of the analysis can then be done as below. This option is more suitable for expert users well familiar with Matlab. Note that in the script `ft_` prefix is added to the names of all Fieldtrip functions. This is a way specific to SPM8 to call the functions from Fieldtrip version included in SPM distribution and located under `external\fieldtrip`.

19.5.9 Prepare

In this section we will add the separately measured headshape points to our merged dataset. This is useful when one wants to improve the coregistration using head shape measured outside the MEG. Also in some cases the anatomical landmarks detectable on the MRI scan and actual locations of MEG locator coils do not coincide and need to be measured in one common coordinate system by an external digitizer (though this is not the case here). First let's examine the contents of the headshape file. If you load it into MATLAB workspace (type `load headshape.mat`), you will see that it contains one MATLAB structure called `shape` with the following fields:

- `.unit` - units of the measurement (optional)

- `.pnt` - Nx3 matrix of headshape points

- `.fid` - substruct with the fields .pnt - Kx3 matrix of points and .label -Kx1 cell array of point labels.

The difference between `shape.pnt` and `shape.fid.pnt` is that the former contains unnamed points (such as continuous headshape measurement) whereas the latter contains labeled points (such as fiducials). Note that this Polhemus space (which will define the "head space") has the X and Y axes switched relative to MNI space.

Now select PREPARE from the "Other" menu and in the file selection window select `cdbespm8_SPM_CTF_MEG_example_faces1_3D.mat`. A menu will appear at the top of SPM interactive window (bottom left window). In the "Sensors" submenu choose "Load MEG Fiducials/Headshape". In the file selection window choose the `headshape.mat` file and save the dataset with `File/Save`.

If you do not have a separately measured headshape and are planning to use the original MEG fiducials for coregistration, this step is not necessary. As an exercise, you can skip it for the tutorial dataset and later do the coregistration without the headshape and see if it affects the results.

19.5.10 Basic ERFs

Press the AVERAGING button and select the `cdbespm8_SPM_CTF_MEG_example_faces1_3D.mat` file. Answer "yes" to "Use robust averaging?". You can either save the weights if you want to examine them or not save if you want the averaging to work faster since the weights dataset that needs to be written is quite large. Answer "no" to "weight by condition" and accept the default
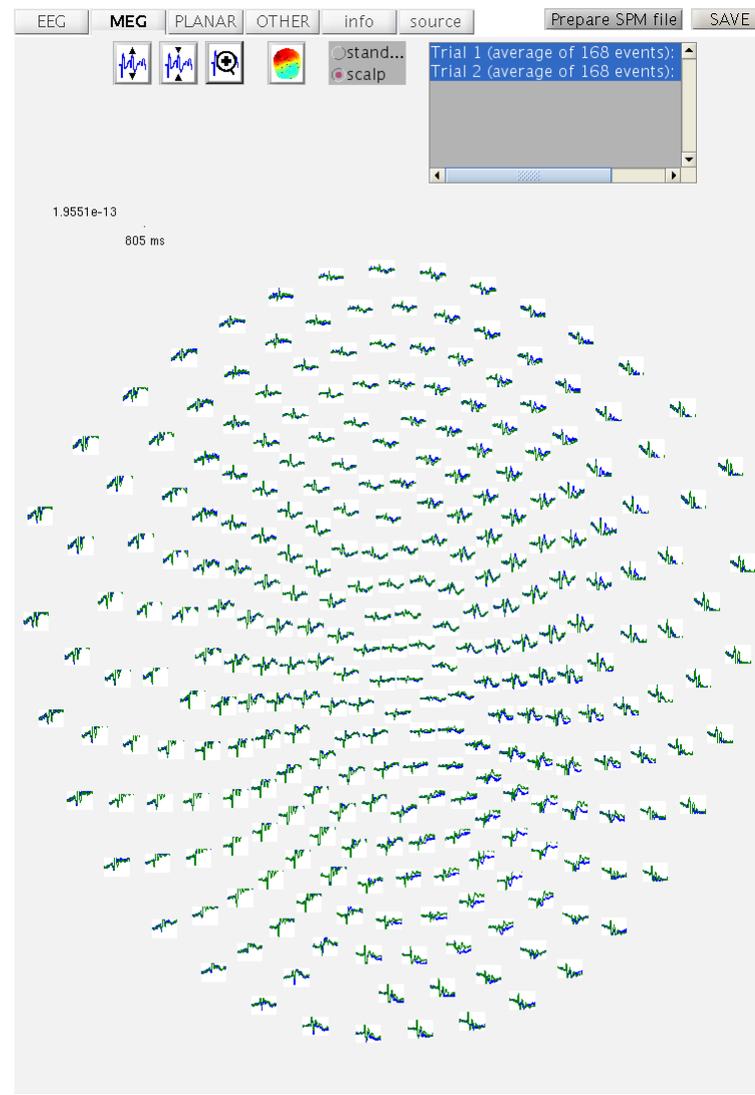
Figure 19.11:   *SPM Display window for mean, smoothed ERF (*`mcdbespm8_SPM_CTF_MEG_-`
`example_faces1_3D.mat`*) for all 275 MEG channels.*

"Offset of the weighting function". A new dataset will be created in the MEG directory called
`mcdbespm8_SPM_CTF_MEG_example_faces1_3D.{mat,dat}` ("m" for "mean").

As before, you can display these data by "Display: M/EEG" and selecting the `mcdbespm8_-`
`SPM_CTF_MEG_example_faces1_3D.mat`. In the MEG tag with the scalp radio button selected,
hold the Shift key and select trial-type 2 with the mouse in the bottom right of the window to
see both conditions superimposed (as Figure 19.11).

Select "Contrast" from the "Other" pulldown menu on the SPM window. This function creates
linear contrasts of ERPs/ERFs. Select the `mcdbespm8_SPM_CTF_MEG_example_faces1_3D.mat` file,
enter $[1 -1]$ as the first contrast and label it "`Difference`", answer "yes" to "Add another", enter
$[1/2\ 1/2]$ as the second contrast and label it "`Mean`". Press "no" to the question "Add another"
and not to "weight by num replications". This will create new file `wmcdbespm8_SPM_CTF_MEG_-`
`example_faces1_3D.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 for faces and scambled faces.

To see the topography of the differential ERF, select "Display: M/EEG", MEG tab and click
on Trial 1, press the "topography" button at the top of the window and scroll to 180ms for the
latency to produce Figure 19.12.
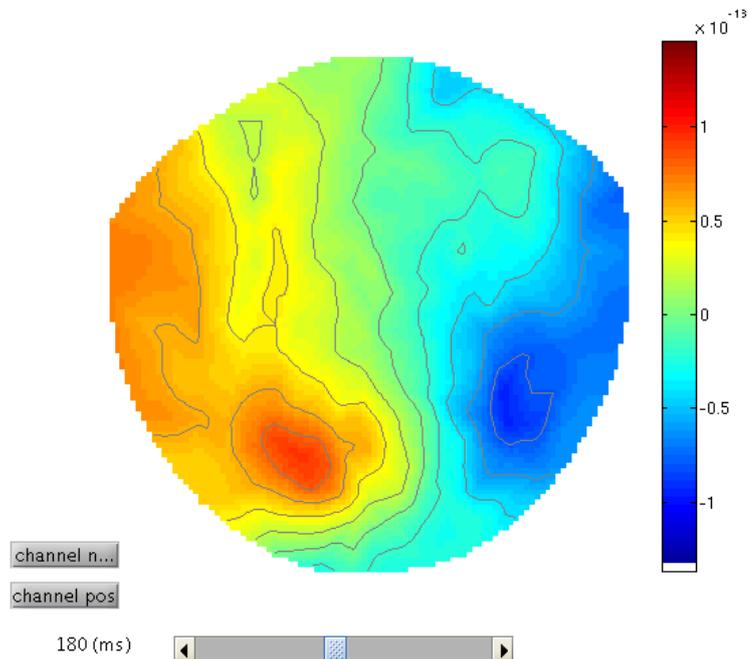
Figure 19.12: *2D topography of the ERF of faces minus scrambled faces at 180ms.*

You can move the slider left and right to see the development of the M170 over time.

19.5.11 Time-Frequency Analysis

SPM can use several methods for time-frequency decomposition. We will use Morlet wavelets for our analyses.

Select the TIME-FREQUENCY option under the "Other" pull-down menu. SPM batch tool with time-frequency configuration options will apear. Double-click on "File name" and select the `cdbespm8_SPM_CTF_MEG_example_faces1_3D.mat` file. Then click on "Channel selection" and in the box below click on "Delete: All(1)" and then on "New: Custom channel". Double-click on "Custom channel" and enter "MLT34".[10] Double-click on "Frequencies of interest" and type [5:40] (Hz). Click on "Spectral estimation" and select "Morlet wavelet transform". Change the number of wavelet cycles from 7 to 5. This factor effectively trades off frequency vs time resolution, with a lower order giving higher temporal resolution. Select "yes" for "Save phase?".

This will produce two new datasets, `tf_cdbespm8_SPM_CTF_MEG_example_faces1_3D.{mat,dat}` and `tph_cdbespm8_SPM_CTF_MEG_example_faces1_3D.{mat,dat}`. The former contains the power at each frequency, time and channel; the latter contains the corresponding phase angles.

Here we will not baseline correc the time-frequency data because for frequencies as low as 5Hz, one would need a longer pre-stimulus baseline, to avoid edge-effects[11]. Later, we will compare two trial-types directly, and hence any pre-stimulus differences will become apparent.

Press the AVERAGING button and select the `tf_cdbespm8_SPM_CTF_MEG_example_faces1_-3D.mat` file. You can use straight (or robust if you prefer) averaging to compute the average time-frequency representation. A new file will be created in the MEG directory called `mtf_cdbespm8_-SPM_CTF_MEG_example_faces1_3D.{mat,dat}`. Note that you can use the reviewing tool to review the time-frequency datasets.

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

---

[10]You can of course obtain time-frequency plots for every channel, but it will take much longer (and result in a large file).

[11]For example, for 5Hz, one would need at least N/2 x 1000ms/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.
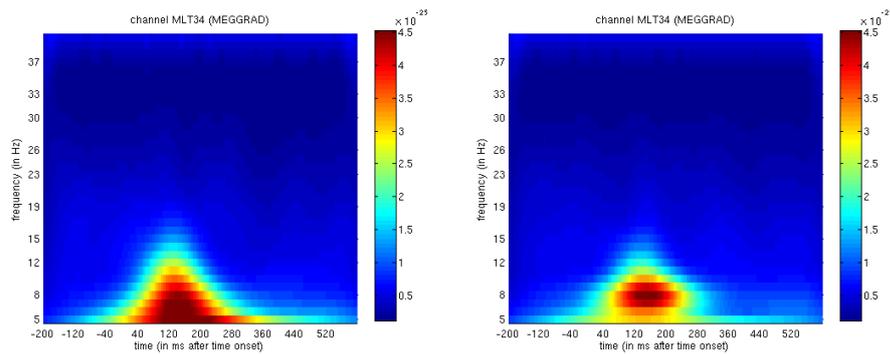
Figure 19.13: *Total power spectra for faces (left) and scrambled faces (right) for channel MLT34*

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 `mcdbespm8_SPM_CTF_MEG_example_-faces1_3D.mat` file).

The power spectra for each trial-type can be displayed using the usual Display button and selecting the `mtf_cdbespm8_SPM_CTF_MEG_example_faces1_3D.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 the magnifying glass icon and on the single channel to get scales for the axes, as in Figure 19.13). 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. We compute the vector mean (by converting the angles to vectors in Argand space), which yelds complex numbers. We can generate two kinds of images from these numbers. The first is an image of the angles, which shows the mean phase of the oscillation (relative to the trial onset) at each time point. The second is an image of the absolute values (also called "Phase-Locking Value", PLV) which lie between 0 for no phase-locking across trials and 1 for perfect phase-locking.

Press the AVERAGING button and select the `tph_cdbespm8_SPM_CTF_MEG_example_faces1_-3D.mat` file. This time you will be prompted for either "angle" or "abs(PLV)" average, for which you should select "abs(PLV)". The MATLAB window will echo:

```
mtph_cdbespm8_SPM_CTF_MEG_example_faces1_3D.mat: Number of replications per contrast:
average faces: 168 trials, average scrambled: 168 trials
```

and a new file will be created in the MEG directory called `mtph_cdbespm8_SPM_CTF_MEG_-example_faces1_3D.mat`.

If you now display the file `mtph_cdbespm8_SPM_CTF_MEG_example_faces1_3D.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 for faces than scrambled faces at lower frequencies, as in Figure 19.14. 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).

19.5.12   2D Time-Frequency SPMs

Analogous to Section 19.4.12, we can also use Random Field Theory to correct for multiple statistical comparisons across the 2-dimensional time-frequency space.

Select CONVERT TO IMAGES in the "Other" pulldown menu, and select the `tf_cdbespm8_-SPM_CTF_MEG_example_faces1_3D.mat` file. Usually you would be asked whether you want to average over channels or frequencies. In this case there is only one channel in this dataset, so the "channels" option will be selected automatically.
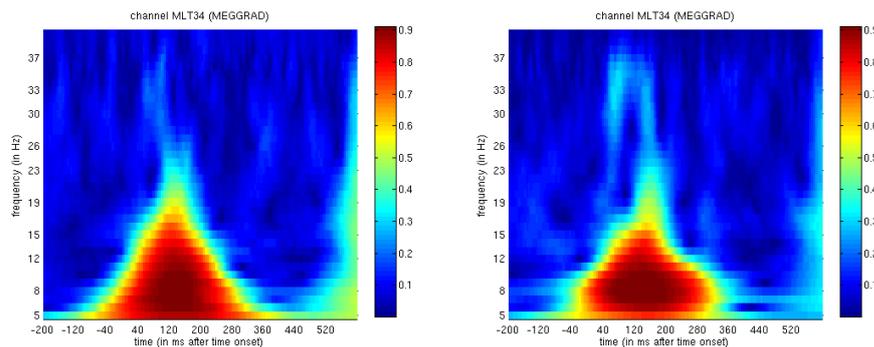
Figure 19.14: *Phase-Locking Values for faces (left) and scrambled faces (right) for channel MLT34*

This will create 2D time-frequency images for each trial of the two types with dimensions $36 \times 161 \times 1$, as for the example shown in Figure 19.15. These images can be found in the subdirectories `1ROI_TF_trialtype_faces` and `1ROI_TF_trialtype_scrambled` of the new directory created `tf_cdbespm8_SPM_CTF_MEG_example_faces1_3D`, and examined by pressing "Display: images" on the main SPM window.

As in Section 19.4.12, these images can be further smoothed in time and frequency if desired.

Then as in Section 19.4.12, 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 `trialtype_faces` (using right click, and "select all") and the images for "group 2" as all those in the subdirectory `trialtype_scrambled`. Finally, specify the new `TFstatsPow` directory as the output directory. (Note that this will be faster if you saved and could load an SPM job file from Section 19.4.12 in order to just change the input files and output directory.) Then add an "Estimate" module from the "SPM" tab, and select the output from the previous factorial design specification stage as the dependency input. Press "Run" (green arrow button).

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 search volume, and then select "Time-Frequency" for the "Data Type". Then press "whole brain", and the Graphics window should now look like that in Figure 19.16.

This will list two "regions" within the 2D time-frequency space in which faces produce greater power than scrambled faces, having corrected for multiple T-tests across pixels. The larger one has a maximum at 5 Hz and 185 ms post-stimulus, corresponding to the M170 seen earlier in the averaged files (but now with a statistical test of its significance, in terms of evoked and induced power). The second, smaller region has a maximum at 12 Hz and 100 ms, possibly corresponding to a smaller but earlier effect on the M100 (also sometimes reported, depending on what faces are contrasted with).

19.5.13 "Imaging" reconstruction of total power for each condition

In Section 19.4.13 we localised the differential evoked potential difference in EEG data corresponding to the N170. Here we will localise the total power of faces, and of scrambled faces, ie including potential induced components (see Friston et al, 2006).

Press the "3D source reconstruction" button, and press the "load" button at the top of the new window. Select the `cdbespm8_SPM_CTF_MEG_example_faces1_3D.mat` file and type a label (eg M170) for this analysis.

Press the "MRI" button, select the `smri.img` file within the `sMRI` sub-directory and select the "normal" mesh.
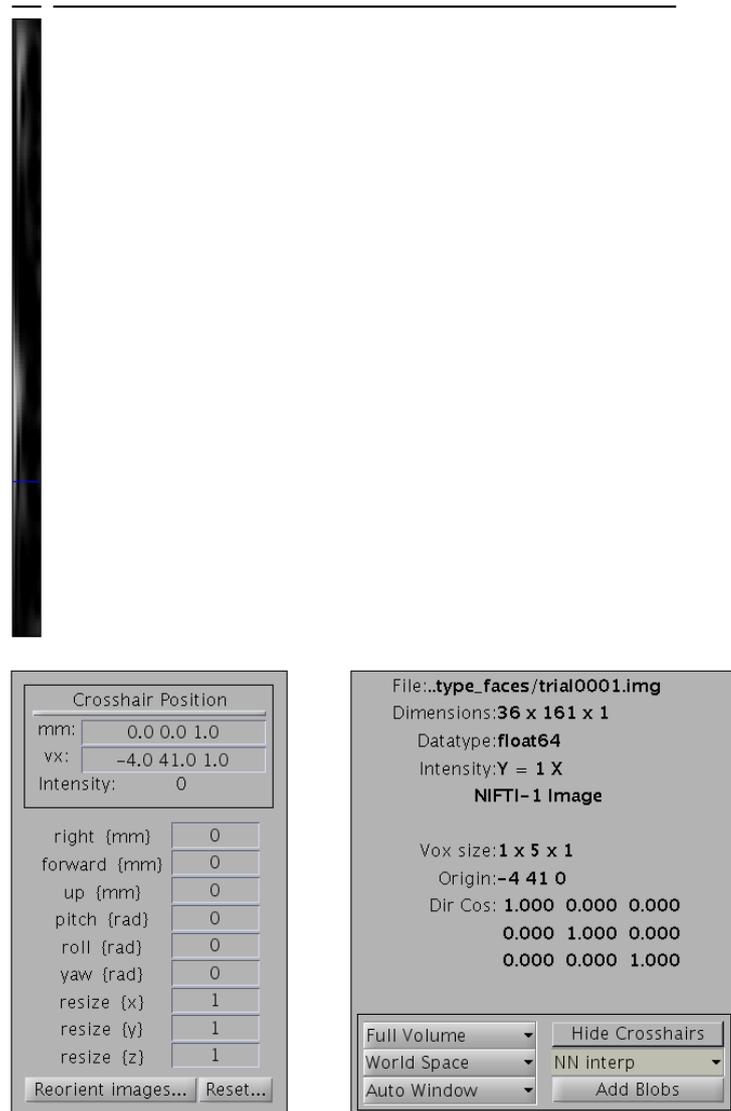
Figure 19.15: *3D image for trial 2 within* **1ROI_TF_trialtype_faces** *subdirectory. The bottom left section is through frequency (x) and time (y) (the other images are strips because there is only one value in the z dimension, i.e, this is really a 2D image).*

Difference

contrast(s)

SPM{T$_{334}$}

SPMresults: ./multimodal/MEG/TFstatsPow
Height threshold T = 3.736016 {p<0.05 (FWE)}
Extent threshold k = 0 voxels

Statistics: *p-values adjusted for search volume*

| set-level		cluster-level				peak-level					Hz ms
p	c	$p_{FWE-corr}$	$q_{FDR-corr}$	k_E	p_{uncorr}	$p_{FWE-corr}$	$q_{FDR-corr}$	T	$(Z_\equiv)$	p_{uncorr}	
0.001	2	0.000	0.006	79	0.003	0.000	0.002	5.40	5.28	0.000	5 185
		0.005	0.092	32	0.092	0.013	0.262	4.12	4.06	0.000	12 100

table shows 3 local maxima more than 8.0mm apart

Height threshold: T = 3.74, p = 0.000 (0.050)	Degrees of freedom = [1.0, 334.0]
Extent threshold: k = 0 voxels, p = 1.000 (0.050)	FWHM = 7.5 58.5 Hz ms ; 7.5 11.7 {voxels}
Expected voxels per cluster, <k> = 13.420	Volume: 28980 = 5796 voxels = 63.7 resels
Expected number of clusters, <c> = 0.05	Voxel size: 1.0 5.0 Hz ms ; (resel = 87.91 voxels)
FWEp: 3.736, FDRp: 5.396, FWEc: 32, FDRc: 79	

Figure 19.16: *2D time-frequency SPMT at $p < .05$ FWE-corrected for the power difference between face and scrambled faces at Channel MLT34.*

If you have not used this MRI image for source reconstruction before, this step will take some time while the MRI is segmented and the deformation parameters computed (see Section 19.4.13 for more details on these files). When meshing has finished, the cortex (blue), inner skull (red) and scalp (orange) meshes will also be shown in the Graphics window with slices from the sMRI image. This makes it possible to verify that the meshes indeed fit the original image well. The field D.inv{1}.mesh will be updated. Press "save" in top right of window to update the corresponding mat file on disk.

Both the cortical mesh and the skull and scalp meshes are not created directly from the segmented MRI, but rather are determined from template meshes in MNI space via inverse spatial normalisation (Mattout et al, 2007; Henson et al, 2009a).

Press the "Co-register" button. You will be asked for each of the 3 fiducial points to specify its location on the MRI images. This can be done by selecting a corresponding point from a hard-coded list ("select"). These points are inverse transformed for each individual image using the same deformation field that is used to create the meshes. The other two options are typing the MNI coordinates for each point ("type") or clicking on the corresponding point in the image ("click"). Here, we will type coordinates based on where the experimenter defined the fiducials on the smri.img. These coordinates can be found in the smri_fid.txt file also provided. So press "type" and for "nas", enter [0 91 -28]; for "lpa" press "type" and enter [-72 4 -59]; for "rpa" press "type" and enter [71 -6 -62]. Finally, answer "no" to "Use headshape points?" (see EEG Section 19.4.13).

This stage coregisters the MEG 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. Press "save" in top right of window to update the corresponding mat file on disk. With the MATLAB Rotation tool on (from the "Tools" tab in the SPM Graphics window, if not already on), right click near the top image and select "Go to X-Z" view. This should produce a figure like that in Figure 19.17, which you can rotate with the mouse to check all sensors. Note that the data are in head space (not MNI space), in
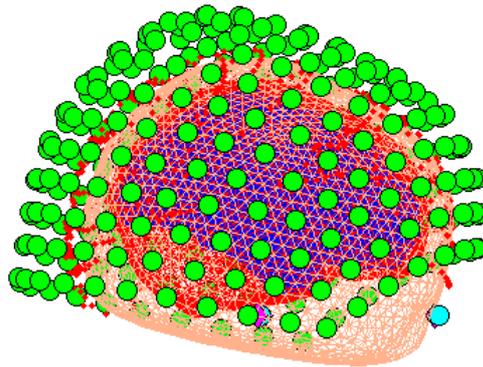
Figure 19.17: *Graphical output of registration of MEG and sMRI data, showing (upper panel) cortex (blue) and scalp (black) meshes, sensor locations (green), MRI and Polhemus fiducials (cyan/magneta), and headshape (red dots).*

this case corresponding to the Polhemus space in which the X and Y dimensions are swapped relative to MNI space.

Press the "Forward Model" button. Choose "MEG Local Spheres" (you may also try the other options; see Henson et al, 2009a). A figure will be displaying showing the local (overlapping) spheres fit to each sensor, and final the set of all spheres.

Press "Invert", select "Imaging", select "yes" to "All conditions or trials?", and "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, 2008, for more details about these parameters). There will be lead field computation followed by the actual inversion. A summary plot of the results will be displayed at the end.

You can now explore the results via the 3D reconstruction window. If you type 180 into the box in the bottom right (corresponding to the time in ms) and press "mip", you should see several ventral temporal hotspots, as in Figure 19.18. Note that this localisation is different from the previous EEG localisation because 1) condition 1 now refers to faces, not the difference between faces and scrambled faces, and 2) the results reflect total power (across trials), induced and evoked, rather than purely evoked[12]. The timecourses come from the peak voxel (with little evidence of a face/scrambled difference for this particular maximum). The red curve shows the condition currently being shown (corresponding to the "Condition 1" toggle bar in the reconstruction window); the grey curve(s) will show all other conditions. If you press the "condition 1" toggle, it will change to Condition 2, which is the total power for 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 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" it will open a graphical interface that is very useful to explore the data (the buttons are fairly self-explanatory).

You can also explore the other inversion options, like COH and IID, which you will note give more superficial solutions (a known problem with standard minimum norm; see also Friston et al, 2008; Henson et al, 2009a). 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.

In the following we will concentrate on how one prepares this single subject data for subsequent entry into a group analysis.

Press the "Window" button in the reconstruction window, enter "150 200" as the timewindow of interest and "5 15" as the frequency band of interest (from the SPM time-frequency analysis, at least from one channel). Then choose the "induced" option. After a delay (as SPM calculates the

---

[12]Though in reality, most of the power is low-frequency and evoked
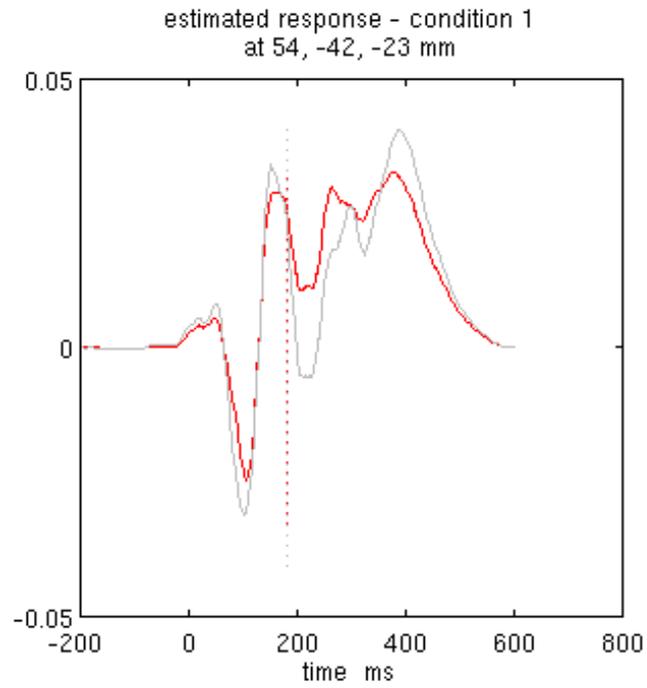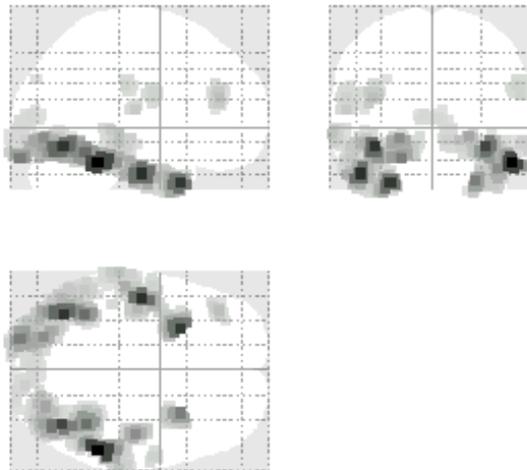
Figure 19.18: *Graphic output for MSP-estimated activity at 159ms for faces.*

power across all trials) 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", SPM will write 3D NIfTI images corresponding to the above contrast for each condition:

```
cdbespm8_SPM_CTF_MEG_example_faces1_3D_1_t150_200_f5_15_1.nii
cdbespm8_SPM_CTF_MEG_example_faces1_3D_1_t150_200_f5_15_2.nii
```

The last number in the file name refers to the condition number; the number following the dataset name refers to the reconstruction number (i.e. 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. 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 `ImCalc` facility to create a difference image of `cdbespm8_SPM_CTF_MEG_example_faces1_3D_1_t150_200_f5_15_1.nii` - `cdbespm8_SPM_CTF_MEG_example_faces1_3D_1_t150_200_f5_15_2.nii`: you should see bilateral fusiform. For further discussion of localising a differential effect (as in Section 19.4.13 with ERPs), vs taking the difference of two localisations, as here, see Henson et al (2007). The above images can then be used at the second level (assuming one also has data from other subjects) to look for effects that are consistent over a group of subjects.

19.6   fMRI analysis

Only the main characteristics of the fMRI analysis are described below; for a more detailed demonstration of fMRI analysis, read previous tutorial chapters describing fMRI analyses.

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)

19.6.1   Preprocessing the fMRI data

Press the BATCH button and then:

- Select SPATIAL: REALIGN: ESTIMATE & RESLICE from the SPM menu, create two sessions, and select the 390 `fM*.img` EPI images within the corresponding Session1 / Session 2 subdirectories (you can use the filter `^fM.*`). In the "Resliced images" option, choose "Mean Image Only".

- Add a SPATIAL: COREG: ESTIMATE module, and select the `smri.img` image in the `sMRI` directory for the "Reference Image" and select a "Dependency" for the "Source Image", which is the Mean image produced by the previous Realign module. For the "Other Images", select a "Dependency" which are the realigned images (two sessions) from Realign.

- Add a SPATIAL: SEGMENT module, and select the `smri.img` image as the "Data".

- Add a SPATIAL: NORMALISE: WRITE module, make a "New: Subject", and for the "Parameter File", select a "Dependency" of the "Norm Params Subj->MNI" (from the prior segmentation module). For the "Images to Write", select a "Dependency" of the "Coreg: Estimate: Coregistered Images" (which will be all the coregistered EPI images) and "Segment: Bias Corr Images" (which will be the bias-corrected structural image). Also, change the "Voxel sizes" to [3 3 3], to save diskspace.

- Add a SPATIAL: SMOOTH module, and for "Images to Smooth", select a "Dependency" of "Normalise: Write: Normalised Images (Subj 1)".

- Save the batch file (e.g, as `batch_fmri_preproc.mat`, and then press the "Run" button.

These steps will take a while, and SPM will output some postscript files with the movement parameters and the coregistration results (see earlier chapters for further explanation). The result will be a series of 2 sets of 390 `swf*.img` files that will be the data for the following 1st-level fMRI timeseries analysis.

19.6.2 Statistical analysis of fMRI data

First make a new directory for the stats output, e.g, a `Stats` subdirectory within the fMRI directory.

Press the BATCH button and then:

- Select "Stats: fMRI model specification" from the SPM module menu, and select the new `Stats` subdirectory as the "Directory".

- Select "Scans" for "Units of design".

- Enter 2 for the "Interscan interval" (i.e, a 2s TR).

- Create a new session from the "Data & Design" menu. For "Scans", select all the `swf*.img` files from the `Session1` subdirectory (except the mean). Under "Multiple Conditions", click "Select File", and select the `trials_ses1.mat` file that is provided with these data. (This file just contains the onsets, durations and names of every event within each session.). For "Multiple regressors", click "Select File", and select the `rp*.txt` file that is also in the `Session1` subdirectory (created during realignment).

- Repeat the above steps for the second session.

- Under "Basis Functions", "Canonical HRF" add the "Time and Dispersion" derivatives.

- Then add a "Stats: Model estimation" module, and for the "Select SPM.mat", choose the "Dependency" of the `SPM.mat` file from the previous "fMRI model specification" module.

- Add a "Stats: Contrast Manager" module, and for the "Select SPM.mat", choose the "Dependency" of the `SPM.mat` file from the previous "Model Estimation module".

- Under "Contrast Sessions", create a new F-contrast with a "Name" like `faces vs scrambled (all BFs)` and then enter [`eye(3) -eye(3) zeros(3,6)`]. This will produce a 3x12 matrix that picks out the three basis functions per condition (each as a separate row), summing across the two conditions (with zeros for the movement parameter regressors, which are of no interest). Then select "Replicate (average over sessions)".

- Under "Contrast Sessions", create a new F-contrast with a "Name" like `faces + scrambled vs Baseline (all BFs)` and then enter the MATLAB [`eye(3) eye(3) zeros(3,6)`]. Again, select "Replicate (average over sessions)".

- Save the batch file (e.g, as `batch_fmri_stats.mat`, and then press the "Run" button.

When this has finished, press RESULTS and select the `SPM.mat` file that should have been created in the new `Stats` directory. The Contrast Manager window will appear, and you can select the "faces vs scrambled (all BFs)" contrast. Do not mask, keep the title, threshold at $p < .05$ FWE corrected, use an extent threshold of 60 voxels, and you should get the MIP and table of values (once you have pressed "whole brain") like that in Figure 19.19. This shows clusters in bilateral midfusiform (FFA), right occipital (OFA), right superior temporal gyrus/sulcus (STS), in addition to posterior cingulate and anterior medial prefrontal cortex. These clusters show a reliable difference in the evoked BOLD response to faces compared with scrambled faces that can be captured within the "signal space" spanned by the canonical HRF and its temporal and dispersion derivatives. Note that this F-contrast can include regions that show both increased and decreased amplitude of the fitted HRF for faces relative to scrambled faces (though if you plot the "faces vs scrambled" contrast estimates, you will see that the leftmost bar (canonical

HRF) is positive for all the clusters, suggesting greater neural activity for faces than scrambled faces (also apparent if you plot the event-related responses)).

There is some agreement between these fMRI effects and the localisation of the differential ERP for faces vs scrambled faces in the EEG data (see earlier section). Note however that the fMRI BOLD response reflects the integrated neural activity over several seconds, so some of the BOLD differences could arise from neural differences outside the 0-600ms epoch localised in the EEG data (there are of course other reasons too for differences in the two localisations; see, eg, Henson et al, under revision).

You can also press RESULTS and select the "faces + scrambled vs Baseline (all BFs)" contrast. Using the same threshold of $p < .05$ FWE corrected, you should see a large swathe of activity over most of the occipital, parietal and motor cortices, reflecting the general visuomotor requirements of the task (relative to interstimulus fixation). The more posterior ventral occipital/temporal BOLD responses are consistent with the MEG localisation of faces (or scrambled faces) versus baseline, though note that the more anterior ventral temporal activity in the MEG localisation is not present in the fMRI data, which suggests (but does not imply) that the MEG localisation may be erroneous.

These contrasts of fMRI data can now be used as spatial priors to aid the localisation of EEG and/or MEG data, as in the next section.

19.7   Multimodal fusion

Here, we will illustrate here two types of multimodal integration:

1. "Fusion" of the EEG and MEG data (Henson et al, 2009b),

2. Use of the fMRI data as spatial priors on the MEG/EEG data (Henson et al, in press).

19.7.1   EEG and MEG fusion

Make a new directory called "Fused', and change into it.

Merging the EEG and MEG datafiles

The first step is to combine the MEG and EEG data into a single SPM file. We will use the (weighted) averaged files for each modality.

Press "Fuse" from the "Others" pulldown menu, and select the `wmcdbespm8_SPM_CTF_MEG_-example_faces1_3D.mat` in the MEG directory and the `wmaceMdspm8_faces_run1.mat` in the EEG directory. This will create a new file called `uwmcdbespm8_SPM_CTF_MEG_example_faces1_-3D.mat` in the new Fused directory. Note that the two files need to have the same number of trials, conditions, samples, etc. Display the new file, and you will see the EEG and MEG data within their respective tabs.

We have to do one extra bit of "preparation" for the EEG data. Because in general, one might want to merge more than one EEG file, integrating all their locations could be tricky. So the simple answer is to clear all locations and force the user to re-specify them. So (as in earlier EEG section), select PREPARE from the "Other" menu and select `uwmcdbespm8_SPM_CTF_MEG_example_faces1_-3D.mat`. Then in the SPM Interactive window, on the "Sensors" submenu, choose "Load EEG sensors"/"Convert locations file", and select the `electrode_locations_and_headshape.sfp` file (in the original EEG directory). Then from the "2D projection" submenu select "Project 3D (EEG)". A 2D channel layout will appear in the Graphics window. Select "Apply" from "2D Projection" and "Save" from "File" submenu.

3D fused "imaging" reconstruction

Now we can demonstrate simultaneous reconstruction of the MEG and EEG data, as described in Henson et al (2009b). This essentially involves scaling each type of data and gain matrix, concatenating them, and inverting using the normal methods, but with separate sensor error covariance terms for each modality.

faces vs scrambled faces (all BFs) - All Sessions

SPMmip
[0, -1, 1]

SPM$\{F_{3,730}\}$

SPMresults:/data/mmfaces/fMRI/Stats
Height threshold F = 10.409554 {p<0.05 (FWE)}
Extent threshold k = 60 voxels

contrast(s)

Statistics: *p-values adjusted for search volume*

set-level		cluster-level				peak-level					mm mm mm
p	c	$p_{FWE-corr}$	$q_{FDR-corr}$	k_E	p_{uncorr}	$p_{FWE-corr}$	$q_{FDR-corr}$	F	$(Z_\equiv)$	p_{uncorr}	
	6			162		0.000	0.000	60.44	Inf	0.000	42 −46 −23
				75		0.000	0.000	33.90	Inf	0.000	0 −49 22
				130		0.000	0.000	32.45	Inf	0.000	−6 53 −14
				70		0.000	0.000	31.93	Inf	0.000	−39 −46 −23
						0.000	0.000	22.48	7.41	0.000	−39 −58 −20
				64		0.000	0.000	29.93	Inf	0.000	42 −85 −8
				62		0.000	0.000	25.22	Inf	0.000	45 −64 16
						0.000	0.001	16.99	6.34	0.000	60 −64 16

table shows 3 local maxima more than 8.0mm apart

Height threshold: F = 10.41, p = 0.000 (0.050)   Degrees of freedom = [3.0, 730.0]
Extent threshold: k = 60 voxels, p = 0   FWHM = 11.5 11.4 11.3 mm mm mm; 3.8 3.8 3.8 {voxels}
Expected voxels per cluster, <k> = 1.320   Volume: 1746360 = 64680 voxels = 1064.5 resels
Expected number of clusters, <c> =   Voxel size: 3.0 3.0 3.0 mm mm mm; (resel = 54.81 voxels)
FWEp: 10.410, FDRp: 14.022

Figure 19.19: *SPM{F} for faces vs scrambled faces.*

- Press the "3D source reconstruction" button, and load the `uwmcdbespm8_SPM_CTF_MEG_-example_faces1_3D.mat` file. Type a label (eg "N/M170").

- Press the "MRI" button, select the `smri.img` file within the `sMRI` sub-directory, and select "normal" for the cortical mesh. Because this MRI was normalised previously, this step should not take long, finishing with display of the cortex (blue), inner skull (red), outer skull (orange) and scalp (pink) meshes.

- Press the "Co-register" button. Press "type" and for "nas", enter [0 91 -28] for "lpa" press "type" and enter [-72 4 -59] for "rpa" press "type" and enter [71 -6 -62]. Finally, answer "no" to "Use headshape points?". Then select either "EEG" or "MEG" to display corresponding data registration. Note that the MEG data will have been coregistered to the EEG data in the headspace. If you want to display the other modality afterwards, just press the "display" button below the "Co-register" button.

- Press "Forward Model", and select "EEG BEM" for the EEG and "Local Spheres" for the MEG. Then select either "EEG" or "MEG" to display corresponding forward model. (If you want to display the other modality afterwards, just press the "display" button below the "Forward Model" button). In the Graphics window the meshes will be displayed with the sensors marked with green asterisks.

- Press "save" to save progress.

- Press "Invert", select "Imaging". Because the fMRI data (see below) already come from a contrast of faces versus scrambled faces, we will only invert the differential ERP/ERFs. So press "no" to the question about invert "all conditions or trials", press "yes" to invert the Difference (between faces and scrambled) but "no" to invert the Mean (of faces and scrambled versus baseline).

  Then press "Standard" to use the default inversion settings (MSP), and then to select both the "EEG" and "MEG" modalities in the new "Select modalities" window, in order to fuse them (simultaneously invert both).

  Lead fields will first be computed for all the mesh vertices and saved in the file `SPMgainmatrix_uwmcdbespm8`. This will take some time. Then the actual MSP algorithm will run and the summary of the solution will be displayed in the Graphics window.

- Press "save" to save the results. You can now explore the results via the 3D reconstruction window. If you type 180 into the box in the bottom right (corresponding to the time in ms) and press "mip", you should see an output similar to Figure 19.20.

Note that because we have inverted only the differential ERP/ERF, these results cannot be compared directly to the unimodal inversions in the previous sections of this chapter. For a fairer comparison:

- Press the "new" button and type "N170" as the label, press "Invert" again (note that all forward models are copied by default from the first inversion) and select the same options as above, except that when asked "Select modalities", select just "EEG". This should produce the results in Figure 19.21. Notice the more posterior maxima.

- Press the "new" button and type "M170" as the label, press "Invert" again and select the same options as above, except that when asked "Select modalities", select just "MEG" this time. This should produce the results in Figure 19.22. Notice the more anterior and medial activity.

By comparing these figures, you can see that the multimodal fused inversion (first inversion) combines aspects of both the unimodal inversions. Unfortunately one cannot simply compare the multi-modal vs uni-modal reconstructions via the log-evidence, because the data differs in each case (rather, one could use an estimate of the conditional precision of the sources, as in Henson et al, 2009b). With multiple subjects though, one could compare statistics across subjects using either multimodal or unimodal inversions.
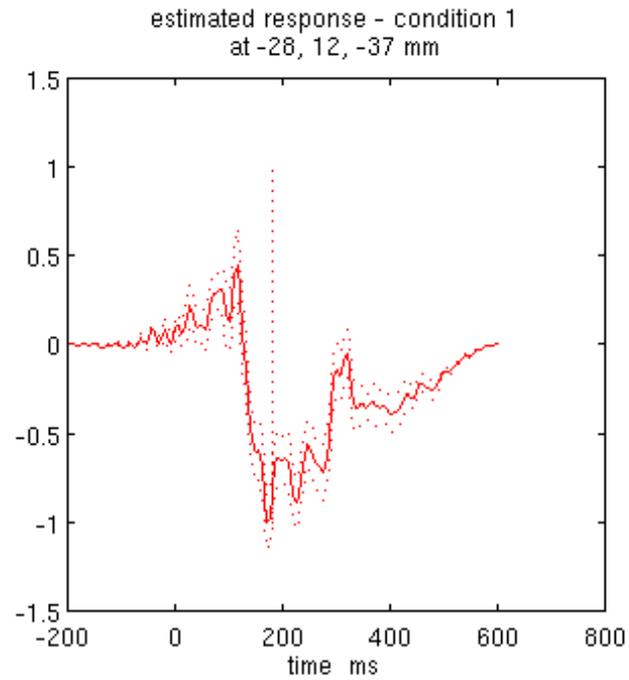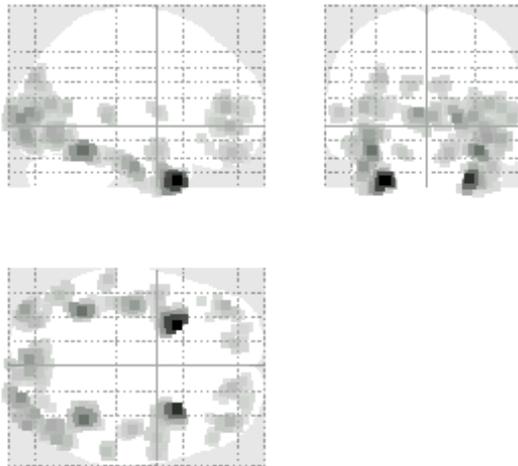
Figure 19.20: *Graphical output of an MSP estimation of the differential evoked response between faces and scrambled faces at 180ms, after fusing both EEG and MEG data.*
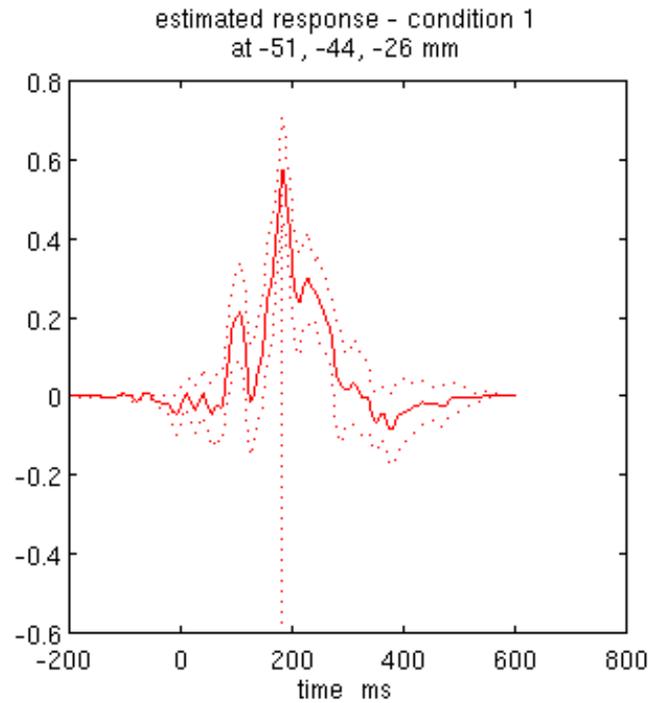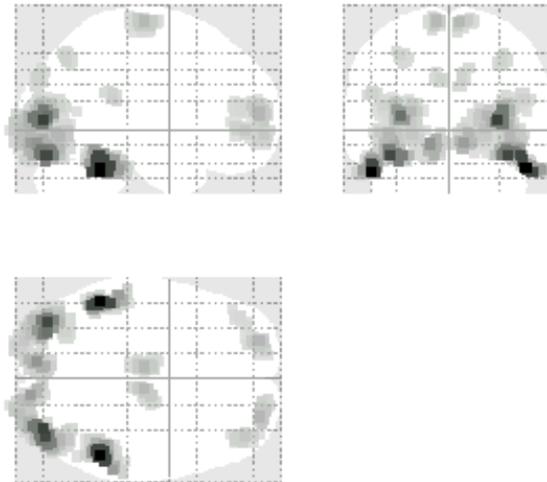
Figure 19.21: *Graphical output of an MSP estimation of the differential evoked response between faces and scrambled faces at 180ms, after inverting just EEG data.*
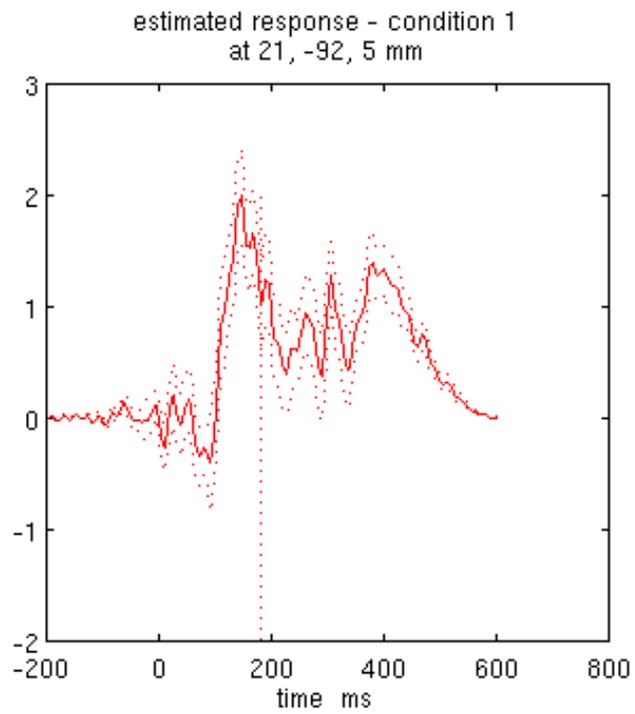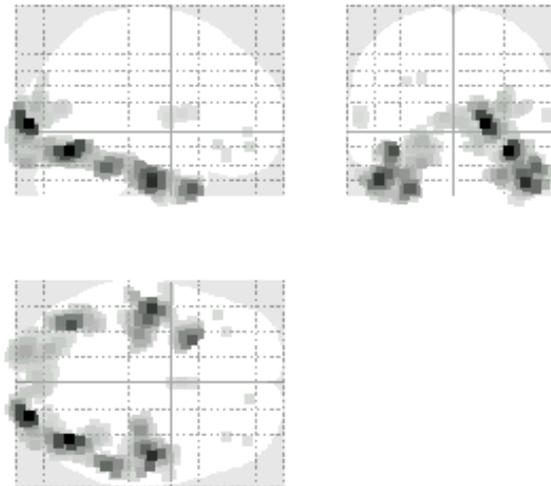
Figure 19.22: *Graphical output of an MSP estimation of the differential evoked response between faces and scrambled faces at 180ms, after inverting just MEG data.*

19.7.2   EEG, MEG and fMRI fusion

Now we can examine the effects of using the fMRI data in Section 19.6 as spatial priors on the sources (Henson et al, in press). But first we need to create an 3D volumetric image of the clusters that we wish to use as spatial priors. These clusters can be defined by thresholding an SPM for a given fMRI contrast: here we will use the contrast in Section 19.6 of faces versus scrambled faces (using all three basis functions). So press "Results" and select the "SPM.mat" file from the "fMRI/Stats" directory. Select the previous faces vs scrambled F-contrast, do not mask or change title, use FWE correction at 0.05 and a 60-voxel extent to reproduce the SPM{F} shown in Figure 19.19 (if you are still in SPM's "EEG" mode, you will also be asked the type of images, for which you reply "Volumetric 2D/3D").

Now press the "save" button in the Interactive window and enter a filename like `FacesVs-Scrambled_FWE05_60`. This will produce a 3D image (which you can display as normal) in which all subthreshold voxels are set to zero (ie, where only 6 clusters containing non-zero voxel values are left).

Now we can use this cluster image in a new inversion:

- Press the "new" button to create a fourth inversion, and type "N/M170+fMRI" as the label.

- Press "Invert", select "Imaging", press "no" for "all conditions or trials", and select only the Difference (not Mean), as before ...

- ... but this time, press "Custom" rather than "Standard" to get more flexibility in the inversion settings. Select "GS" for the type of inversion (the default MSP with a Greedy Search), enter default time window of "-200 to 600", "yes" to a Hanning window, "0" for the highpass and "48" for the lowpass, and then press "yes" to the question of "Source priors?"...

- ... select the `FacesVsScrambled_FWE05_60.img` file in the "fMRI/Stats" directory, and select "MNI" for the "Image space" (because the fMRI images were spatially normalised).

- Say "No" to "Restrict solutions", and then select both the "EEG" and "MEG" modalities in the "Select modalities" window, in order to fuse them (together with the fMRI priors).

  Note that, once the inversion has finished, a new image will be created (in the "fMRI/Stats" directory) called `cluster_FacesVsScrambled_FWE05_60.img`, which contains the six binary priors, as will a GIfTI version called `priors_uwmcdbespm8_SPM_CTF_MEG_example_faces1_-3D_4.func.gii`. If you want to display each spatial prior on the cortical mesh, first make sure you save the current reconstruction, and then type in the MATLAB window:

  ```
  D = spm_eeg_load('uwmcdbespm8_SPM_CTF_MEG_example_faces1_3D.mat');
  val = 4;    % Fourth inversion; assuming you have followed the above steps
  G = gifti(D.inv{val}.mesh.tess_mni);
  P = gifti(D.inv{val}.inverse.fmri.texture);
  for i=1:size(P.cdata,2);
    figure, plot(G,P,i);
  end
  ```

  Finally, a new MATLAB file called `priors_uwmcdbespm8_SPM_CTF_MEG_example_faces1_3D_-4.mat` will also be saved to the current directory, which contains the information necessary to construct the covariance components for subsequent inversion. So if you want to use these fMRI priors in another inversion, next time you are prompted for the "Source priors?", rather than selecting an image ("img" file) as we did above, you can select this "mat" file, so SPM will not need to recreate the covariance matrices from the image file, but can use the covariance matrices directly.

- Again, type 180 into the box in the bottom right and press "mip". This should produce the output in Figure 19.23. Notice how the more posterior midfusion clusters (particularly on the left) have become stronger (where there were fMRI priors). (Note also that fMRI
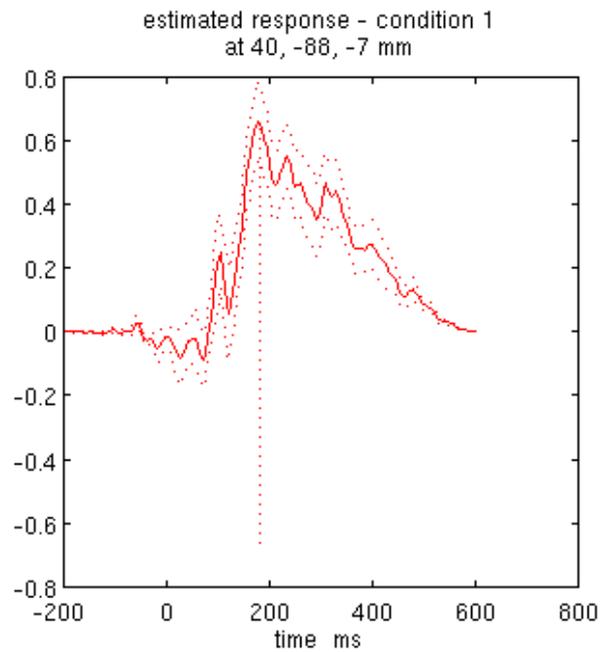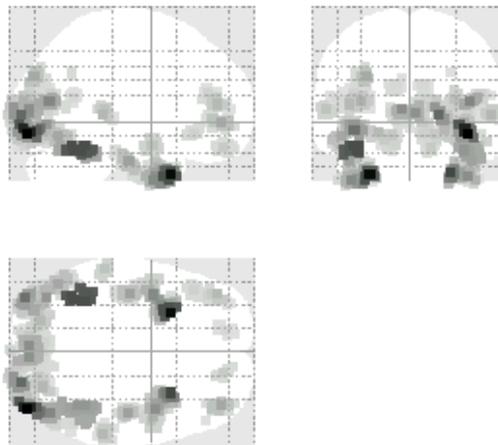
Figure 19.23: *Graphical output of an MSP estimation of the differential evoked response between faces and scrambled faces at 180ms, after fusing EEG, MEG and fMRI data.*

priors have generally been found to have a greater effect on IID or COH inversions, given the implicit flexibility of MSP priors, Henson et al, in press).

- Press "save".

You can repeat the above steps to use the common fMRI effect of faces and scrambled faces versus baseline (though at a higher threshold perhaps) as an alternative set of spatial priors for inverting either the differential evoked MEG/EEG response, or the mean evoked MEG/EEG response vs baseline.

19.8 References

1. Friston, K, Daunizeau, J, Kiebel, S, Phillips, C, Trujillo-Barreto, N, Henson, R, Flandin, G, Mattout, J (2008). Multiple sparse priors for the M/EEG inverse problem. Neuroimage, 39(3):1104-20.

2. Friston, K, Carlton Chu, Janaina Mouro-Miranda, Oliver Hulme, Geraint Rees, Will Penny and John Ashburner (2008). Bayesian decoding of brain images. NeuroImage, 39(1):181-205.

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, 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. Henson, R, Mattout, J, Phillips, C and Friston, K.J. (2009a). Selecting forward models for MEG source-reconstruction using model-evidence. Neuroimage, 46, 168-176.

9. Henson, R, Mouchlianitis, E and Friston, K.J. (2009b). MEG and EEG data fusion: Simultaneous localisation of face-evoked responses. Neuroimage, 47, 581-589.

10. Henson, R, Flandin, G, Friston, K.J. and Mattout, J. (in press). A Parametric Empirical Bayesian framework for fMRI-constrained MEG/EEG source reconstruction. Human Brain Mapping.

11. Kilner, J., Kiebel, S and Friston, K. J. (2005). Applications of random field theory to electrophysiology. Neuroscience Letters, 374:174-178.

12. Kilner, J. and Penny. W. (2006). Robust Averaging for EEG/MEG. HBM06 Abstract.

13. Kiebel S and Friston K (2004). Statistical Parametric Mapping for Event-Related Potentials II: A Hierarchical Temporal Model. NeuroImage, 22, 503-520.

14. 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.

15. Mattout J, Pelegrini-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.

16. Mattout, J, Phillips, C, Penny, W, Rugg, M and Friston, KJ (2005b). MEG source localisation under multiple constraints: an extended Bayesian framework. NeuroImage.

17. Mattout, J., Henson, R N. and Friston, K.J. (in press). Canonical Source Reconstruction for MEG. Computational Intelligence and Neuroscience.

18. Phillips, C, M.D. Rugg, M and Friston, K.J. (2002). Systematic Regularization of Linear Inverse Solutions of the EEG Source Localisation Problem. NeuroImage, 17, 287-301.

19. 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.

20. Wager, TD, Keller, MC, Lacey SC and Jonides J. (2005). Increased sensitivity in neuroimaging analyses using robust regression. NeuroImage, 26(1), 99-113.

Chapter 20

DCM for Induced Responses

This chapter shows an example of Dynamic Causal Modelling for Induced Responses (DCM-IR) [10]. The example is based on the analysis described by Chen et al. [9]. The purpose of the analysis is to look at the effective connectivity between cortical areas involved in the processing of faces and specifically at non-linearities in the connections expressed as cross-frequency coupling. DCM-IR is an example of a phenomenological rather than physiological DCM. The advantage of this approach is that it can directly model a particular feature extracted from the data, namely event-related spectral perturbations. This feature has been a popular object of study in the neuroscience literature. However, since computing event-related power involves discarding phase information, it is not possible to model this feaure with a physiologically realistic model such as the one used in DCM for evoked responses. An important feature of DCM for induced responses is that it models the full time-frequency spectrum. This differs from typical approaches, where a few specific frequency bands are selected *a priori*. DCM-IR models spectral dynamics in terms of a mixture of frequency modes (obtained with singular value decomposition). The dynamics of each mode are encoded by the evolution of a state. It is this multi-state vector, for each source, that captures how the energy in different frequencies interacts, either linearly or non-linearly, among sources.

20.1 Data

We will use the epoched and merged MEG dataset from Chapter 19.5 saved in the files:

```
cdbespm8_SPM_CTF_MEG_example_faces1_3D.mat
cdbespm8_SPM_CTF_MEG_example_faces1_3D.dat
```

See 19.5.1 for the instructions for how to generate these files from raw MEG data. DCM-IR also requires a head model and coregistration. If you have performed "Imaging" reconstruction of differential power 19.5.13 and saved the results, the head model should already be defined. Otherwise, you will be asked to define the head model while configuring the DCM (see below).

20.2 Getting Started

You need to start SPM and toggle "EEG" as the modality (bottom-right of SPM main window), or start SPM with `spm eeg`. In order for this to work you need to ensure that the main SPM directory is on your MATLAB path.

20.3 Setting up DCM

After calling `spm eeg`, you see SPM's graphical user interface, the top-left window. The button for calling the DCM-GUI is found in the second partition from the top, on the right hand side. When pressing the button, the GUI pops up (Figure 20.1). The GUI is partitioned into five parts, going from the top to the bottom. The first part is about loading and saving existing DCMs, and

selecting the type of model. The second part is about selecting data, the third is for specification of the spatial forward model, the fourth is for specifying connectivity, and the last row of buttons allows you to estimate parameters and view results.

You have to select the data first and specify the model in a fixed order (data selection > spatial model > connectivity model). This order is necessary, because there are dependencies among the three parts that would be hard to resolve if the input could be entered in any order. At any time, you can switch back and forth from one part to the next. Also, within each part, you can specify information in any order you like.

20.3.1   load, save, select model type

At the top of the GUI, you can load an existing DCM or save the one you are currently working on. In general, you can `save` and `load` during model specification at any time. You can also switch between different DCM analyses (the left menu). The default is "ERP" which is DCM for evoked responses. You should switch to "IND" which is the option for DCM-IR. The menu on the right-hand side lets you choose the neuronal model. Once you switch to "IND", it will be disabled since neuronal models are not relevant for DCM-IR, which is a phenomenological DCM.

20.3.2   Data and design

In this part, you select the data and model between-trial effects. Press "new data" and select the data file `cdbespm8_SPM_CTF_MEG_example_faces1_3D.mat`.The data file will usually be an epoched file with multiple trials per condition. These data must be in SPM-format. On the right-hand side you can enter trial indices of the evoked responses in this SPM-file. For example, if you want to model the first and second condition contained within an SPM-file, specify indices `1` and `2`. You can type:

```
D = spm_eeg_load('cdbespm8_SPM_CTF_MEG_example_faces1_3D.mat');D.condlist
```

in the command line to see the list of condition labels in the order that corresponds to these indices. This order is defined in the dataset and can be modified by selecting "Sort Conditions" from the "Other" submenu in main SPM window (*spm_eeg_sort_conditions*). SPM should echo:

```
ans =

    'faces'     'scrambled'
```

meaning that index `1` corresponds to presentation of faces and index `2` - to presentation of scrambled faces. The box below the list of indices allows specifying experimental effects on connectivity. The specification can be quite generic as in the design matrix for a General Linear Model (GLM). Our case is quite simple though. We have a baseline condition which is "scrambled" and we would like to know how the condition of interest "faces" differs from it. We will therefore enter:

```
1 0
```

in first row of the box, which means that there will be some additive modulation of connections that we will define later for "faces" (some coefficient multiplied by 1) and this modulation will not be there for "scrambled" (the same coefficient multiplied by 0). If we now click somewhere outside the box, a default name will be assigned to this effect - "effect1". It will appear in the small text box next to the coefficients box. It is possible to change this name to something else e.g. "face".

Now we can select the peristimulus time window we want to model. These are the two test boxes in the top left corner of the panel. Enter `-50` in the left box and `300` in the right box to select the segment `-50` to `300` ms relative to the presentation of the visual stimulus.

You can choose whether you want to remove low-frequency drifts of the data at sensor level. If you don't, select `1` for "detrend", to just remove the mean. Otherwise select the number of discrete cosine transform terms you want to remove. You can also subsample your data (prior to computing the time-frequency decomposition) using the "subsample" option. In general, it is advised to filter out drifts and downsample the data during preprocessing. The options here are
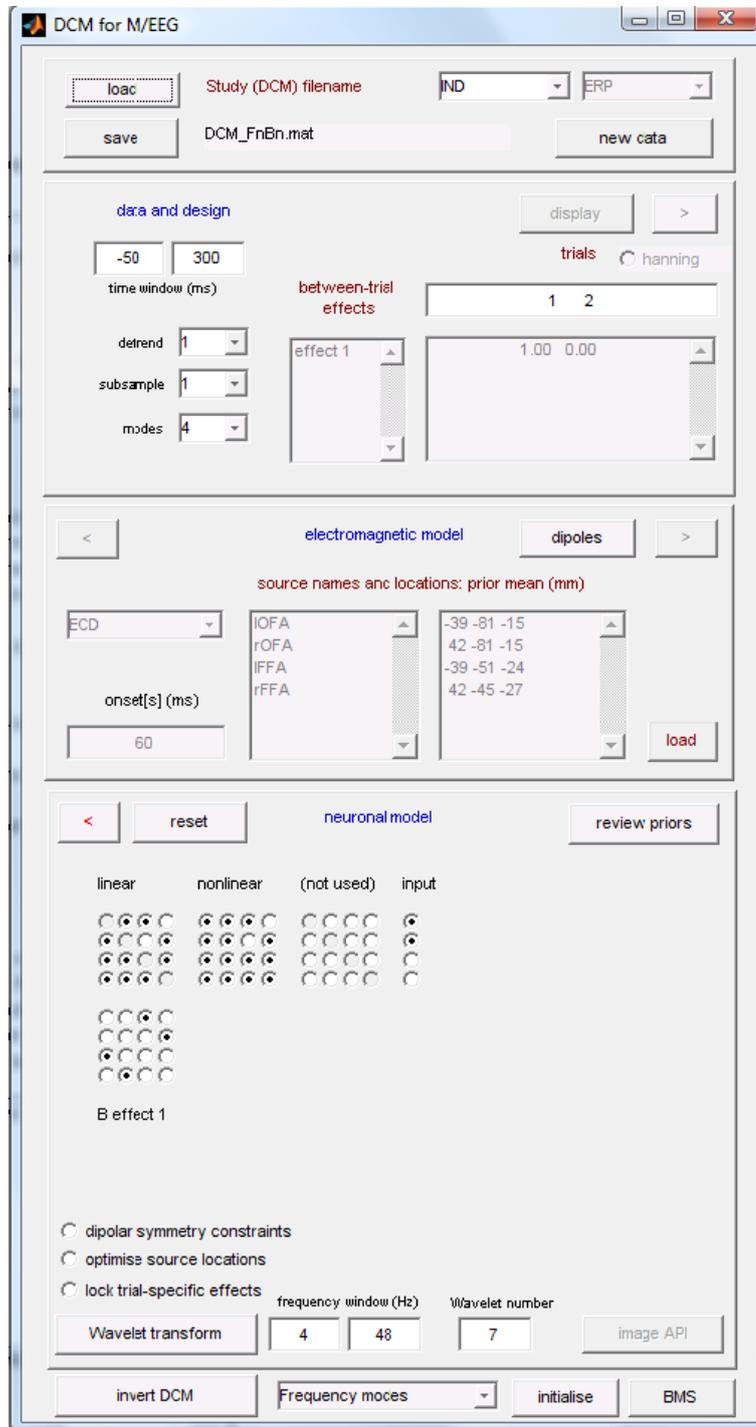
Figure 20.1: *DCM window configured analysing induced responses and the FnBn model specified.*

just to *play around* with, clean up or reduce the data to see what effect it might have without running additional processing steps outside DCM.

Press the "Display" button to look at the selected data. You will see the evoked responses for the two conditions (Figure 20.2) which help you get some idea about your choice of time window. It is possible to change the "detrend" and "subsample" values or the time window and press "Display' again to see what effect these changes have.

An important parameter for DCM-IR is the number of modes. These are the frequency modes mentioned above. The idea is that the main features of the time-frequency image can be represented by a small number of components with fixed frequency profiles that are modulated over time. These components can be determined automatically using "Singular Value Decomposition" (SVD). Generally SVD preserves information from the original time-frequency image and produces as many components as there are frequency bins. However, usually only the first few components are physiologically relevant and the rest are just noise. Using a small number of components will greatly speed-up DCM model inversion. You cannot know in advance what the optimal number of components for your data is. What you can do is try once with a relatively large number (e.g. 8) and then see from the time and frequency profile of the later components (in the Results view, see below) whether they are important. Then you can reduce the number and try again. For the example here it is sufficient to use 4 modes so change the number in "modes" from 8 to 4.

If you are happy with your data selection, the subsampling and the detrending terms, you can click on the > (forward) button, which will bring you to the next stage *electromagnetic model*. From this part, you can press the red < button to get back to the data and design part.

20.3.3   Electromagnetic model

With DCM-IR, you have two options for how to extract the source data for time-frequency analysis. Firstly, you can use 3 orthogonal single equivalent current dipoles (ECD) for each source and invert the resulting source model to get source waveforms. This option is suitable for multichannel EEG or MEG data. Alternatively, you can treat each channel as a source (LFP option). This is appropriate when the channels already contain source data either recorded directly with intracranial electrodes or extracted (e.g. using a beamformer).

Note that a difference with DCM for evoked responses is that the parameters of the spatial model are not optimized. This means that DCM-IR will project the data into source space using the spatial locations you provide.

We will use the ECD option. This requires specifying a list of source names in the left large text box and a list of MNI coordinates for the sources in the right large text box. Enter the following in the left box:

```
lOFA
rOFA
lFFA
rFFA
```

Now enter in the right text box:

```
-39 -81 -15
 42 -81 -15
-39 -51 -24
 42 -45 -27
```

These correspond to left Occipital Face Area, right Occipital Face Area, left Fusiform Face Area and right Fusiform Face Area respectively. See [10] for more details.

The onset-parameter determines when the stimulus, presented at 0 ms peri-stimulus time, is assumed to trigger the cortical induced response. In DCM, we usually do not model the rather small early responses, but start modelling at the first large deflection. Because the propagation of the stimulus impulse through the input nodes causes a delay, we found that the default value of 60 ms onset time is a good value for many responses where the first large deflection is seen around 100 ms. However, this value is a prior, i.e. the inversion routine can adjust it. The prior
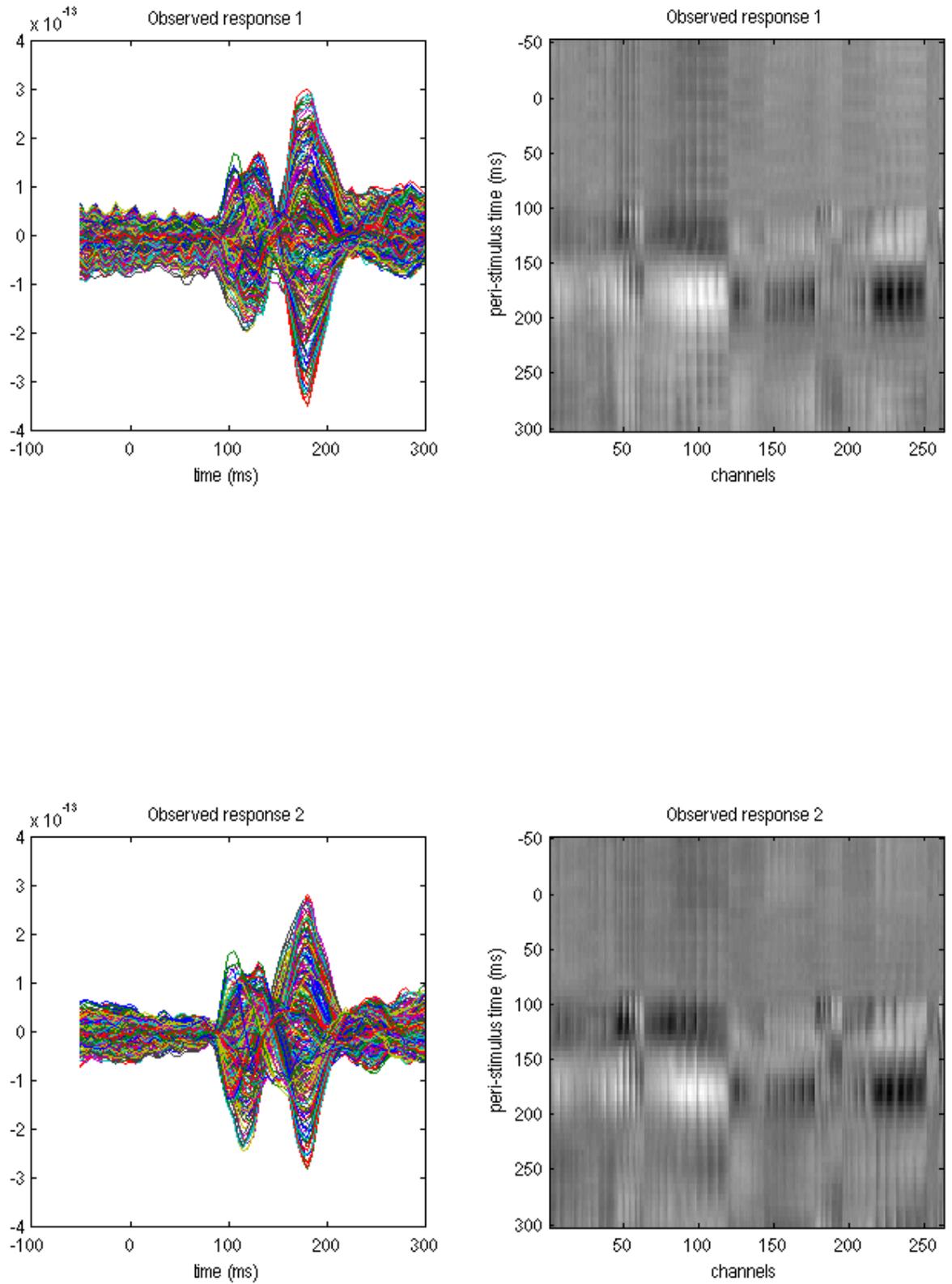
Figure 20.2: *Averaged evoked responses after configuring the 'Data and design' section.*

mean should be chosen according to the specific responses of interest. This is because the time until the first large deflection is dependent on the paradigm or the modality you are working in, e.g. audition or vision. You may also find that changing the onset prior has an effect on how your data are fitted. This is because the onset time has strongly nonlinear effects (a delay) on the data, which might cause differences in the maximum found at convergence, for different prior values. Note, that it is possible to enter more than one number in the "onset[s] (ms)" box. This will add several inputs to the model. These inputs can then be connected to different nodes and/or their timing and frequency profiles can be optimized separately.

When you want to proceed to the next model specification stage, hit the > (forward) button and proceed to the *neuronal model*. If you have not used the input dataset for 3D source reconstruction before you will be asked to specify the parameters of the head model at this stage. See 19.5.13.

20.4   Neuronal model

There are 4 (or more) matrices which you need to specify by button presses. In the first row there are matrices that define the connectivity structure of the model and in the second row there are matrices that specify which connections are affected by experimental effects. All the matrices except one are square. In each of these square matrices you specify a connection *from* a source area *to* a target area. For example, switching on the element (2, 1) means that you specify a connection from area 1 to 2 (in our case from lOFA to rOFA). Some people find the meaning of each element slightly counter-intuitive, because the column index corresponds to the source area, and the row index to the target area. This convention is motivated by direct correspondence between the matrices of buttons in the GUI and connectivity matrices in DCM equations and should be clear to anyone familiar with matrix multiplication.

The leftmost matrix in the first row specifies the *linear* connections. These are the connections where frequency dynamics in one source affects the dynamics at the same frequencies in another source. Note that all connections in the model should be at least linear, so if you think some connection should be present in the model, the corresponding button in this matrix should be on. Also the buttons on the leading diagonal of the matrix are always on because each node in the model has a linear intrinsic connection with negative sign. This means that the activity has a tendency to dissipate. To the right of the linear connectivity matrix there is a *nonlinear* connectivity matrix. The idea here is the same, just remember to enable the corresponding linear connection as well. When a connection is nonlinear, a frequency mode in the source node can affect all the frequency modes in the target node. Intrinsic connections can be made non-linear as well. It is actually recommended to always make the intrinsic connections non-linear unless there is a good theoretical reason not to do it. Since we are mainly interested in non-linearities in the extrinsic connections we would like to be over-conservative and first explain away anything that can be explained by non-linearities in the intrinsic connnections.

The rightmost matrix in the first row is the input matrix. It is usually not square, and in the case of a single input, as we have here, is reduced to a column vector. The entries of this vector specify which areas receive the external input (whose onset time we specified above). In the case of several inputs the input matrix will have several columns.

The matrix (matrices) in the second row specify which of the connections defined in the first row can be modified by experimental effects. A connection which is not modified will have the same value for all conditions. If you don't allow modification of any of the connections, then exactly the same model will be fitted to all conditions. For the purpose of allowing modification by experimental effects, it does not matter whether a connection is linear or non-linear. Hence, there is one modulation matrix per experimental effect (defined in the "Data and design" panel). In our case there is only one effect - faces vs. scrambled faces. Also self connections can be modified by experimental effects, thus the diagonal entries of the second row matrices can also be toggled.

Figure 20.3 is taken from the paper of Chen et al. [9] and shows several alternative models that could apply to the data. We will start by specifying the model with nonlinear forward and backward connections (FnBn) and with effect of condition on these connections. The corresponding button configuration is shown in Figure 20.4. Compare the depiction of FnBn model in
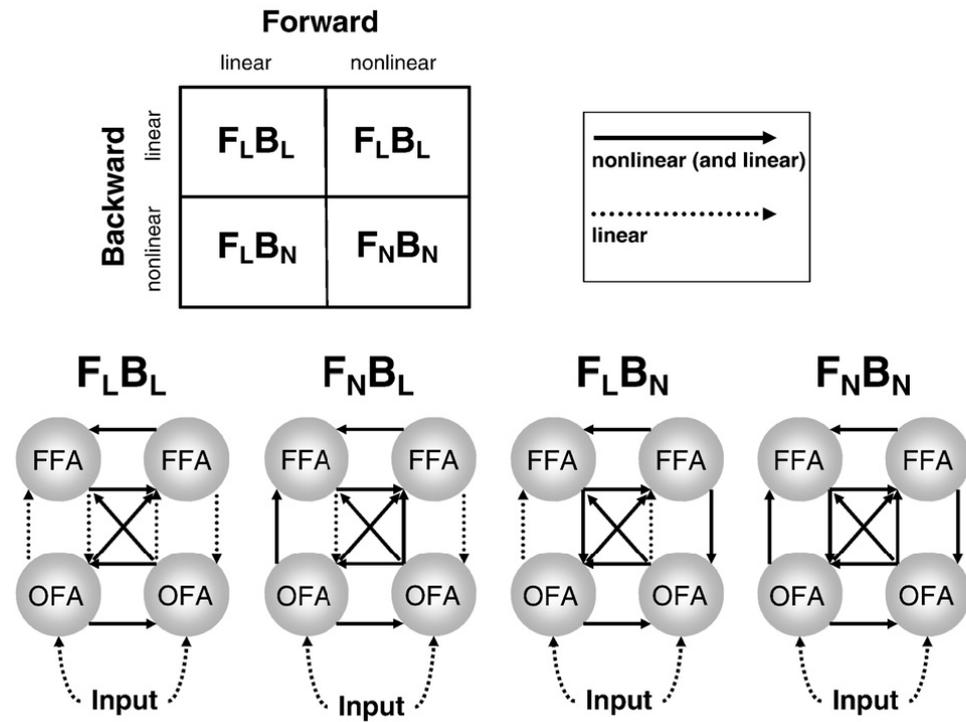
Figure 20.3: *Four different DCM-IR models proposed by Chen et al. [9]*

Figure 20.3 with the specification in Figure 20.4 to see the correspondence. Note that the effect of condition is not shown in Figure 20.3. Now copy the specification to the DCM GUI.

At the bottom of this panel there are additional radio buttons for options that are not relevant for DCM-IR. Below these buttons there are controls for specifying the parameters of the wavelet transform for computing the time-frequency decomposition. We will keep the default frequency window 4 to 48 Hz and increase the number of wavelet cycles to 7. You can press the *Wavelet transform* button to preview the time-frequency plots and optimize the parameters if necessary before inverting the model.

20.5 Estimation

When you have finished model specification, you can hit the *invert DCM* button in the lower left corner. DCM will now estimate the model parameters. You can follow the estimation process by observing the model fit in the output window. Note that in DCM-IR there is no difference
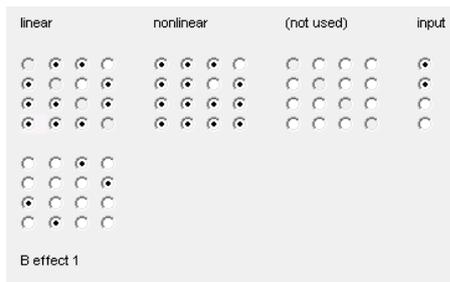
Figure 20.4: *Connectivity configuration for the FnBn model.*

between the hidden states and the predicted responses because the dynamics of the hidden states fit directly the time course of frequency modes (shown as dotted lines in the middle plot). This is different from DCM for ERP where the hidden states correspond to neural dynamics and a subset of the hidden states (activation of pyramidal cells) are projected via the forward model to generate predictions of sensor data. In the MATLAB command window, you will see each iteration print an expectation-maximization iteration number, free energy F, and the predicted and actual change of F following each iteration step. At convergence, DCM saves the results in a DCM file, by default named `DCM_*.mat` where `*` corresponds to the name of the original SPM MEG file you specified. You can save to a different name, e.g. if you are estimating multiple models, by pressing 'save' at the top of the GUI and writing to a different name.

20.6   Results

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

20.6.1   Frequency modes

This will display the frequency profiles of the modes, identified using singular value decomposition of spectral dynamics in source space (over time and sources).

20.6.2   Time modes

This will display the observed time courses of the frequency modes (dashed lines) and the model predictions (solid lines). Here you can also see whether the activity picked up by the minor modes is noise, which is helpful for optimizing the number of modes.

20.6.3   Time-Frequency

This will display the observed time-frequency power data for all pre-specified sources (upper panel) and the fitted data features (lower panel).

20.6.4   Coupling (A-Hz)

This will display the coupling matrices representing the coupling strength from source to target frequencies. These matrices are obtained by multiplying the between-mode matrices estimated with the frequency profiles of the modes (see [10]). The arrangement of the matrices corresponds to arrangements of the buttons in the connectivity matrices above.

20.6.5   Coupling (B-Hz)

This presentation of results is similar to the above and reports modification of coupling by condition (eg. in our example it shows which frequency couplings are different for faces as opposed to scrambled faces).

20.6.6   Coupling (A-modes)

This will display the coupling matrices between modes and the conditional probabilities that the coefficients are different from zero. This representation is useful for diagnostics when something is wrong with the inversion, but the physiological interpretation is less straightforward.

20.6.7   Coupling (B-Hz)

This presentation is similar to the above and reports the modification of coupling by condition.
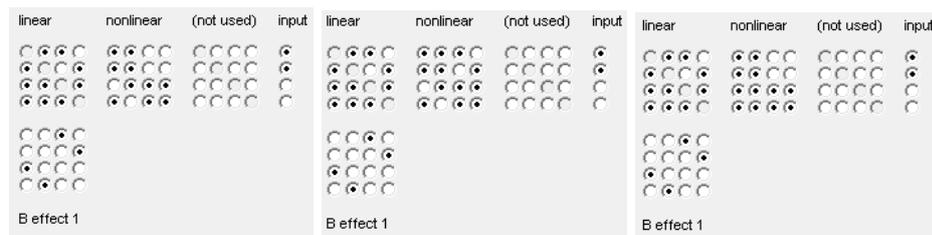
Figure 20.5: *Connectivity configurations for the alternative models. Left to right: FlBl, FlBn, FnBl.*

20.6.8   Input (C-Hz)

This shows the frequency profiles of the inputs estimated. This is again a multiplication between the mode-specific coefficients and the frequency profiles of the modes.

20.6.9   Input (u-ms)

This shows the time courses of the inputs.

20.6.10   Dipoles

This shows the positions of the sources as specified in the "Electromagnetic model" section.

20.6.11   Save as img

Here you can save the cross-frequency coupling matrices as images. If you are analyzing a group of subjects you can then enter these images into parametric statistical tests to find common features in coupling and coupling changes accross subjects. The image names will include identifiers like "A12" or "B31" which relate to the source connection matrices; either the basic (A) or experimental effects (B).

20.7   Model comparison

You can now compare the fully nonlinear model with alternative models (eg. those shown in Figure 20.3). You can start by saving the DCM you have already specified under a different name using the *Save* button. Then just modify the connectivity matrices and reinvert the DCM by pressing the "Estimated" button (but not using previous posterior or prior estimates). As an exercise, you can specify the other models from Figure 20.3 yourself. If in doubt look at Figure 20.5 for the three alternative models. Once you have specified and inverted the three additional models, you can perform Bayesian model comparison.

Press the BMS button. This will open the SPM batch tool for model selection. Specify a directory to write the output file to. For the "Inference method" select "Fixed effects" (see [65] for additional explanations). Then click on "Data" and in the box below click on "New: Subject". Click on "Subject" and in the box below on "New: Session". Click on models and in the selection window that comes up select the DCM mat files for all the models (remember the order in which you select the files as this is necessary for interpreting the results). Then run the model comparison by pressing the green "Run" button. You will see, at the top, a bar plot of the log-model evidences for all models 20.6. At the bottom, you will see the posterior probability, for each model, given the data. By convention, a model can be said to be the best among a selection of other models, with strong evidence, if its log-model evidence exceeds all other log-model evidences by at least 3. In our case the FnBn model is superior to the other models as was found in the original paper [9] for a different group of subjects.
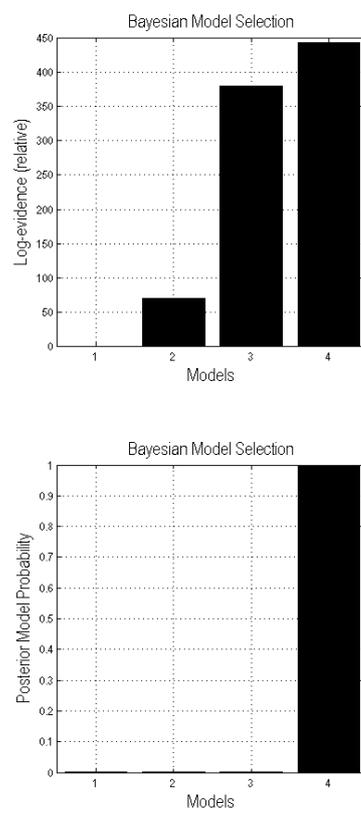
Figure 20.6: *Bayesian comparison of the four DCM-IR models shown in Figure 20.3.*

Chapter 21

DCM for Phase Coupling

This chapter presents an extension of the Dynamic Causal Modelling (DCM) framework to the analysis of phase-coupled data. A weakly coupled oscillator approach is used to describe dynamic phase changes in a network of oscillators. The influence that the phase of one oscillator has on the change of phase of another is characterised in terms of a Phase Interaction Function (PIF) as described in [60]. SPM supports PIFs specified using arbitrary order Fourier series. However, to simplify the interface, one is restricted to simple sinusoidal PIFs with the GUI.

21.1   Data

We will use the merged epoched MEG dataset from Chapter 19.5:

```
cdbespm8_SPM_CTF_MEG_example_faces1_3D.mat
cdbespm8_SPM_CTF_MEG_example_faces1_3D.dat
```

See 19.5.1 for instructions on how to generate this file from raw MEG data. DCM-Phase requires a head model and coregistration. If you have been following the previous chapters of this tutorial, these should already be available in the dataset. Otherwise, you should perform the 'Prepare' and '3D Source reconstruction' steps described earlier in the chapter, with the latter comprising the MRI, Co-register, Forward and Save sub-steps (see 19.5.13).

21.2   Getting Started

You need to start SPM and toggle "EEG" as the modality (bottom-right of SPM main window), or start SPM with `spm eeg`. In order for this to work you need to ensure that the main SPM directory is on your MATLAB path. After calling `spm eeg`, you see SPM's graphical user interface, the top-left window. The button for calling the DCM-GUI is found in the second partition from the top, on the right hand side. When pressing the button, the GUI pops up (Figure 20.1).

21.3   Data and design

You should switch the DCM model type to "PHASE" which is the option for DCM-Phase. Press "new data" and select the data file **cdbespm8_SPM_CTF_MEG_example_faces1_3D.mat**. This is an epoched data file with multiple trials per condition. On the right-hand side enter the trial indices

```
1 2
```

for the 'face' and 'scrambled' evoked responses (we will model both trial types). The box below this list allows for specifying experimental effects on connectivity. Enter

```
1 0
```

in the first row of the box. This means that "face" trial types can have different connectivity parameters than "scrambled" trial types. If we now click somewhere outside the box, a default name will be assigned to this effect - "effect1". It will appear in the small text box next to the coefficients box. It is possible to change this name to something else e.g. "face". Now we can select the peristimulus time window we want to model. Set it to:

```
1 300
```

ms. Select `1` for "detrend", to remove the mean from each data record. The `sub-trials` option makes it possible to select just a subset of trials for the analysis (select 2 for every second trial, 3 - for every third etc.). This is useful because DCM-Phase takes quite a long time to invert for all the trials and you might want to first try a smaller subset to get an idea about the possible results. Here we will assume that you used all the trials (sub-trials was set to 1). You can now click on the > (forward) button, which will bring you to the next stage *electromagnetic model*. From this part, you can press the red < button to get back to the data and design part.

21.4   Electromagnetic model

With DCM-Phase, there are two options for how to extract the source data. Firstly, you can use 3 orthogonal single equivalent current dipoles (ECD) for each source, invert the resulting source model to get 3 source waveforms and take the first principal component. This option is suitable for multichannel EEG or MEG data. Alternatively, you can treat each channel as a source (LFP option). This is appropriate when the channels already contain source data either recorded directly with intracranial electrodes or extracted (e.g. using a beamformer).

Note that a difference to DCM for evoked responses is that the parameters of the spatial model are not optimized. This means that DCM-Phase (like DCM-IR) will project the data into source space using the spatial locations you provide.

We will use the ECD option and specify just two source regions. This requires specifying a list of source names in the left large text box and a list of MNI coordinates for the sources in the right large text box. Enter the following in the left box:

```
LOFA
LFFA
```

Now enter in the right text box:

```
-39 -81 -15
-39 -51 -24
```

These correspond to left Occipital Face Area, and left Fusiform Face Area. The onset-parameter is irrelevant for DCM-Phase. Now hit the > (forward) button and proceed to the *neuronal model*. Generally, if you have not used the input dataset for 3D source reconstruction before you will be asked to specify the parameters of the head model at this stage. See 19.5.13.

21.5   Neuronal model

We will now define a coupled oscillator model for investigating network synchronization of alpha activity. To this end, we first enter the values 8 and 12 to define the frequency window. The wavelet number is irrelevant for DCM-Phase. After source reconstruction (using a pseudo-inverse approach), source data is bandpass filtered and then the Hilbert transform is used to extract the instantaneous phase. The DCM-Phase model is then fitted used standard routines as described in [60].

Figure 21.1 shows the four models we will apply to the M/EEG data. We will first fit model 4. This model proposes that alpha activity in region LOFA changes its phase so as to synchronize with activity in region LFFA. In this network LFFA is the master and LOFA is the slave. Moreover, the connection from LFFA to LOFA is allowed to be different for scrambled versus unscrambled faces.
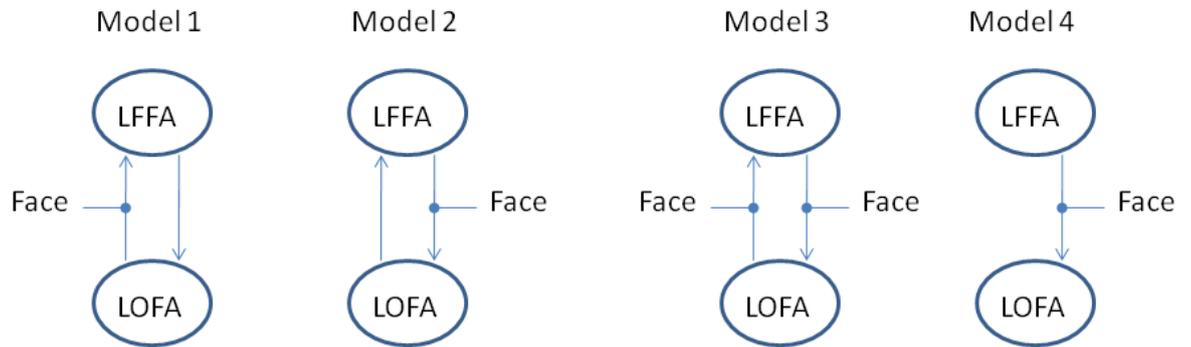
Figure 21.1: *Four different DCM-Phase models*
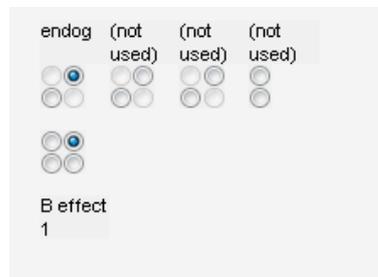
Figure 21.2: *Radio button configurations for DCM-Phase model 4*

The connectivity for Model 4 can be set up by configuring the radio buttons as shown in Figure 21.2. You can now press the `Invert DCM` button. It can take up to an hour to estimate the model parameters depending on the speed of your computer.

21.6 Results

After estimation is finished, you can assess the results by choosing from the pull-down menu at the bottom (middle). The `Sin(Data)-Region i` option will show the sin of the phase data in region i, for the first 16 trials. The blue line corresponds to the data and the red to the DCM-Phase model fit. The `Coupling(As)` and `Coupling(Bs)` buttons display the estimated endogenous and modulatory activity shown in Figure 21.3.

If one fits all the four models shown in Figure 21.1 then they can be formally compared using Bayesian Model Selection. This is implemented by pressing the `BMS` button. You will need to first create a directory for the results to go in e.g. `BMS-results`. For 'Inference Method' select FFX (the RFX option is only viable if you have models from a group of subjects). Under 'Data', Select 'New Subject' and under 'Subject' select 'New Session'. Then under 'Models' select the `DCM.mat` files you have created. Then press the green play button. This will produce the results plot shown in Figure 21.4. This leads us to conclude that LFFA and LOFA act in master slave arrangement with LFFA as the master.

21.7 Extensions

In the DCM-Phase model accessible from the GUI, it is assumed that the phase interaction functions are of simple sinusoidal form ie. $a_{ij} \sin(\phi_j - \phi_i)$. The coefficients a_{ij} are the values shown in the endogenous parameter matrices in eg. Figure 21.3. These can then be changed by an amount b_{ij} as shown in the modulatory parameter matrices. It is also possible to specify

changes in coupling (Bs)
face

Posterior mean

	LOFA	LFFA
LOFA	0.000	-0.517
LFFA	0.000	0.000

Posterior probability (Bs) > 0

	LOFA	LFFA
LOFA	NaN	0.965
LFFA	NaN	NaN

Figure 21.3: *The figure shows the estimated parameters for endogenous coupling (left column) and modulatory parameters (right column) for the 4th DCM.*

Bayesian Model Selection
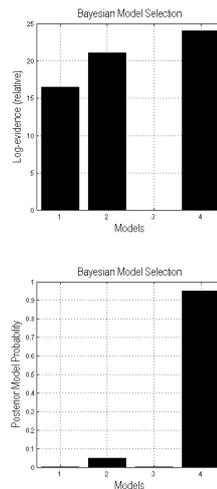
Data identity has not been verified

Figure 21.4: *Bayesian comparison of the four DCM-Phase models shown in Figure 21.1.*

and estimate DCM-Phase models using matlab scripts. In this case it is possible to specify more generic phase interaction functions, such as arbitrary order Fourier series. Examples are given in [60].

Chapter 22

DCM for Steady State Responses: Anaesthesia Depth in Rodent Data

22.1   Overview

This chapter describes the analysis of a 2-channel Local Field Potential (LFPs) data set using dynamic causal modelling. The LFPs were recorded from a single rodent using intracranial electrodes [53]. We thank Marc Tittgemeyer for providing us with this data. The theory behind DCM for steady state responses (DCM-SSR) is described in [54].

We measured local field potentials from primary (A1) and secondary auditory (A2) cortex in a rodent following the application of four different doses of the anaesthetic agent Isoflurane; 1.4, 1.8, 2.4 and 2.8mg. The rodent was presented with a white noise auditory input for several minutes at each anaesthetised level and time series recordings were obtained for the entire epoch. We performed a steady state DCM analysis using different models, defined according to the extrinsic connections between A1 and A2.

Using Bayesian model comparison we found very strong evidence (Bayes Factor$_{1,2}$ > 100) in favour of a model comprising a network of two neural masses connected by forward (excitatory) and backward (inhibitory + excitatory) connections (`m1`). This outperformed both a model comprising two neural masses with forward (excitatory) connections only (`m2`) and a model comprising two neural masses and lateral (inhibitory + excitatory + excitatory) connections only (`m3`). Within this best performing model we observed changes in forward and backward connections dependent on dose. The strength of inhibitory backward connections increased with successive anaesthetic dose, constituting a greater GABAergic influence postsynaptically for higher doses. Conversely, the solely excitatory forward connections reduced in strength with successive doses

In what follows, these results will be recreated step-by-step using SPM8.

To proceed with the data analysis, first download the data set from the SPM website[1]. The data comprises a data file called `dLFP_white_noise_r24_anaes.dat` and its corresponding mat file `dLFP_white_noise_r24_anaes.mat`. This has been converted from ASCII data using `spm_lfp_txt2mat_anaes.m` also on the website and subsequently downsampled to 125 Hz. The conversion script can be altered to suit your own conditions/sampling parameters.

22.2   The data

- To check data parameters after conversion using ASCII files: in the SPM M/EEG GUI press DISPLAY/M/EEG.

- In our data set we can see there are five trials: four depths of anaesthetic: `Iso14`, `Iso18`, `Iso24` and `Iso28` and one awake trial `awake`.

---

[1]Anaesthesia Depth in Rodent Dataset: http://www.fil.ion.ucl.ac.uk/spm/data/dcm_ssr/

- We can also see that the data are clean for both the A1 and A2 channels, to view this press OTHER, as the data are LFP type.

- We are now ready to begin the DCM analysis. To open the DCM GUI press DCM in the SPM M/EEG GUI.

22.3 Dynamic Causal Modelling of Steady State Responses

- Before you begin any DCM analysis you must decide on two things: the data feature from your time series and the model you wish to use.

- For our long time series we will examine the steady state and so in the top panel of the DCM GUI select SSR in the data drop-down menu.

- Next in the second drop down menu we select our model. For this LFP data we employ the LFP model and so this is then selected.

- Then we are ready to load our data: press NEW DATA and select the file dLFP_white_noise_r24_anaes.mat.

- Press the red arrow to move forward.

- The second panel allows you to specify the data and design. We will use the first 30sec of the time series for our analysis. To specify this timing enter 1 and 30000 in the time window.

- Next we select the detrending parameters which we set to 1 for detrend, 1 for subsample (as the data has already been downsampled) and 2 for the modes (in this case this is the same as the number of channels) using the drop down menus.

- We can then specify which trials we want to use. Since we are interested in the anaesthetized trials we enter 1 2 3 4 under the trials label and effect1 effect2 effect3 in the between trial effects panel. Next we specify the design matrix. This is entered numerically in the large panel. Since we have 4 trials and 3 between trial effects (one less) we enter a matrix with rows:[0 1 0 0] (row 1), [0 0 1 0] (row 2) and [0 0 0 1] (row 3). This will allow us to examine "main effect" differences between the four conditions.[2]

- Press the red arrow to move forward.

- The third panel contains the spec for the electromagnetic model. This is very simple for LFPs. In the drop down menu select LFP. In the source names panel, enter A1 and A2. You are finished.

- Press the red arrow to move forward.

- At this point all that is left to specify is the neuronal model. We wish to compare three different models so we can save the specifications so far using the SAVE button and reload the above specs for all three neuronal models.

- To specify the neuronal model, load the DCM (that you just saved) as it has been so far specified.

- Our first model is the forward model. So we specify forward connections from A1 to A2 and forward connections from A2 to A1 (Fig 22.2)

- We also specify the input. In our experiment we assume input at both sources.

---

[2]The effect of anaesthesia could also be specified as a single linear effect, for instance by using a vector of mean-corrected anaesthetic concentrations: [14 18 24 28] - mean([14 18 24 28]) = [-7 -3 3 7]. Here we specified 3 separate effects to show that even when the effects are specified separately, there is a consistent trend in connectivity changes with the increase in anaesthetic concentration.
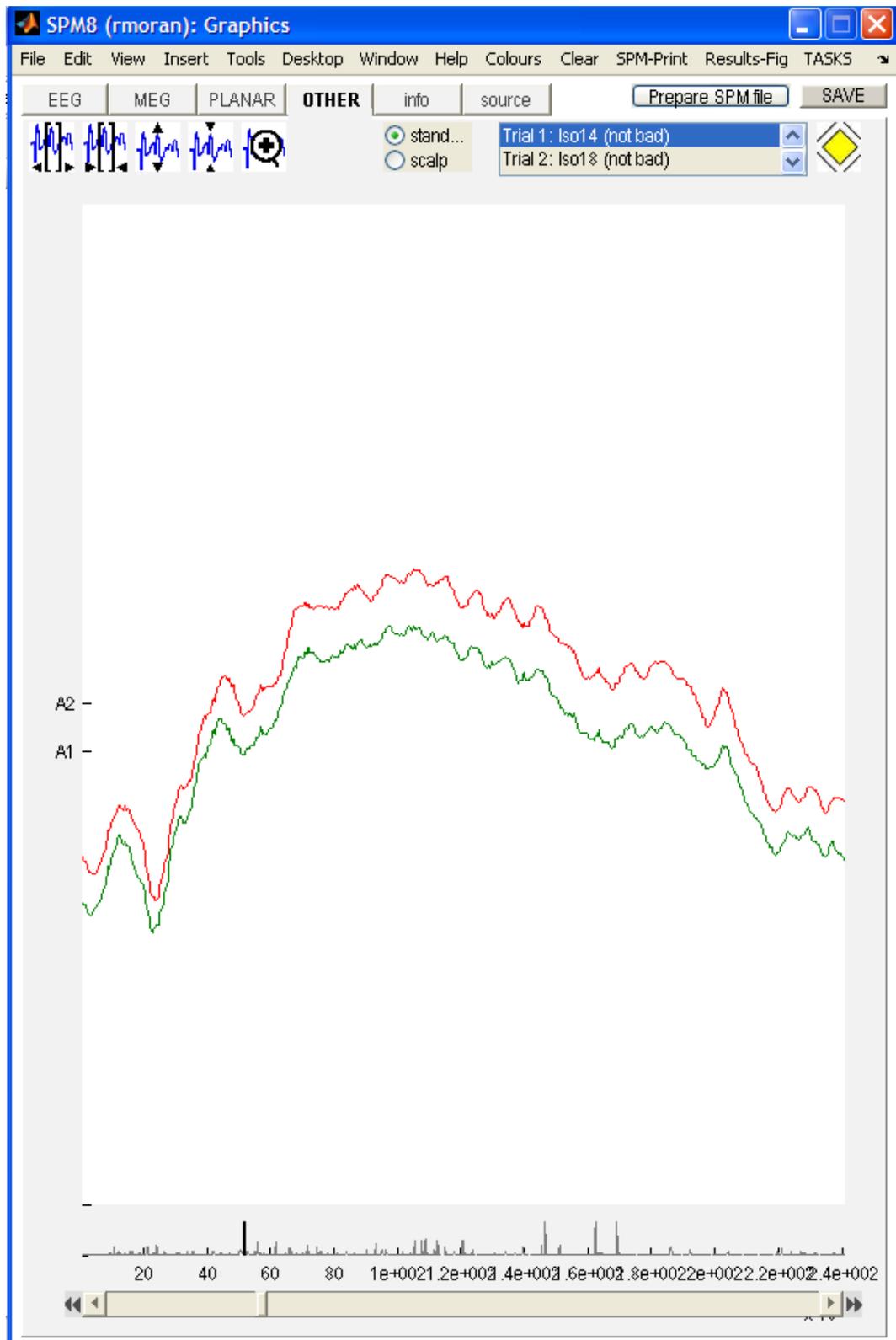
Figure 22.1: *Screenshot of data exploration table. By displaying the converted file* LFP_white_noise_r24_anaes.mat *one can check the data quality, length, trials and sampling parameters.*
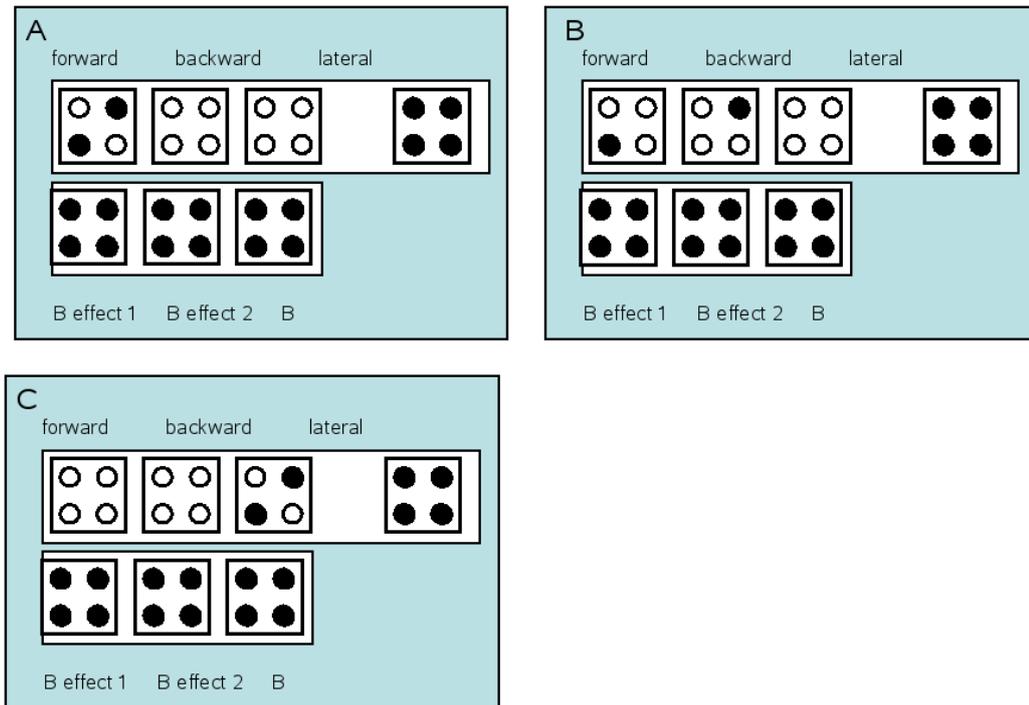
Figure 22.2: *Model Specifications A: forward model, B: forward-backward model, C: lateral model.*

- We finally specify the B effects where we enter our hypothesis of connectivity changes between trial 1 (Iso1.4mg) trial 2 trial 3 and trial 4. Changes will be specified relative to trial 1.

- We enter the off diagonal entries to correspond to forward connections (as entered in the above panel) and the main diagonal entries to specify intrinsic connectivity changes with A1 and A2 due to (anaesthetic) condition.

- Finally we enter the frequencies we are interested in: we will fit frequencies from 4 to 48 Hz.

- To invert the model press INVERT DCM.

- Repeat the procedure after loading the saved specs and repeating for new neuronal models as per Fig 22.2.

22.4   Results

- Once all models have run, we compare their evidences to find to best or winning model.

- To do this press the BMS button. This will open the SPM batch tool for model selection. Specify a directory to write the output file to. For the INFERENCE METHOD select `Fixed effects` (see [65] for additional explanations). Then click on DATA and in the box below click on NEW: SUBJECT. Click on SUBJECT and in the box below on NEW: SESSION. Click on models and in the selection window that comes up select the DCM mat files for all the models (remember the order in which you select the files as this is necessary for interpreting the results). Then run the model comparison by pressing the green RUN button. You will see, at the top, a bar plot of the log-model evidences for all models 20.6. At the bottom,

Figure 22.3: *Off diagonal bar charts show forward (A1 to A2) and backward (A2 to A1) connections strengths for trials 1 (1.4 Iso), 2 (1.8 Iso), 3 (2.4 Iso) and 4 (2.8 Iso)*

you will see the probability, for each model, that it produced the data. By convention, a model can be said to be the best among a selection of other models, with strong evidence, if its log-model evidence exceeds all other log-model evidences by at least 3. You can also compare model evidences manually if you load the DCMs into Matlab's workspace and find the evidence in the structure under `DCM.F`

- We can then examine the fits and posterior parameters of the winning model by loading it into the DCM GUI and selecting options from the drop down results menu.

- In the "trial specific effects" window we are interested particularly in the off-diagonal entries specifying trial specific extrinsic connectivities. Here for the f-b model we see increasing inhibitory (backward) connectivity and decreasing (forward) connectivity for increasing Isoflorane levels (Fig 22.3.

Chapter 23

Using DARTEL

DARTEL[1] 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[2] and increased sensitivity[3].

23.1   Using DARTEL for VBM

The following procedures could be specified one at a time, but it is easier to use the batching system. The sequence of jobs (use the *TASKS* pull-down from the *Graphics* window to select *BATCH*) would be:

- **Module List**

    - **SPM→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.

    - **SPM→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.

    - **SPM→Tools→DARTEL Tools→Run DARTEL (create Template)**: Determine the nonlinear deformations for warping all the grey and white matter images so that they match each other.

    - **SPM→Tools→DARTEL Tools→Normalise to MNI Space**: Actually generate the smoothed "modulated" warped grey and white matter images.

Alternatively, the **New Segment** procedure could be used. Although still at the slightly experimental stages of development, this procedure has been found to be generally more robust than the implementation of "Unified Segmentation" from SPM→Spatial→Segment (which is the version from the Segment button - the same as that in SPM5). The new segmentation can require quite a lot of memory, so if you have large images (typically greater than about $256 \times 256 \times 150$) and trying to run it on a 32 bit computer or have relatively little memory installed, then it may throw up an out of memory error. The new segmentation procedure includes the option to generate DARTEL "imported" data, so the **Initial Import** step is skipped.

- **Module List**

    - **SPM→Tools→New Segment**: To generate the roughly (via a rigid-body) aligned grey and white matter images of the subjects.

---

[1]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.

[2]Less smoothing is needed, and there are fewer problems relating to how to interpret the differences.

[3]More sensitivity could mean that fewer subjects are needed, which should save shed-loads of time and money.

- **SPM→Tools→DARTEL Tools→Run DARTEL (create Template)**: Determine the nonlinear deformations for warping all the grey and white matter images so that they match each other.

- **SPM→Tools→DARTEL Tools→Normalise to MNI Space**: Actually generate the smoothed "modulated" warped grey and white matter images.

The segmentation and importing steps of these two alternative processing streams are described next.

23.1.1   Using Spatial→Segment and DARTEL Tools→Initial Import

The first step is to classify T1-weighted scans[4] of a number of subjects into different tissue types via the Segmentation routine in SPM. The *SPM→Spatial→Segment* pull-down can be used here:

- **Segment**

    - **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**: It is suggested that *Native Space* grey (and possibly white) matter images are created. These are c1*.img and c2*.img. 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 *SPM→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 version of the image is needed - usually only the grey and white matter tissue classes are used (ie choose *None*).

    - **Grey Matter**: Yes, you need this.

    - **White Matter**: Yes, you also need this.

---

[4]Other types of scan may also work, but this would need some empirical exploration.

– **CSF**: The CSF is not usually segmented very reliably because the segmentation only
has tissue probability maps for GM WM an 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.

23.1.2   Using Tools→New Segment

Note: This subsection will be elaborated on later.

There is a new segmentation option in SPM8, which can be found within Tools→New Segment.
If this option is used, then the "imported" tissue class images (usually rc1.nii and rc2.nii) would
be generated directly and the initial import step is skipped. It is also suggested that *Native Space*
versions of the tissues in which you are interested are also generated. For VBM, these are usually
the c1*.nii files, as it is these images that will eventually be warped to MNI space. Both the
imported and native tissue class image sets can be specified via the Native Space options of the
user interface.

23.1.3   Using DARTEL Tools→Run DARTEL (create Template)

The output of the previous step(s) are a series of rigidly aligned tissue class images (grey matter
is typically encoded by rc1*.nii and white matter by rc2*.nii – see Fig 23.1). 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 *SPM→Tools→Dartel Tools→Run DARTEL (create
Template)*.

- **Run DARTEL (create Template)**

  – **Images**

    * **Images**: Select all the rc1*.nii files generated by the import step.
    * **Images**: Select all the rc2*.nii files, in the same subject order as the rc1*.nii files.
      The first rc1*.nii is assumed to correspond with the first rc2*.nii, the second with
      the second, and so on.

  – **Settings**: Default settings generally work well, although you could try changing them
    to see what happens. A series of templates are generated called Template_basename_0.nii,
    Template_basename_1.nii etc. If you run multiple DARTEL sessions, then it may be a
    good idea to have a unique template basename for each.

The procedure begins by computing an initial template from all the imported data. If u_rc1*.nii
files exist for the images, then these are treated as starting estimates and used during the creation
of the initial template. If any u_rc1*.nii files exist from previous attempts, then it is usually
recommended that they are removed first (this sets all the starting estimates to zero). Template
generation incorporates a smoothing procedure, which may take a while (several minutes). Once
the original template has been generated, the algorithm will perform the first iteration of the
registration on each of the subjects in turn. After the first round of registration, a new template
is generated (incorporating the smoothing step), and the second round of registration begins.
Note that the earlier iterations usually run faster than the later ones, because fewer "time-steps"
are used to generate the deformations. The whole procedure takes (in the order of) about a week
of processing time for 400 subjects.

The end result is a series of templates (see Fig 23.2), and a series of u_rc1*.nii files. The
first template is based on the average[5] of the original imported data, where as the last is the
average of the DARTEL registered data. The u_rc1*.nii files are flow fields that parameterise

---

[5] They are actually more similar to weighted averages, where the weights are derived from the Jacobian de-
terminants of the deformations. There is a further complication in that a smoothing procedure is built into the
averaging.
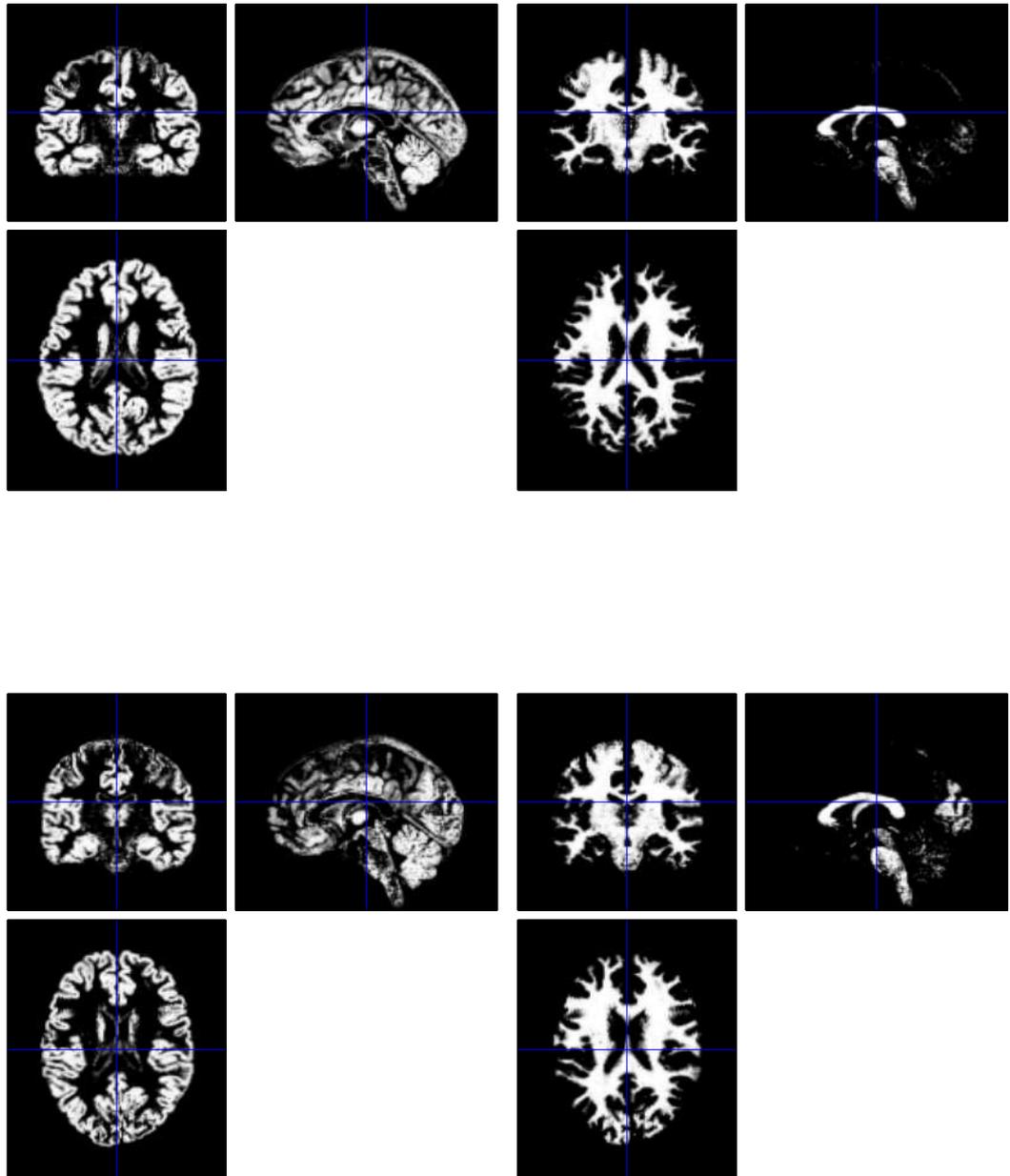
Figure 23.1: Imported data for two subjects (A and B). Top row: rc1A.nii and rc2A.nii. Bottom row: rc1B.nii and rc2B.nii.
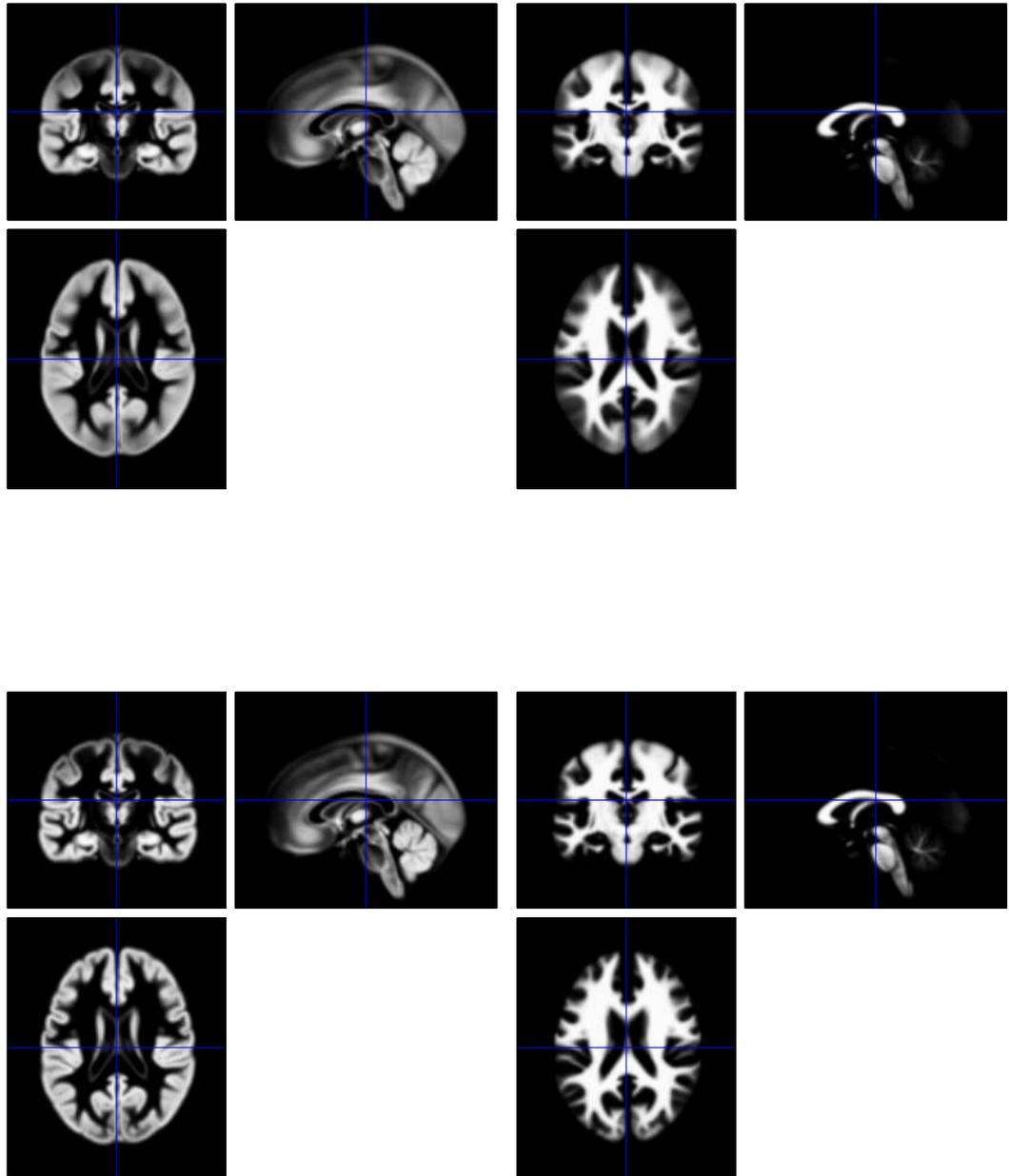
Figure 23.2: Different stages of template generation. Top row: an intermediate version of the template. Bottom row: the final template data.

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").

23.1.4 Using DARTEL Tools→Normalise to MNI Space

The next step is to create the Jacobian scaled ("modulated") warped tissue class images, by selecting *SPM→Tools→DARTEL Tools→Normalise to MNI Space*. The option for spatially normalising to MNI space automatically incorporates an affine transform that maps from the population average (DARTEL Template space) to MNI space, as well as incorporating a spatial smoothing step.

- **Normalise to MNI Space**

    - **DARTEL Template**: Specify the last of the series of templates that was created by *Run DARTEL (create Template)*. This is usually called *Template_6.nii*. Note that the order of the N volumes in this template should match the order of the first N volumes of the *toolbox/DARTEL/TPM.nii* file.

    - **Select according to** either *Few Subjects* or *Many Subjects*. For VBM, the *Many Subjects* option would be selected.

        * **Flow Fields**: Specify the flow fields (u_rc1*.nii) generated by the nonlinear registration.

        * **Images**: You may add several different sets of images.

            · **Images**: Select the c1*.nii files for each subject, in the same order as the flow fields are selected.

            · **Images**: This is optional, but warped white matter images can also be generated by selecting the c2*.nii files.

    - **Voxel sizes**: Specify the desired voxel sizes for the spatially normalised images (NaN, NaN, NaN gives the same voxel sizes as the DARTEL template).

    - **Bounding box**: Specify the desired bounding box for the spatially normalised images (NaN, NaN, NaN; NaN NaN NaN gives the same bounding box as the DARTEL template).

    - **Preserve**: Here you have a choice of *Preserve Concentrations* (ie not Jacobian scaled) or *Preserve Amount* (Jacobian scaled). The *Preserve Amount* would be used for VBM, as it does something similar to Jacobian scaling (modulation).

    - **Gaussian FWHM**: Enter how much to blur the spatially normalised images, where the values denote the full width at half maximum of a Gaussian convolution kernel, in units of mm. Because the inter-subject registration should be more accurate than when done using other SPM tools, the FWHM can be smaller than would be otherwise used. A value of around 8mm (ie $[8, 8, 8]$) should be about right for VBM studies, although some empirical exploration may be needed. If there are fewer subjects in a study, then it may be advisable to smooth more.

The end result should be a bunch of smwc1*.nii files[6] (possibly with smwc2*.nii if white matter is also to be studied).

---

[6] The actual warping of the images is done slightly differently, with the aim that as much of the original signal is preserved as possible. This essentially involves pushing each voxel from its position in the original image, into the appropriate location in the new image - keeping a count of the number of voxels pushed into each new position. The procedure is to scan through the original image, and push each voxel in turn. The alternative (older way) was to scan through the spatially normalised image, filling in values from the original image (pulling the values from the original). The results of the pushing procedure are analogous to Jacobian scaled ("modulated") data. A minor disadvantage of this approach is that it can introduce aliasing artifacts (think stripy shirt on TV screen) if the original image is at a similar - or lower - resolution to the warped version. Usually, these effects are masked by the smoothing.
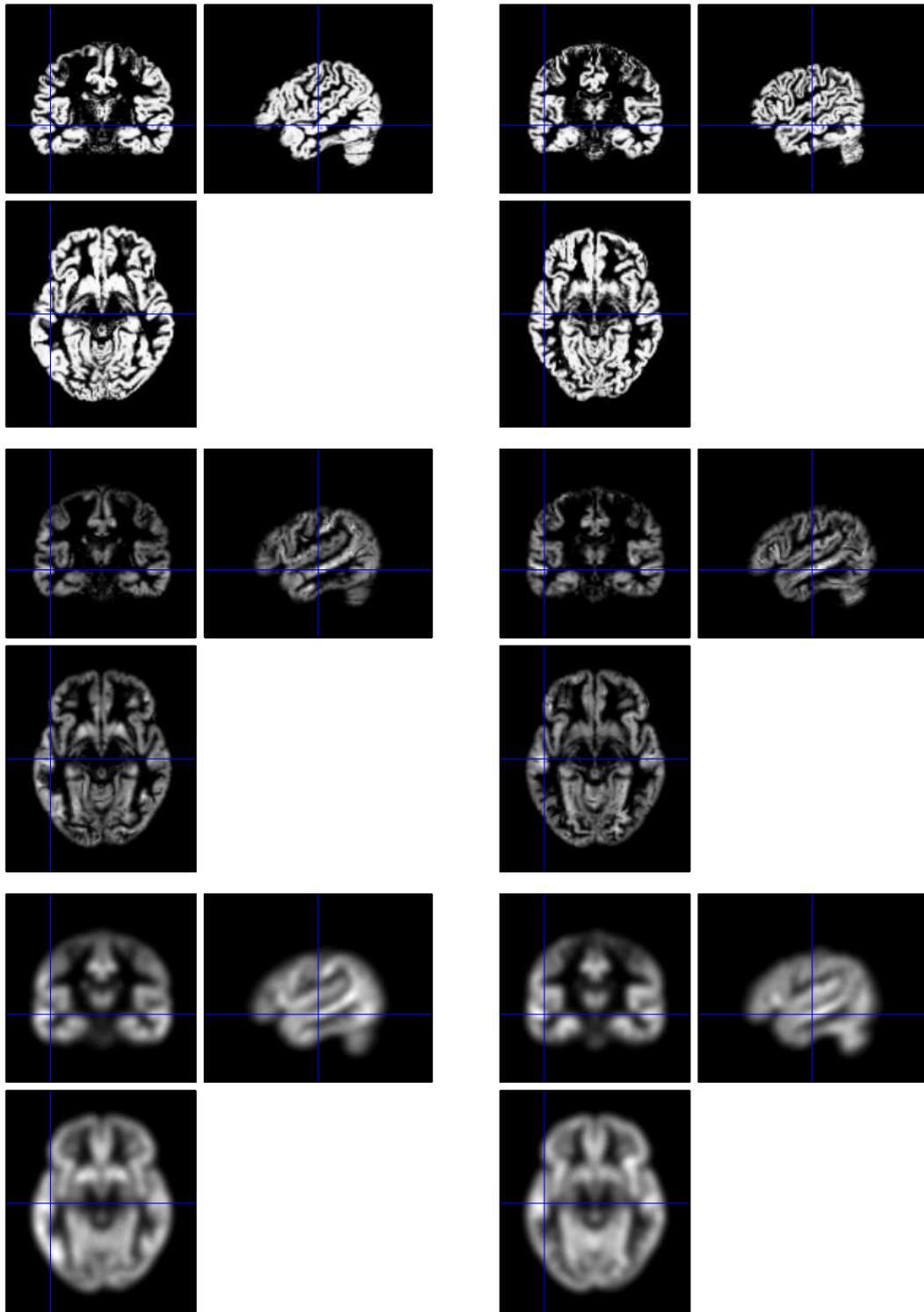
Figure 23.3: Pre-processing for VBM. Top row: Imported grey matter (rc1A.nii and rc1B.nii). Centre row: Warped with *Preserve Amount* option and zero smoothing ("modulated"). Bottom row: Warped with *Preserve Amount* option smoothing of 8mm (smwc1A.nii and smwc1B.nii).

The final step is to perform the statistical analysis on the preprocessed data (smwc1*.nii files), which should be in MNI space. The next section says a little about how data from a small number of subjects could be warped to MNI space.

23.2   Spatially normalising functional data to MNI space

Providing it is possible to achieve good alignment between functional data from a particular subject and an anatomical image of the same subject (distortions in the fMRI may prevent accurate alignment), then it may be possible to achieve more accurate spatial normalisation of the fMRI data using DARTEL. There are several advantages of having more accurate spatial normalisation, especially in terms of achieving more significant activations and better localisation.

The objectives of spatial normalisation are:

- To transform scans of subjects into alignment with each other. DARTEL was developed to achieve better inter-subject alignment of data.

- To transform them to a standard anatomical space, so that activations can be reported within a standardised coordinate system. Extra steps are needed to achieve this aim.

The option for spatially normalising to MNI space automatically incorporates an affine transform that maps from the population average (DARTEL Template space) to MNI space. This transform is estimated by minimising the KL divergence between the final template image generated by DARTEL and tissue probability maps that are released as part of SPM (in the new segmentation toolbox). MNI space is defined according to affine matched images, so an affine transform of the DARTEL template to MNI space would appear to be a reasonable strategy.

For GLM analyses, we usually do not wish to work with Jacobian scaled data. For this reason, warping is now combined with smoothing, in a way that may be a bit more sensible than simply warping, followed by smoothing. The end result is essentially the same as that obtained by doing the following with the old way of warping

- Create spatially normalised and "modulated" (Jacobian scaled) functional data, and smooth.

- Create spatially normalised maps of Jacobian determinants, and smooth by the same amount.

- Divide one by the other, adding a small constant term to the denominator to prevent divisions by zero.

This should mean that signal is averaged in such a way that as little as possible is lost. It also assumes that the procedure does not have any nasty side effects for the GRF assumptions used for FWE corrections.

Prior to spatially normalising using DARTEL, the data should be processed as following:

- If possible, for each subject, use *SPM→Tools→FieldMap* to derive a distortion field that can be used for correcting the fMRI data. More accurate within-subject alignment between functional and anatomical scans should allow more of the benefits of DARTEL for inter-subject registration to be achieved.

- Use either *SPM→Spatial→Realign→Realign: Estimate Reslice* or *SPM→Spatial→Realign Unwarp*. If a field map is available, then use the *Realign Unwarp* option. The images need to have been realigned and resliced (or field-map distortion corrected) beforehand - otherwise things are not handled so well. The first reason for this is that there are no options to use different methods of interpolation, so rigid-body transforms (as estimated by Realign but without having resliced the images) may not be well modelled. Similarly, the spatial transforms do not incorporate any masking to reduce artifacts at the edge of the field of view.

- For each subject, register the anatomical scan with the functional data (using *SPM → Spatial → Coreg → Coreg: Estimate*). No reslicing of the anatomical image is needed. Use *SPM→Util→Check Registration* to assess the accuracy of the alignment. If this step

is unsuccessful, then some pre-processing of the anatomical scan may be needed in order to skull-strip and bias correct it. Skull stripping can be achieved by segmenting the anatomical scan, and masking a bias corrected version (which can be generated by the segmentation option) by the estimated GM, WM and CSF. This masking can be done using *SPM→Util→Image Calculator* (*ImCalc* button), by selecting the bias corrected scan (m*.img), and the tissue class images (c1*.img, c2*.img and c3*.img) and evaluating "i1. × ((i2+i3+i4)>0.5)". If segmentation is done before coregistration, then the functional data should be moved so that they align with the anatomical data.

- Segment the anatomical data and generate "imported" grey and white matter images. If *SPM→Tools→New Segment* is used, then make sure that "imported" grey and white matter images are created. If *SPM→Spatial→Segment* (the SPM5 segmentation routine, which is the one under the *Segment* button), then an additional *SPM→Tools→DARTEL Tools→Initial Import* will be needed.

- To actually estimate the warps, use *SPM→Tools→DARTEL Tools→Run DARTEL (create Templates)* in order to generate a series of templates and a flow field for each subject.

In principle (for a random effects model), you could run the first level analysis using the native space data of each subject. All you need are the contrast images, which can be warped and smoothed. Alternatively, you could warp and smooth the reslices fMRI, and do the statistical analysis on the spatially normalised images. Either way, you would select *SPM→Tools→DARTEL Tools→Normalise to MNI Space*:

- **Normalise to MNI Space**

  - **DARTEL Template**: Template_6.nii,1 is usually the grey matter component of the final template of the series. An affine transform is determined using this image.
  - **Select according to** either *Few Subjects* or *Many Subjects*. For fMRI analyses, the *Few Subjects* option would be selected, which gives the option of selecting a flow field and list of images for each subject.
    * **Subject**
      · **Flow Field**: Specify the flow field ("u_c1*.nii") for this subject.
      · **Images**: Select the images for this subject that are to be transformed to MNI space.
  - **Voxel sizes**: Specify the desired voxel sizes for the spatially normalised images (NaN, NaN, NaN gives the same voxel sizes as the DARTEL template).
  - **Bounding box**: Specify the desired bounding box for the spatially normalised images (NaN, NaN, NaN; NaN NaN NaN gives the same bounding box as the DARTEL template).
  - **Preserve**: Here you have a choice of *Preserve Concentrations* (ie not Jacobian scaled) or *Preserve Amount* (Jacobian scaled). The *Preserve Concentrations* option would normally be used for fMRI data, whereas *Preserve Amount* would be used for VBM.
  - **Gaussian FWHM**: Enter how much to blur the spatially normalised images, where the values denote the full width at half maximum of a Gaussian convolution kernel, in units of mm.

An alternative approach is now presented, which does not attempt to make optimal use of the available signal.

23.2.1 An alternative approach for using DARTEL to spatially normalise to MNI Space

During spatial normalisation of a brain image, some regions need to expanded and other regions need to contract in order to match the template. If some structure is excessively shrunk by DARTEL (because it has the freedom to estimate quite large deformations), then this will lead to a systematic reduction in the amount of BOLD signal being detected from that brain region. For this reason, the normalise to MNI space option would generally be preferred when working with functional data that is to be smoothed.

Affine transform of DARTEL template to MNI space

DARTEL works with images that are of average size. When DARTEL is used to generate an average shaped template (represented by a series of tissue probability maps) from a group of scans of various individuals, the result is of average size. Brains normalised to MNI space are slightly larger than average. In order to spatially normalise to MNI space, the deformation that maps from MNI space to the space of the group average is required. Because the MNI space was derived by affine registration of a number of subjects to a common coordinate system, in most cases it should be possible to achieve a reasonable match of the template generated by DARTEL using only an affine spatial normalisation. This can be achieved by matching the grey matter component of the template with a grey matter tissue probability map in MNI space. The spatial normalisation routine in SPM can be used to achieve this.

- **Normalise: Estimate**

    - **Data**

        * **Subject**

            · **Source Image**: Template_6.nii,1 is usually the grey matter component of the final template of the series.
            · **Source Weighting Image**: <None>

    - **Estimation Options**

        * **Template Image**: Should be the apriori/grey.nii file distributed in SPM.
        * **Template Weighting Image**: <None>
        * **Source Image Smoothing**: 8mm (the same as the apriori/grey.nii file has been smoothed).
        * **Template Image Smoothing**: 0mm (because the data in the apriori folder are already smoothed by 8mm.)
        * **Affine Regularisation**: Usually, you would specify "ICBM space template".
        * **Nonlinear Frequency Cutoff**: Set this to infinity (enter "Inf") for affine registration.
        * **Nonlinear Iterations**: Setting this to zero will also result in affine-only spatial normalisation.
        * **Nonlinear Regularisation**: Setting this to infinity is another way of doing affine-only spatial normalisation.

For some populations of subjects, an affine transform may not be adequate for achieving good registration of the average shape to MNI space. Nonlinear spatial normalisation may be more appropriate for these cases. As ever, determining which procedure is better would involve a degree of empirical exploration.

Combining deformations

Once you have the spatial transformation that maps from MNI space to the space of the DARTEL template, it is possible to combine this with the DEFORMATIONS estimated by DARTEL. Rather than warping the image data twice (introducing interpolation artifacts each time), the two spatial transforms can be combined by composing them together. The required deformation, for spatially normalising an individual to MNI space, is a mapping from MNI space to the individual image. This is because the spatially normalised images are generated by scanning through the (initially empty) voxels in the spatially normalised image, and figuring out which voxels in the original image to sample from (as opposed to scanning through the original image and putting the values into the right places in the spatially normalised version).

The desired mapping is from MNI space to DARTEL template to individual scan. If A is the mapping from MNI to template, and B is the mapping from template to individual, then this mapping is $B \circ A$, where "$\circ$" denotes the composition operation. Spatially normalising via the composed deformations can be achieved through the *Deformations* utility from the *TASKS* pull-down (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.

23.3 Warping Images to Existing Templates

If templates have already been created using DARTEL, then it is possible to align other images with such templates. The images would first be imported in order to generate rc1*.nii and rc2*.nii files. The procedure is relatively straight-forward, and requires the *SPM→Tools→DARTEL Tools→Run DARTEL (existing Template)* option to be specified. Generally, the procedure would begin by registering with a smoother template, and end with a sharper one, with various intermediate templates between.

- **Run DARTEL (existing Templates)**
  - **Images**
    * **Images**: Select the rc1*.nii files.
    * **Images**: Select the corresponding rc2*.nii files.
  - **Settings**: Most settings would be kept at the default values, except for the specification of the templates. These are specified in within each of the *Settings→Outer Iterations→Outer Iteration→Template* fields. If the templates are Template_*.nii, then enter them in the order of Template_1.nii, Template_2.nii, ... Template_6.nii.

Running this option is rather faster than *Run DARTEL (create Template)*, as templates are not created. The output is in the form of a series of flow fields (u_rc1*.nii).

23.4   Warping one individual to match another

Sometimes the aim is to deform an image of one subject to match the shape of another. This can be achieved by running DARTEL so that both images are matched with a common template, and composing the resulting spatial transformations. This can be achieved by aligning them both with a pre-existing template, but it is also possible to use the *Run DARTEL (create Template)* option with the imported data of only two subjects. Once the flow fields (u_rc1*.nii files) have been estimated, then the resulting deformations can be composed using *SPM→Utils→Deformations*. If the objective is to warp A.nii to align with B.nii, then the procedure is set up by:

- **Deformations**

    - **Composition**

        * **DARTEL flow**
            · **Flow field**: Specify the u_rc1A_Template.nii flow field.
            · **Forward/Backwards**: Backward.
            · **Time Steps**: Usually 64.
        * **DARTEL 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 labelled, then this option is useful for transferring the labels on to images of other subjects.
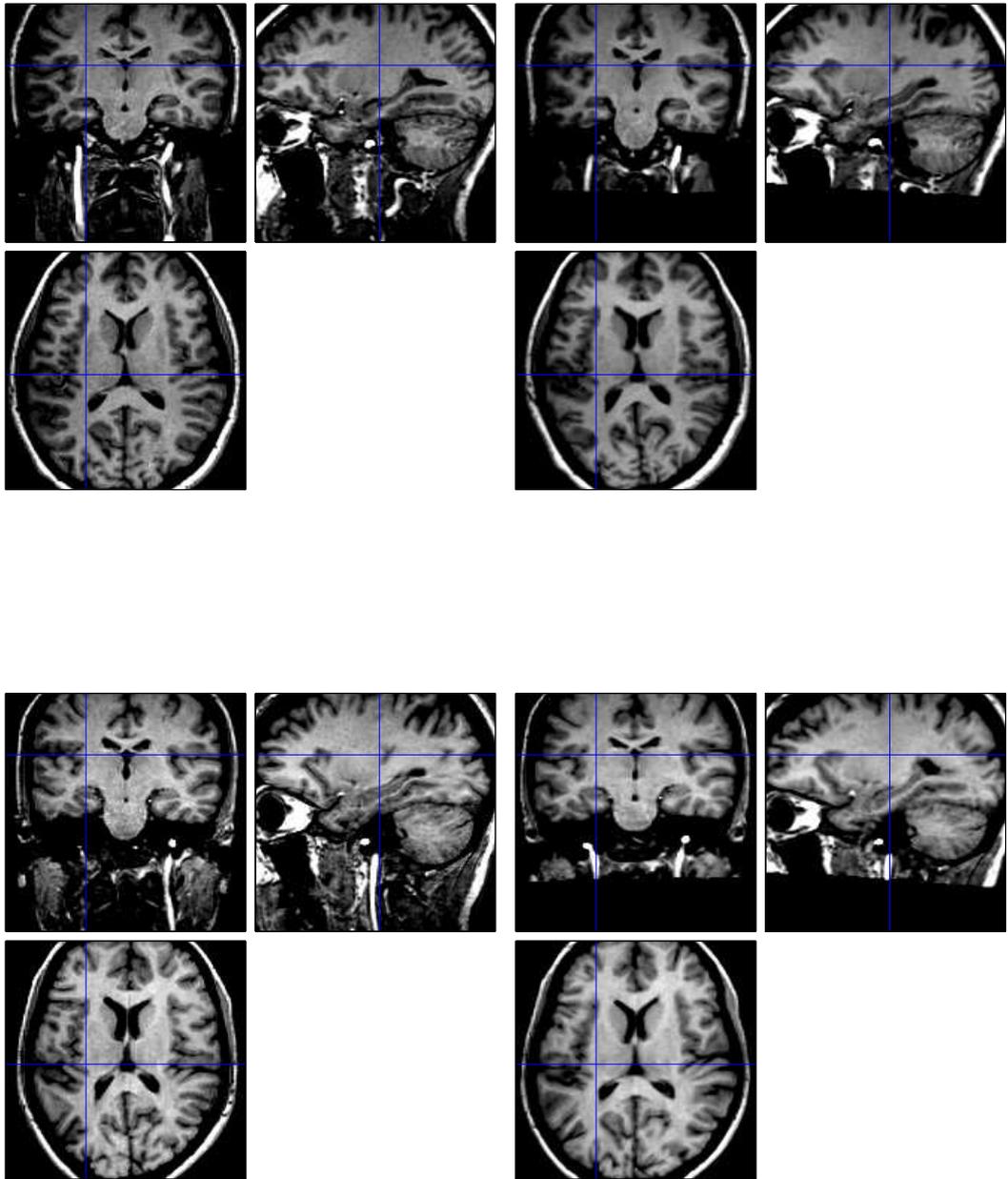
Figure 23.4: Composition of deformations to warp one individual to match another. Top-left: Original A.nii. Top-right: A.nii warped to match B.nii. Bottom-left: Original B.nii. Bottom-right: B.nii warped to match A.nii.

Part VIII

Batch Interface

Chapter 24

Batch tutorial

Details about the algorithms used for data processing are given in the other sections of this manual. This section explains how a sequence of processing steps can be run at once without MATLAB programming. SPM8 includes `matlabbatch`[1] 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.

24.1   Single subject

In this tutorial we will develop a batch for spatial processing and fMRI statistics of a single subject of the "Face" example dataset (see chapter 11). 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.

---

[1] http://sourceforge.net/projects/matlabbatch

24.1.1   Study specific input data

This dataset consists of fMRI data acquired in a single session and a structural MRI. See section 24.2 to learn how to deal efficiently with multi-session data. MRI parameters and experiment details are described in chapter 11.

24.1.2   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

24.1.3   Add modules to the batch

The helper modules and the SPM processing modules can be assembled using the GUI. Click the "BATCH" button in the SPM Menu window. First, add the helper modules, followed by the SPM modules in the order listed above. Do not configure any details until you have selected all modules.

24.1.4   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 11. 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 11.

The file `man/batch/face_single_subject_template_nodeps.m` contains the batch after all modules have been added and subject-independent data has been entered.
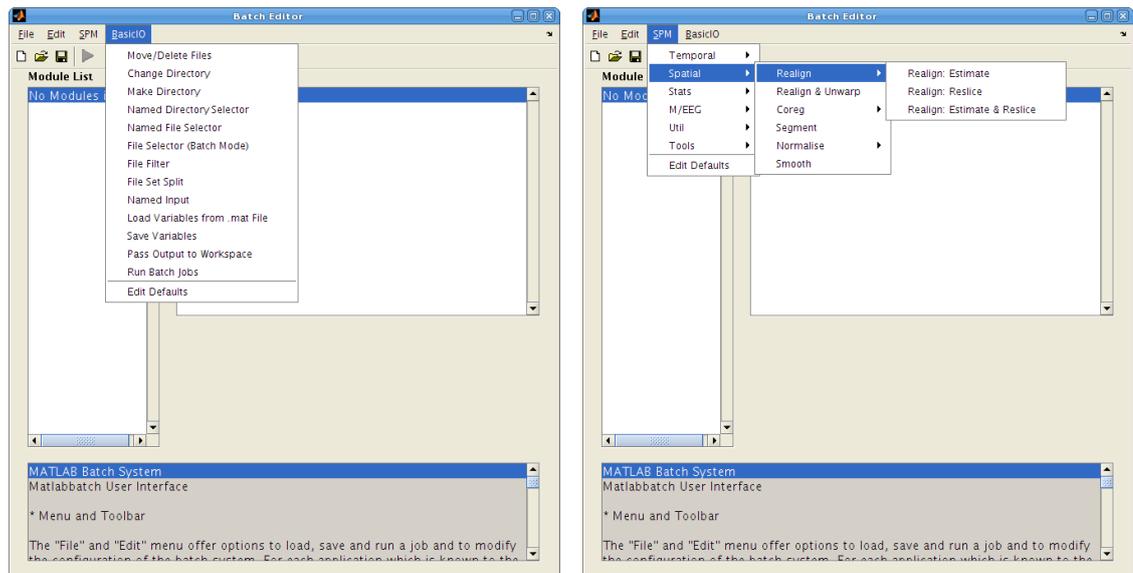
Figure 24.1: BasicIO and SPM application menus

Named Directory Selector

Input Name Give this selection a name (e.g. "Subject directory") - this name will be shown in the dependency list of this batch.

Directories Add a new directory selector, but do not enter a directory itself.

Change Directory

Nothing to enter now.

Make Directory

New Directory Name "categorical" - the name of the analysis directory. This directory will be created at batch run-time in the subject directory.

Realign: Estimate & Reslice

Data Add a new "Session" item. Do not enter any files for this session now.

Slice Timing

Data Add a new "Session" item. Do not enter any files for this session now.

Timing options Enter data for "Number of slices", "TR", "TA", "Slice order", "Reference slice".

Coreg: Estimate

Nothing to enter now.

Segment

Nothing to enter now.

Normalise: Write

Data Add a new "Subject". Do not enter any files now.

Writing Options Adjust bounding box, voxel sizes, interpolation

Smooth

FWHM Enter FWHM

fMRI model specification

Enter all data which is constant across subjects.

Timing parameters Enter values for "Units for design", "Interscan interval", "Microtime resolution", "Microtime onset"

Data & Design Add a new "Session" item. Do not enter scans, conditions or regressors yet. They will be added as dependencies or subject specific inputs. If you want to make sure to remember this, you can highlight "Multiple conditions" and select "Clear Value" from the "Edit" menu. Do the same for "Multiple regressors". This will mark both items with an `<-X`, indicating that something must be entered there.

Factorial design Enter the specification for both factors.

Basis functions Select the basis function and options you want to use.

Model estimation

Nothing to be entered yet for classical estimation.

Contrast manager

If you have selected the "Factorial design" option as described above, SPM will automatically create some contrasts for you. Here, you can create additional T- or F-contrasts. As an example, we will add an "Effects of interest" F-contrast.

Contrast session Add a new "F-contrast" item.

Name Enter a name for this contrast, e.g. "Effects of interest".

Contrast vectors Add a new "Contrast vector" item. F-contrasts can have multiple rows. You can either enter a contrast matrix in an "F contrast vector" entry, or enter them row by row. To test for the effects of interest (1 basis function and 2 derivatives for each of the four conditions) enter `eye(12)` as F contrast vector.

Replicate over sessions This design does not have multiple sessions, so it is safe to say "No" here.

Results report

Reviewing individual results for a large number of subjects can be very time consuming. Results report will print results from selected contrasts to a PostScript file.

Contrast(s) Enter `Inf` to print a report for each of the defined contrasts.

24.1.5   Data flow

In chapter 11, 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 24.2. Only the coloured items at the top of the flow chart are subject specific and need to be entered in the final batch. All arrow connections are subject-independent and can be specified in the batch template.
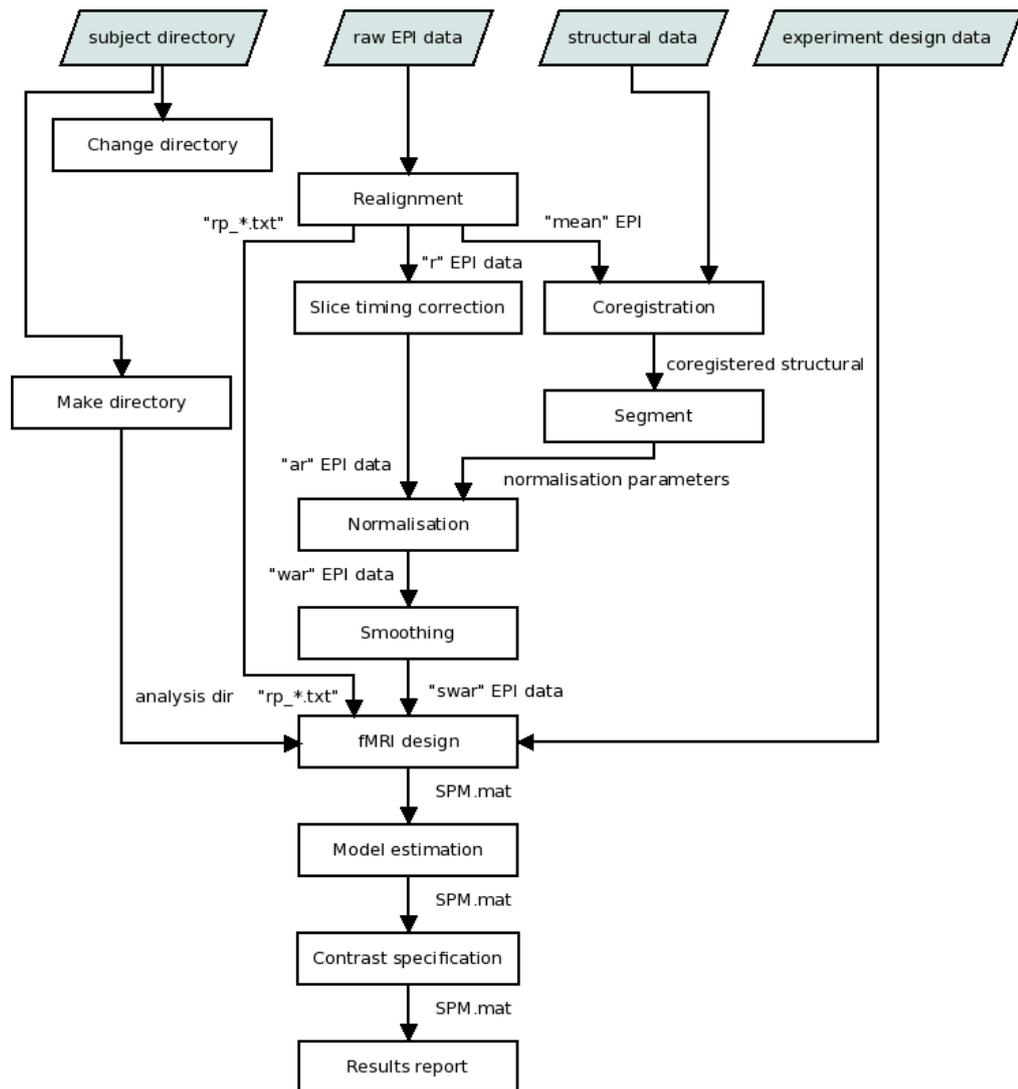
Figure 24.2: Flow chart for batch

Add dependencies

Based on the data flow in figure 24.2, modules in the batch can now be connected. The batch containing all dependencies can be found in `man/batch/face_single_subject_template.m`.
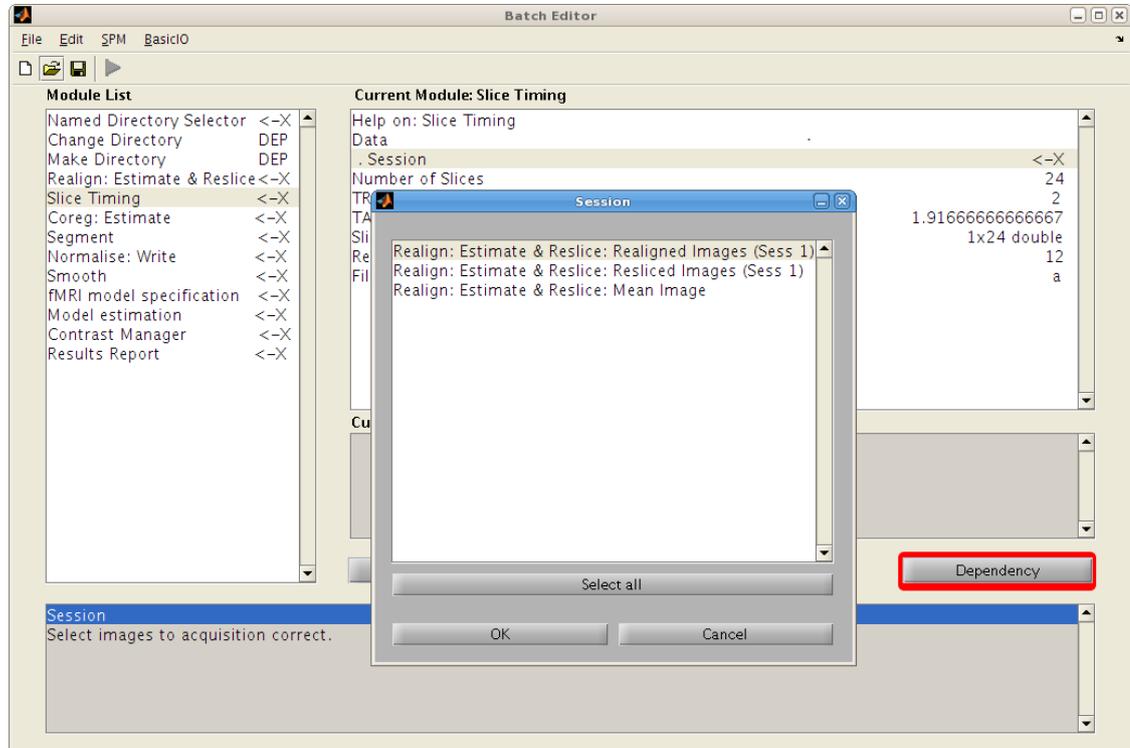

Figure 24.3: Dependency selection

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)"

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 24.2 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)".

24.1.6 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 24.2.3 for details.
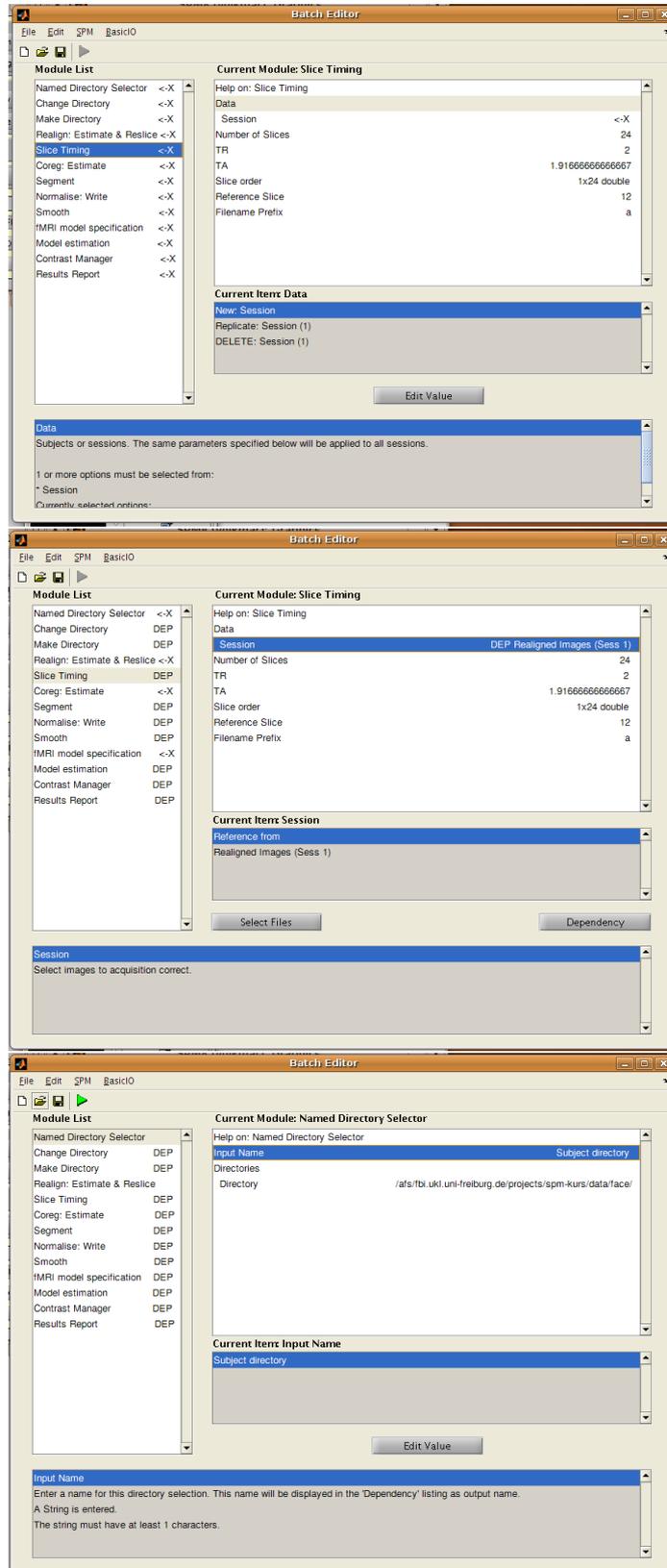
Figure 24.4: All stages of batch entry

24.2 Advanced features

24.2.1 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.

24.2.2 Processing multiple subjects in GUI

There are different ways to process multiple subjects in the batch editor:

- Add the necessary processing steps when creating the job.

- Create a per-subject template, save it and load it multiple times (i.e. in the file selector, add the same file multiple times to the list of selected files).

- Use "Run Batch Jobs" from "BasicIO"
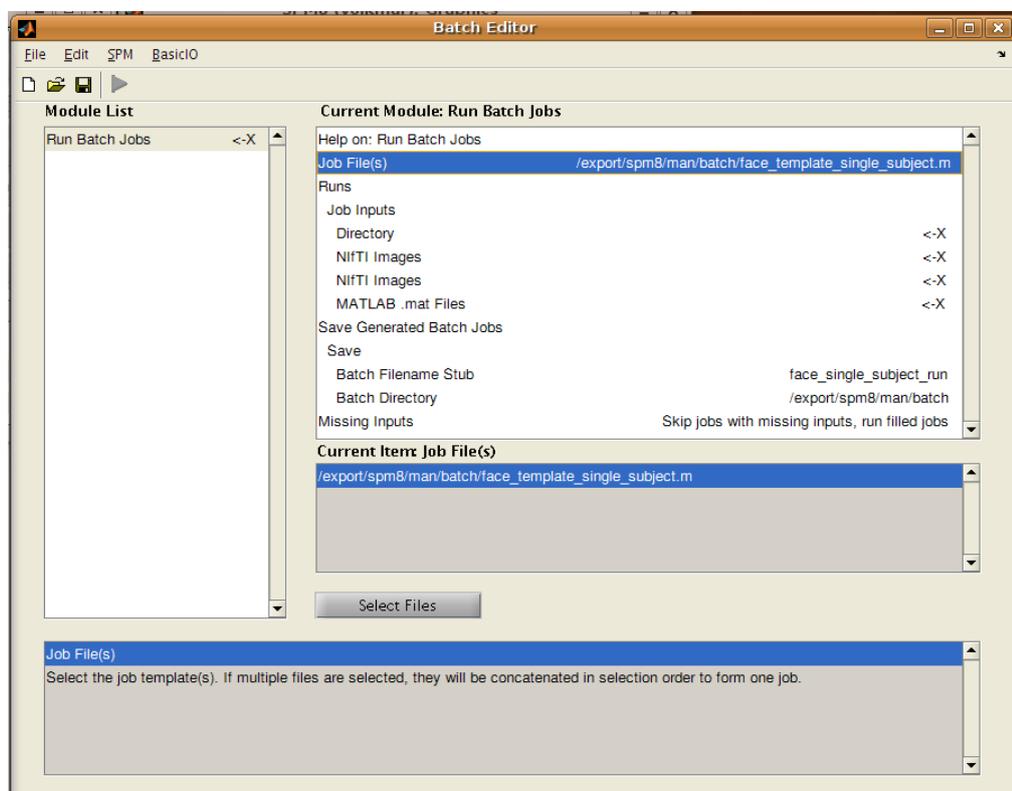
Figure 24.5: Using "Run Batch Jobs"

In all cases, the data for all subjects has to be entered through the GUI, and computation will be done for all subjects at once after all data is entered. There is an example job `face_multi_subject_template.m` that demonstrates the usage of "Run Batch Jobs" to run the

single subject template job described above. Note that the order and type of inputs in the single subject template is important. Also, consistency checks are limited. If inconsistent data is entered, the job will fail to execute and return an error message.

To run this job for multiple subjects, simply repeat the "Runs" item as many times as necessary and fill in the required data.

24.2.3   Command line interface

The command line interface is especially useful to run multiple jobs at once without user interaction, e.g. to process multiple subjects or to combine separate processing steps. There is a "high-level" interface using `spm_jobman`, which combines "low-level" callbacks to `cfg_util`.

SPM startup in command line mode

During normal startup, SPM performs important initialisation steps. Without initialisation, SPM and its batch system will not function properly. Consequently, an initialisation sequence needs to be run before any batch job can be submitted.

MATLAB has several command line options to start without its GUI (`-nodesktop`) or even without any graphics output to a screen (`-nodisplay`). See MATLAB documentation for details.

To run SPM in `-nodisplay` mode, the file `spm_defaults.m` has to be modified. The line `defaults.cmdline = 0;` must be changed to `defaults.cmdline = true;`. In command line mode, SPM will not open any figure window except the "Graphics" window.

Within MATLAB, the following commands are sufficient to set up SPM

1. `spm('defaults', MODALITY)` where `MODALITY` has to be replaced by the desired modality (e.g. `'fmri'`)

2. `spm_jobman('initcfg')`

After executing these commands, any SPM functions and batch jobs can be run in the same MATLAB session.

Complete and run a pre-specified job

`spm_jobman('serial', 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. Note that only items without a pre-set value will be filled (marked with `<-X` in the GUI). To force a item to to be filled, use "Edit:Clear Value" in the GUI or set its value to `'<UNDEFINED>'` in the harvested job.

The job argument is very flexible, it can e.g. be a job variable, the name of a script creating a job variable, even a cell list of any mixture of variables and scripts. All job snippets found will be concatenated into a single job, the missing inputs will be filled and the resulting job will be run.

The batch system can generate a script skeleton for any loaded job. From the batch GUI, this feature is accessible via "File:Save Batch and Script". This skeleton consists of a commented list of necessary inputs, a `for` loop to enter inputs for multiple runs or subjects and the code to initialise and run the job. An example is available in `face_single_subject_script.m`:

```
% List of open inputs
% Named Directory Selector: Directory - cfg_files
% Realign: Estimate & Reslice: Session - cfg_files
% Coreg: Estimate: Source Image - cfg_files
% fMRI model specification: Multiple conditions - cfg_files
nrun = X; % enter the number of runs here
jobfile = {fullfile(spm('dir'),'man','batch','face_single_subject_template.m')};
jobs = repmat(jobfile, 1, nrun);
inputs = cell(4, nrun);
for crun = 1:nrun
```

```
    % Named Directory Selector: Directory - cfg_files
    inputs{1, crun} = MATLAB_CODE_TO_FILL_INPUT;
    % Realign: Estimate & Reslice: Session - cfg_files
    inputs{2, crun} = MATLAB_CODE_TO_FILL_INPUT;
    % Coreg: Estimate: Source Image - cfg_files
    inputs{3, crun} = MATLAB_CODE_TO_FILL_INPUT;
    % fMRI model specification: Multiple conditions - cfg_files
    inputs{4, crun} = MATLAB_CODE_TO_FILL_INPUT;
end
spm('defaults','fmri');
spm_jobman('serial',jobs,'',inputs{:});
```

The skeleton needs to be adapted to the actual data layout by adding MATLAB code which specifies the number of runs and the input data in the `for` loop.

Another example script and batch is available for the multimodal dataset, called `multimodal_fmri_script.m` and `multimodal_fmri_template.m`.

24.2.4 Modifying a saved job

In some cases, instead of using the serial interface it may be more appropriate to modify the fields of a saved or harvested job. By default, jobs are saved as MATLAB `.mat` files, but they can also be saved as `.m` files. These files contain a number of MATLAB commands, which will create a variable `matlabbatch`. The commands can be modified to set different values, add or remove options.

Chapter 25

Developer's guide

25.1 SPM5 to Matlabbatch transition guide

This is a short overview to describe code organisation and interfaces between SPM and the batch system.

25.1.1 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.

25.1.2 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).

25.2   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.

25.2.1   Virtual Outputs

Virtual outputs are described by arrays of `cfg_dep` objects. These objects contain a "source" and a "target" part. Functions may have more than one virtual output (e.g. one output per session, a collection of results variables). One `cfg_dep` object has to be created for each output.

Only two fields in the "source" part need to be set in a `.vout` callback:

sname A display name for this output. This will appear in the dependencies list and should describe the contents of this dependency.

src_output A subscript reference that can be used to address this output in the variable returned at run-time.

tgt_spec (optional) A description on what kind of inputs this output should be displayed as dependency. This is not very convenient yet, the `match` and `cfg_findspec` methods are very restrictive in the kind of expressions that are allowed.

The `.vout` callback will be evaluated once the configuration system thinks that enough information about the *structure* of the outputs is available. This condition is met, once all in-tree nodes `cfg_(ex)branch`, `cfg_choice`, `cfg_repeat` have the required number of child nodes.

The `.vout` callback is called with a job structure as input, but its code *should not rely* on the evaluation of any contents of this structure (or at least provide a fallback). The contents of the leaf nodes may not be set or may contain a dependency object instead of a value during evalution of `.vout`.

The "target" part will be filled by the configuration classes, the `src_exbranch` field is set in `cfg_util`.

25.2.2   SPM Startup

The top level configuration file for SPM is `spm_cfg.m`. It collects SPM core configuration files and does toolbox autodetection. If a toolbox directory contains `*_cfg_*.m` files, they will be loaded. Otherwise, if there are only SPM5-style `*_config_*.m` files, the configuration will be converted at run-time using `cfg_struct2cfg`.

25.2.3   Defaults Settings

In Matlabbatch, there are different ways to set defaults:

1. in the configuration file itself,

2. in a defaults file, which has a structure similar to a harvested job,

3. using a `.def` field for leaf items.

Defaults set using option 1 or 2 will only be updated at SPM/matlabbatch startup. Defaults set using option 3 will be set once a new job is started. These defaults take precedence over the other defaults.

In core SPM, these defaults refer to `spm_get_defaults`, which accesses `spm_defaults`. Toolboxes may use their own callback functions.

25.2.4 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.

Toolboxes should set their defaults using the `.def` fields, using a mechanism similar to `spm_get_defaults`. This allows for flexibility without interfering with SPMs own defaults.

25.3 Utilities

25.3.1 Batch Utilities

Matlabbatch is designed to support multiple applications. A standard application "BasicIO" is enabled by default. Among other options, it contains file/file selection manipulation utilities which can be used as 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_confgui` 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.

25.3.2 MATLAB Code Generation

The `gencode` utility generates MATLAB `.m` file code for any kind of MATLAB variable. This is used to save batch files as well as to generate configuration code.

25.3.3 Configuration Management

The backend utility to manage the configuration data is `cfg_util`. It provides callbacks to add application configurations, and to load, modify, save or run jobs. These callbacks are used by two frontends: `cfg_ui` is a MATLAB GUI, while `cfg_serial` can be used both as a GUI and in script mode. In script mode, it will fill in job inputs from an argument list. This allows to run predefined jobs with e.g. subject dependent inputs without knowing the exact details of the job structure.

Part IX

Bibliography

Bibliography

[1] J. Andersson, J. Ashburner, and K.J. Friston. A global estimator unbiased by local changes. *NeuroImage*, 13(6):1193–1206, 2001.

[2] J. Andersson, C. Hutton, J. Ashburner, R. Turner, and K.J. Friston. Modelling geometric deformations in EPI time series. *NeuroImage*, 13(5):903–919, 2001.

[3] S. Baillet, J.C. Mosher, and R.M. Leahy. Electromagnetic brain mapping. *IEEE Sign. Proc. Mag.*, 18:14–30, 2001.

[4] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pat. Anal. and Mach. Intel.*, 142:239–256, 1992.

[5] 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.

[6] 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.

[7] C. Buchel, O. Josephs, G. Rees, R. Turner, and C. FrithK.J. Friston. The functional anatomy of attention to visual motion. a functional mri study. *Brain*, 121:1281–1294, 1998.

[8] C. Buchel, R.J.S. Wise, C.J. Mummery, J.B. Poline, and K.J. Friston. Nonlinear regression in parametric activation studies. *NeuroImage*, 4:60–66, 1996.

[9] C.C. Chen, R.N.A. Henson, K.E. Stephan, J. Kilner, and K.J. Friston. Forward and backward connections in the brain: A dcm study of functional asymmetries. *NeuroImage*, 45(2):453–462, 2009.

[10] C.C. Chen, S.J. Kiebel, and K.J. Friston. Dynamic causal modelling of induced responses. *NeuroImage*, 41(4):1293–1312, 2008.

[11] 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.

[12] J. Daunizeau, S.J. Kiebel, and K.J. Friston. Dynamic causal modelling of distributed electromagnetic responses. *NeuroImage*, 2009.

[13] 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.

[14] Matthias Fastenrath, K.J. Friston, and S.J. Kiebel. Dynamical causal modelling for m/eeg: Spatial and temporal symmetry constraints. *NeuroImage*, 2008.

[15] K.J. Friston. Functional and effective connectivity in neuroimaging: A synthesis. *Human Brain Mapping*, 2:56–78, 1995.

[16] K.J. Friston, J. Ashburner, S.J. Kiebel, T.E. Nichols, and W.D. Penny, editors. *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. Academic Press, 2007.

[17] K.J. Friston, C. Buchel, G.R. Fink, J. Morris, E. Rolls, and R. Dolan. Psychophysiological and modulatory interactions in neuroimaging. *NeuroImage*, 6:218–229, 1997.

[18] 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.

[19] K.J. Friston, L. Harrison, J. Daunizeau, S.J. Kiebel, C. Phillips, N. Trujillo-Bareto, R.N.A. Henson, G. Flandin, and J. Mattout. Multiple sparse priors for the m/eeg inverse problem. *NeuroImage*, 39(3):1104–1120, 2008.

[20] K.J. Friston, L. Harrison, and W.D. Penny. Dynamic Causal Modelling. *NeuroImage*, 19(4):1273–1302, 2003.

[21] 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.

[22] 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.

[23] K.J. Friston and W.D. Penny. Posterior probability maps and SPMs. *NeuroImage*, 19(3):1240–1249, 2003.

[24] K.J. Friston, W.D. Penny, and D.E. Glaser. Conjunction revisited. *NeuroImage*, 25(3):661–667, 2005.

[25] 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.

[26] M.I. Garrido, K.J. Friston, K.E. Stephan, S.J. Kiebel, T. Baldeweg, and J. Kilner. The functional anatomy of the MMN: A DCM study of the roving paradigm. *NeuroImage*, 42(2):936–944, 2008.

[27] M.I. Garrido, J. Kilner, S.J. Kiebel, and K.J. Friston. Evoked brain responses are generated by feedback loops. *PNAS*, 104(52):20961–20966, 2007.

[28] M.I. Garrido, J. Kilner, S.J. Kiebel, K.E. Stephan, and K.J. Friston. Dynamic causal modelling of evoked potentials: A reproducibility study. *NeuroImage*, 36:571–580, 2007.

[29] D.R. Gitelman, W.D. Penny, J. Ashburner, and K.J. Friston. Modeling regional and psychophysiologic interactions in fMRI: the importance of hemodynamic deconvolution. *NeuroImage*, 19:200–207, 2003.

[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, E. Mouchlianitis, and K.J. Friston. Meg and eeg data fusion: Simultaneous localisation of face-evoked responses. *NeuroImage*, 2009.

[33] R.N.A. Henson and W.D. Penny. ANOVAs and SPM. Technical report, Wellcome Department of Imaging Neuroscience, 2003.

[34] R.N.A. Henson, C.J. Price, M.D. Rugg, R. Turner, and K.J. Friston. Detecting latency differences in event-related bold responses: application to words versus non-words and initial versus repeated face presentations. *NeuroImage*, 15(1):83–97, 2002.

[35] R.N.A. Henson, M.D. Rugg, and K.J. Friston. The choice of basis functions in event-related fMRI. *NeuroImage*, 13(6):149, June 2001. Supplement 1.

[36] 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.

[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, J. Daunizeau, C. Phillips, and K.J. Friston. Variational bayesian inversion of the equivalent current dipole model in eeg/meg. *NeuroImage*, 39(2):728–741, 2008.

[43] 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.

[44] S.J. Kiebel, M.I. Garrido, and K.J. Friston. Dynamic causal modelling of evoked responses: The role of intrinsic connections. *NeuroImage*, 36:332–345, 2007.

[45] S.J. Kiebel, M.I. Garrido, R. Moran, and K.J. Friston. Dynamic causal modelling for eeg and meg. *Cognitive Neurodynamics*, 2(2):121–136, 2008.

[46] S.J. Kiebel, S. Klppel, N. Weiskopf, and K.J. Friston. Dynamic causal modeling: A generative model of slice timing in fMRI. *NeuroImage*, 34:1487–1496, 2007.

[47] A. P. Leff, T. M. Schofield, K. E. Stephan, J. T. Crinion, K. J. Friston, and C. J. Price. The cortical dynamics of intelligible speech. *J. Neurosci.*, 28:13209–13215, Dec 2008.

[48] V. Litvak and K.J. Friston. Electromagnetic source reconstruction for group studies. *NeuroImage*, 42(4):1490–1498, 2008.

[49] A.C. Marreiros, J. Daunizeau, S.J. Kiebel, and K.J. Friston. Population dynamics: Variance and the sigmoid activation function. *NeuroImage*, 42(1):147–157, 2008.

[50] J. Mattout, C. Phillips, W.D. Penny, M. Rugg, and K.J. Friston. Meg source localization under multiple constraints: an extended Bayesian framework. *NeuroImage*, 30(3):753–767, 2006.

[51] R. Moran, S.J. Kiebel, N. Rombach, W.T. O'Connor, K.J. Murphy, R.B. Reilly, and K.J. Friston. Bayesian estimation of synaptic physiology from the spectral responses of neural masses. *NeuroImage*, 42(1):272–284, 2008.

[52] R. Moran, S.J. Kiebel, K.E. Stephan, R.B. Reilly, J. Daunizeau, and K.J. Friston. A neural mass model of spectral responses in electrophysiology. *NeuroImage*, 37(3):706–720, 2007.

[53] R. Moran, K.E. Stephan, F. Jung, R. Graf, R. Dolan, K.J. Friston, and M. Tittgemeyer. Dcm for steady state responses: a case study of anaesthesia dept. In *Proc Soc for Neuroscience*, 2009.

[54] R. Moran, K.E. Stephan, T. Seidenbecher, H.C. Pape, R. Dolan, and K.J. Friston. Dynamic causal models of steady-state responses. *NeuroImage*, 44(3):796–811, 2009.

[55] 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.

[56] W.D. Penny and G. Flandin. Bayesian analysis of single-subject fMRI: SPM implementation. Technical report, Wellcome Department of Imaging Neuroscience, 2005.

[57] W.D. Penny, G. Flandin, and N. Trujillo-Bareto. Bayesian Comparison of Spatially Regularised General Linear Models. *Human Brain Mapping*, 28:275–293, 2005.

[58] 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.

[59] W.D. Penny, S.J. Kiebel, and K.J. Friston. Variational Bayesian Inference for fMRI time series. *NeuroImage*, 19(3):727–741, 2003.

[60] W.D. Penny, V. Litvak, L. Fuentemilla, E. Duzel, and K.J. Friston. Dynamic Causal Models for phase coupling. *Journal of Neuroscience Methods*, 183(1):19–30, 2009.

[61] W.D. Penny, K.E. Stephan, J. Daunizeau, M.J. Rosa, K.J. Friston, T.M.Schofield, and A.P. Leff. Comparing families of dynamic causal models. *PLoS Comput Biol*, 6(3):e1000709, 2010.

[62] W.D. Penny, K.E. Stephan, A. Mechelli, and K.J. Friston. Comparing Dynamic Causal Models. *NeuroImage*, 22(3):1157–1172, 2004.

[63] W.D. Penny, N. Trujillo-Bareto, and K.J. Friston. Bayesian fMRI time series analysis with spatial priors. *NeuroImage*, 24(2):350–362, 2005.

[64] 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.

[65] K.E. Stephan, W.D. Penny, J. Daunizeau, R. Moran, and K.J. Friston. Bayesian model selection for group studies. *NeuroImage*, 46(3):1004–10174, 2009.

[66] P. Thévenaz, T. Blu, and M. Unser. Interpolation revisited. *IEEE Transactions on Medical Imaging*, 19(7):739–758, 2000.

[67] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing: Part I – theory. *IEEE Transactions on Signal Processing*, 41(2):821–833, 1993.

[68] 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.