Dynamic Causal Modelling of M/EEG data in SPM12

Pranay Yadav[1]* & Richard N Henson[1,2]

[1]MRC Cognition & Brain Sciences Unit, University of Cambridge, Cambridge, UK

[2] Department of Psychiatry, University of Cambridge, Cambridge, UK

*** Correspondence:**
Corresponding Author
pranay.yadav@mrc-cbu.cam.ac.uk

Keywords: MEG, EEG, fMRI, multimodal, fusion, SPM, inversion, faces

Abstract

Dynamic Causal Modelling (DCM) is a widely used method for inferring effective connectivity from various kinds of neuroimaging data. This tutorial demonstrates a step-by-step walkthrough for using DCM to investigate group-level effective connectivity from a publicly available open-access MEEG dataset for face processing from 16 subjects. We illustrate a reproducible analysis pipeline that makes use of a hierarchical Bayesian framework called Parametric Empirical Bayes (PEB) to characterize inter-individual variability in neural circuitry. At the group level, we show various approaches for performing testing focused hypotheses on the estimated connectivity using Bayesian model comparison.

1. Introduction

This paper is a tutorial for Dynamic Causal Modelling (DCM) of fMRI data. DCM is a method for inferring effective connectivity between brain regions from neuroimaging data, and is part of the Statistical Parametric Mapping (SPM) free academic software (https://www.fil.ion.ucl.ac.uk/spm/), which is implemented in Matlab (The MathWorks Inc., 2018). We focus on basic DCM specification, estimation, Bayesian model reduction and Bayesian model comparison, using the recent Parametric Empirical Bayesian (PEB) framework for group-level inference across multiple subjects. Together with its companion document describing DCM of fMRI data, this document extends the DCM PEB tutorial by Zeidman and colleagues (Zeidman, Jafarian, Corbin, et al., 2019; Zeidman, Jafarian, Seghier, et al., 2019; https://github.com/pzeidman/dcm-peb-example) to MEG/EEG data, using a different multimodal dataset and illustrating other features of DCM and PEB.

The dataset contains fMRI and MEG+EEG data on 16 subjects from a face-processing paradigm described in Wakeman & Henson, 2015. The raw data in BIDS format are available on OpenNeuro (https://openneuro.org/datasets/ds000117).[1] This tutorial continues a previous tutorial on the same dataset (Henson et al., 2019), which illustrated basic pre-processing and source localisation of MEG/EEG data in SPM. Here we assume this pre-processing has already been done, though you can download the pre-processed data as described below.

We describe practical steps using SPM's graphical user interface (GUI), its "batch" interface for linear pipeline creation and finally "scripting" in MATLAB for (parallelised) loops across subjects. We use version 12.5 of DCM in version 12 of SPM. The paper is organised into sections with a brief theoretical background followed by a detailed step-by-step walkthrough. The background is only brief because we refer to previous published papers, many of which are available from the SPM website: https://www.fil.ion.ucl.ac.uk/spm/doc/biblio/. We do not provide a full tour of all the available options in SPM for M/EEG, which is already present in Litvak et al. (2011). Rather, we focus on the typical steps for group-level DCM inference using PEB. Our experience with teaching SPM is that students appreciate having a concrete example, which they can then adjust to their own needs.

The steps below are also scripted in the "code" directory that you can download or clone from https://github.com/pranaysy/cognestic22_multimodal_dcm. This contains two sub-directories, one for fMRI and one for MEG, which themselves contain two groups of files: one of these are MATLAB files derived from SPM's "batch" interface (filenames beginning with batch*), in which various analysis steps (batch "modules") were created by the GUI, saved and then called from loops across subjects (all within the "spm_master_script_dcm_*_peb_batch.m"); the other consists of a script that implements exactly the same analyses, but with direct calls to underlying "spm*.m" functions, bypassing SPM's Batch interface (e.g., contained within the script in "spm_master_script_dcm_*_peb_direct.m").

2. The Multimodal Dataset

The dataset comes from a paradigm in which participants saw a series of faces and phase-scrambled faces, and made left-right symmetry judgments to each stimulus. There were 300 unique faces and 150 unique scrambled images. Half of the faces were famous and half non-famous, but we ignore this distinction in this tutorial. Each stimulus was presented for 900ms on average, followed by 2200ms on average. Each stimulus was repeated either immediately, or after 5-15 intervening stimuli, but again we ignore the effects of repetition here. Thus we only analyse two conditions: faces vs scrambled faces. See Lee et al. (2022) for DCM analysis of effects of repetition and recognition (familiar vs unfamiliar) in the fMRI data.

[1] There are 16 subjects on OpenNeuro, and we use MEEG data from all here but we have excluded one from the fMRI tutorial due to missing data in one run.

To give participants a break, the fMRI experiment was split into 9 runs, with approximately equal numbers of each condition per run, though to avoid delayed repetition across runs, a small number of these trials were dropped. In addition, in order to estimate the response versus inter-stimulus baseline, six periods of 20s of a fixation cross were added after a block of 9-20 trials. The MEG experiment was split into 6 runs, and there was no need for interspersing longer periods of fixation as in the fMRI experiment. For more details, see Wakeman & Henson (2015).

3. Background

3.1 DCM

DCM is a tool for inferring "effective connectivity" between brain regions of interest (ROIs), based on explicit network models and assumptions about neural dynamics. DCM is a state-space model, consisting of 1) first-order differential equations that relate changes in each latent neural variable to other variables in the network (depending on connections), and 2) an observation model that maps neural variables to the measured fMRI and M/EEG signals.

For fMRI, the neural model is based on simple exponential decay of activity within each area, offset by input from other areas, as captured by a simple bilinear approximation. For MEG, the neural model is much more complex (e.g, containing multiple differential equations relating many parameters within each ROI, based on knowledge of the neurophysiology of human cortical layers, with several variants; see Section 5). For more background on DCM, see David et al. (2006), Moran et al. (2013) and Pereira et al. (2021).

For fMRI, the observation model is a temporal model that maps brief changes in neural activity to the more dispersed BOLD impulse response (a so-called HaemoDynamic Model, HDM). For MEG, the observation model is a spatial model that maps certain neural variables to electrical/magnetic fields recorded by sensors outside the head.

3.2 Current Network (model)

Here we focus on 4 ROIs: left and right occipital face area (OFA) and fusiform face area (FFA). These are the main four peaks that survive correction for multiple comparisons in the group analysis of the contrast of faces versus scrambled faces (see Supplementary Figure A2.1 of Henson et al. 2019; https://www.frontiersin.org/articles/10.3389/fnins.2019.00300/full#supplementary-material); the left FFA appears if the cluster threshold is reduced from 30 to 10 or fewer voxels. We could define the ROIs by saving each of them as thresholded clusters (as in Section 3.2 of Henson et al., 2019), but here for MEG/EEG, we use the coordinates of significant peaks as the prior location of an equivalent current dipole (ECD) (see Section 4.2.1.3).

We connect the four ROIs as shown in Figure 1. This assumes bidirectional connections between OFA and FFA within each hemisphere, bidirectional connections between hemispheres for both OFA and FFA, but no direct connections between OFA and FFA in different hemispheres (based on neuroanatomical principles). Together with each ROI's (inhibitory) self-connection, these are the "A" connections in DCM. We further assume that all of these fixed connections can be modulated by the presence of faces (vs scrambled faces). Finally, we will assume that the input (for all stimuli) enters left and right OFA (but see Lee et al., 2022, for more nuanced treatment of possible inputs to the network).

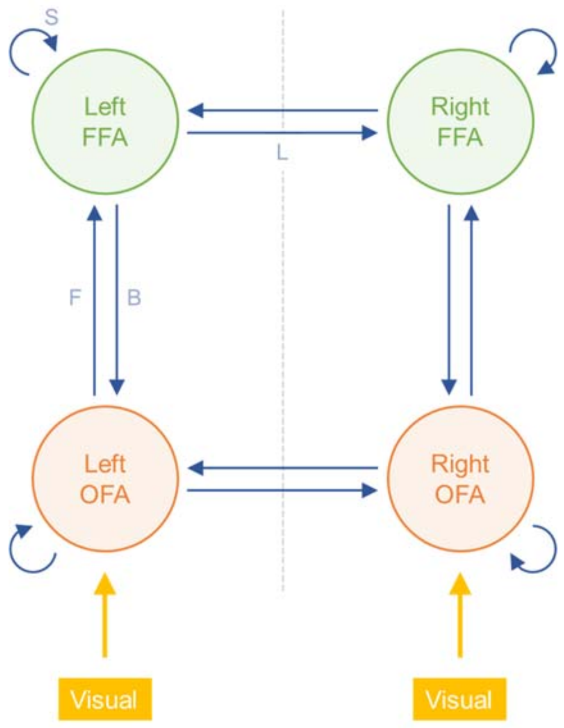


Figure 1. The 4-node DCM used for fMRI and MEG/EEG. F = forward, B = backward, S = self, L = lateral (note F, B and L only have different properties in DCM for MEG/EEG). OFA = Occipital Face Area; FFA = Fusiform Face Area.

DCM proceeds by comparing different models of the data through (an approximate lower bound on) the Bayesian model evidence, where models typically differ in connections (parameters), e.g. which connections are modulated by an experimental manipulation (here, by faces). When there are multiple subjects, one can create a single hierarchical model, enabling an Empirical Bayesian approach in which the mean and covariance of parameters across subjects can act as a prior on individual subject parameter values. Since DCM also assumes multivariate normal error terms (so-called "parametric" assumptions), this approach is called Parametric Empirical Bayes (PEB); see papers by Zeidman and colleagues (Zeidman, Jafarian, Corbin, et al., 2019; Zeidman, Jafarian, Seghier, et al., 2019) for more details.

There are many models one could compare. For example, one could ask whether face processing modulates connections between ROIs, or whether it is sufficient to explain face-related responses in each ROI simply via modulations of each ROI's self-connections. If the latter "self-only modulation" model were more parsimonious (had higher model evidence), then there would be no need to assume that faces change the effective connectivity between regions (and the traditional voxel-wise analysis of univariate statistics would be sufficient, as in Henson et al., 2019).

The tutorial consists of four main sections: 1) specifying single-subject DCM for MEG and 2) estimating group-level DCMs for MEG with inference based on model comparisons.

4. DCM for M/EEG

DCM for M/EEG consists of various types of analyses that span from biologically plausible neuronal models that effectively describe evoked time courses and spectral activity, to phenomenological models of oscillatory entrainment effects. In this tutorial, we focus on DCM for evoked responses or DCM-ERP, and use the extended Jansen-Rit neuronal model, called the 'ERP' neuronal model. For a comprehensive overview of this model, see David et al. (2006) and Spiegler et al. (2010). We first present specification and estimation of DCM from MEG magnetometers of a single subject, and then extend this to all 16 subjects from the dataset for ‘group’ DCM. Lastly, we show estimation of 2nd level effects using PEB and inference on connectivity based on various approaches to model comparisons.

4.1 Preparation

In order to run this tutorial, there are two preparatory steps involved: setting up the software environment, and organizing data. For preparing the environment, install SPM12 from https://www.fil.ion.ucl.ac.uk/spm/software/download/ somewhere on your local system. The results in this tutorial were obtained with release “r7771”. Once installed, clone or download the git repository at https://github.com/pranaysy/cognestic22_multimodal_dcm/ to a folder. This contains some small edits to spm functions that are detailed in the README. Launch MATLAB and using the file browser panel, navigate to the cloned or downloaded repository folder. We will refer to this folder as ‘base_dir’ in the tutorial as well as scripts.

Once the environment is ready, data can be organized depending on the kind of processing it has been subject to. There are two starting points:

1. You could begin with raw data from OpenNeuro (https://openneuro.org/datasets/ds000117) and process it for fitting DCMs. Note that the full data is about 85GB in size. After downloading the data, process the data as per the demo tutorial demo in Henson et al. (2019) except for the following changes:

   A. During epoching of continuous data, perform baseline correction for minimising activity around stimulus onset. This is necessary because the neuronal models for evoked responses in DCM assume that prior to an ‘input pulse’ to a source, the activity of that source is zero. Typically this input pulse (for cortical ROIs) is modelled as a Gaussian function peaking around 60ms after stimulus onset, to capture the typical transmission time for sensory information from thalamus to cortex. Therefore activity prior to this input pulse should be zero.
   B. In the step for aggregating epochs, use the per-condition ‘Robust averaging’ option. This assigns weights to trials in each condition based on how typical a given sample in that trial is, and uses these weights to attenuate the effect on the average of atypical values.
   C. Lastly, after creating averaged epochs for each subject, calculate two contrasts of conditions: the first being [0 0 1] for ‘Scrambled’, and the second being [1/2 1/2 0] for ‘Faces’ (averaging ‘Famous’ and ‘Unfamiliar’ into one condition).
   D. Optionally, after creating the forward model, you may enforce the estimation of the gain matrix which is stored in a separate file (with the prefix: ‘SPMgainmatrix_*’). This can be done by running the command ‘spm_eeg_lgainmat(D)’. Precomputing the gain matrix will prevent its estimation during DCM inversion, since the same gain matrix, once computed, will be used for all subsequent DCM inversions for that subject’s data. This is not necessary, and if a gain matrix has not been estimated already from the specified forward model, DCM will estimate it and store it in the same file.

   The script ‘spm_master_script_data_preprocessing.m’ in the /code/meg folder includes the above steps. Once you have finished processing the data, you will have a BIDS directory tree with processed derivatives in the folder “ds000117 / derivatives / SPM12”. The data in this folder are ready for fitting DCMs.

2. You could start directly with processed DCM-ready data, which have already undergone the pre-processing outlined above. We provide the estimated gain matrices along with averaged evoked data for each subject on Figshare:

Download the data and extract it inside the 'data' folder in 'base_dir', which is the repository you cloned earlier. The data folder should now have a directory tree that looks like 'data / derivatives / SPM12'.

Initialisation

Open the 'spm_master_dcm_meg_peb_batched_script.m' from the 'code/meg' folder. There is some MATLAB code at the start that is necessary to start SPM & define some key variables. First start SPM12

```
SPM12PATH = <insert path to your local install of SPM12>
addpath(SPM12PATH);
spm eeg
```

Then define some paths where you have downloaded the repository. The 'base_dir' folder is the one containing the folders named 'code', 'data', and 'fits':

```
base_dir = '/user/py01/cognestic22_dcm_meeg/'; % example
cd(base_dir)
```

Next, add the 'code' folder and all subfolders in it to MATLAB's path by running this line:

```
addpath(genpath(fullfile(base_dir, 'code')))
```

This ensures that all the code we provide is available in MATLAB's environment. This is necessary because our scripts for batch processing rely on functions in various files – some of which enable parallel processing, some are 'job files' used by SPM's batching interface while some are modified SPM functions, which have been updated for this tutorial. It is crucial to add this 'code' folder to the MATLAB path after your local installation of SPM has been added to MATLAB's path and launched. This sequence of operations will put our 'code' folder above SPM's on MATLAB's list of paths, which can be viewed by typing `pathtool` on the command line. Since the updated SPM functions we provide share the same filenames with the original SPM functions, adding the two in this exact order guarantees that when a function is to be executed, MATLAB will first look at our 'code' folder, since it is higher up, and then SPM's folders. Therefore, running any code that relies on our modified SPM functions, will always correctly use them and not the default SPM functions. A list of modified SPM functions with a brief overview of changes is provided in the README file on the GitHub repository for this tutorial, linked earlier.

Lastly, if you have access to multiple cores (parallel processing in MATLAB), you can run below (if not, set "numworkers" to 0):

```
numworkers = 16; % Number of workers for distributed computing (depends on
system, here 16 implies one worker for each subject)
if numworkers > 0
    delete(gcp('nocreate')) % Shut down any existing pool
    parpool(numworkers);
end
```

4.2 DCM definition: for single subject

Specification of DCMs is done through the SPM GUI's "DCM" button after launching SPM's M/EEG interface. In MATLAB, use the file selector on the left-hand side to change the working directory to sub-15.

4.2.1 Specifying a full model

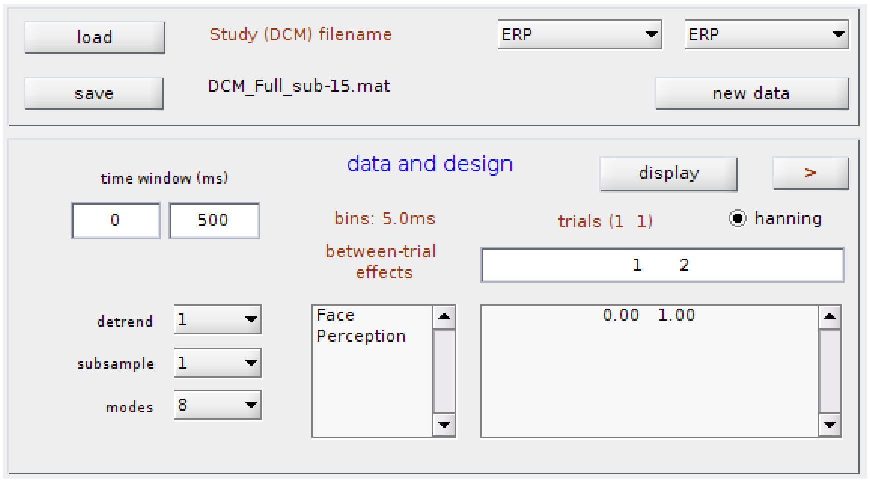
DCM specification for M/EEG involves four steps: choosing modelling approach, specifying data and design matrix, specifying the electromagnetic observer model and lastly, defining the neuronal connectivity model. More practical help on DCM for fMRI can be found in Chapter 44 of the SPM12 manual, https://www.fil.ion.ucl.ac.uk/spm/doc/spm12_manual.pdf.

4.2.1.1 Modelling approach

- In the top-left corner of the DCM graphical interface (Figure 2), the 'load' button allows selection and loading of an existing DCM. This is useful if you want to make copies of a specified model with different parameters.
- To the right of the 'load' button, there are two drop-down lists, the left of which allows you to select the type of DCM analysis. Select the option 'ERP' from this list. This corresponds to DCM for evoked responses. Other options are explained in Litvak et al. (2011).
- The second drop-down list in the top-right corner of the DCM GUI allows you to select an underlying generative neuronal model for modelling each source's dynamics and connectivity. The various models differ in their internal architecture and parametrisations. Detailed descriptions of these models can be found in Moran et al. (2013) and Pereira et al. (2021). We focus on the convolutional, 3-population, extended Jansen-Rit model, referred to as the 'ERP' neuronal model as described in David et al. (2006). Select this option from the list.
- Lastly, click on the 'new data' button and select the file 'wmaMceffdspmeeg_sub-15_ses-meg_task-facerecognition_run-01_proc-sss_meg.mat' from the folder containing processed data for sub-15. Since the data consists of multiple sensor modalities, and DCM presently can only model one modality at a time, the interface will prompt you to select one of the three modalities. Select 'MEGPLANAR' for proceeding with MEG Planar Gradiometers.

4.2.1.2 Data and design

- Select data for a 500ms window after stimulus onset by setting the values 0 and 500, respectively, in the boxes under 'time window' (see Figure 2).
- By default, only the first trial is selected in 'between-trial effects'. In the input box, change '1' to '1 2' to select two trials, corresponding to scrambled faces and famous faces respectively. Click the 'display' button above to verify time-courses of the two observed trials across all sensors.
- The large white box, located right under the box where the two trials were selected, allows you to specify the design matrix for modelling effects as contrasts of trials. Enter the contrast vector '0 1' in this box corresponding to the effect of face perception i.e. scrambled versus faces (famous + unfamiliar). In the white box to the left of the design matrix, names to each contrast vector can be assigned. Change 'effect 1' to 'Face Perception'.
- Select the radio button for 'hanning' to apply a Hann taper to the time courses. This will improve the signal-to-noise ratio by suppressing activity towards the end of the 0-500ms time window of interest, i.e., focus on effects occurring in the middle of the window. Confirm this by clicking the 'display' button.
- Leave the remaining options to their default values and press the button with the right-pointing red arrowhead to continue to the next section.



Study (DCM) filename: DCM_Full_sub-15.mat

ERP | ERP

time window (ms): 0 500

bins: 5.0ms

trials (1 1) ● hanning

between-trial effects: 1 2

detrend: 1

subsample: 1

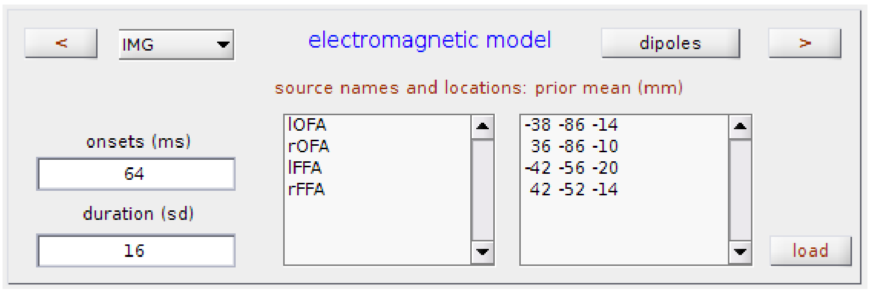
modes: 8

Face Perception: 0.00 1.00

Figure 2: Data and Design

4.2.1.3 Electromagnetic model

- In this set of options, we specify the spatial constraints for parametrising the leadfield used for projecting source activity to the sensors (Figure 3). There are two ways of doing this, the first approach models the leadfield of each source as an equivalent current dipole (ECD), while the second one, called the 'Imaging' solution, models each source as a 'patch' of dipoles on the cortical mantle. Alternatively, the spatial model may not be required at all if the data come from intra-cortical electrodes (like local field potentials), or if the data are already in source space after some bespoke source reconstruction. In such a scenario, the 'LFP' electromagnetic model should be used. Here, we use the IMG approach, so select this option from the drop-down list in the top-left corner of this section.
- Next, enter names and locations of sources in the two white boxes in this section. In the left box, enter the following names, one on each line: lOFA, rOFA, lFFA, rFFA. In the right box, enter their respective locations (in mm, in MNI coordinates), one on each line again, as: -38 -86 -14, +36 -86 -10, -42 -56 -20, +42 -52 -14. These coordinates are the same as the ones specified during VOI extraction for fMRI, in the DCM for fMRI tutorial.
- The specified dipoles can be visualized by pressing the button 'dipoles'. This will update the 'Graphics' window and either one or all the specified dipoles can be visualized in the brain.
- Leave the remaining options to their default values, press the button with the right-pointing red arrowhead to continue to the next section.



IMG

source names and locations: prior mean (mm)

onsets (ms): 64

duration (sd): 16

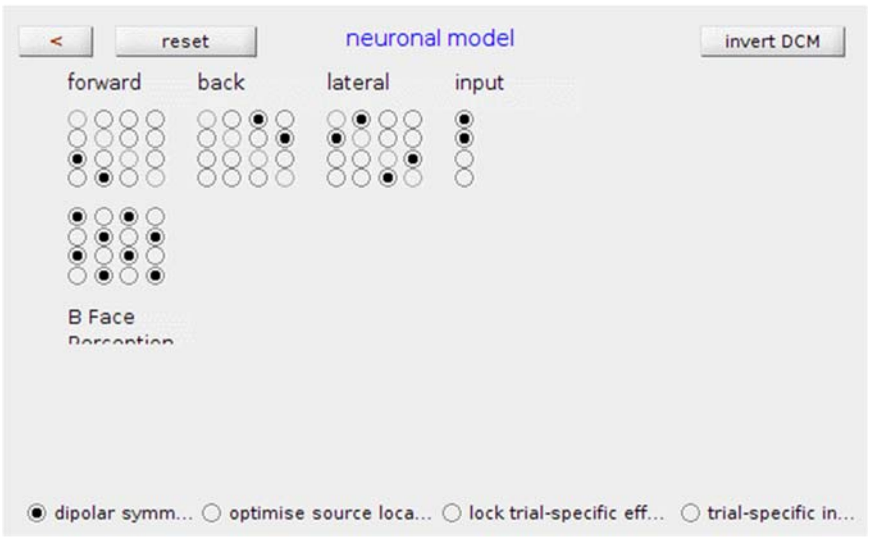
lOFA	-38 -86 -14
rOFA	36 -86 -10
lFFA	-42 -56 -20
rFFA	42 -52 -14

Figure 3: Electromagnetic observation model

4.2.1.4 Neuronal model

- In this set of options, we define how the sources defined in the previous step are connected to each other and how these connections are modulated by experimental effects (Figure 4). The connectivity architecture also depends on the underlying generative neuronal model chosen in Section 4.2.1.1. Since we chose the 3-population 'ERP' model, this section will consist of three A-matrices, one B-matrix and one C-matrix.

- The A-matrices correspond to three sets of connections that originate from the layer V pyramidal cell population in each modelled source: 'forward' ones carry bottom-up information and terminate in the layer IV spiny stellate cell population of another source; 'backward' ones carry top-down information and terminate in layer II/III inhibitory interneuronal as well as layer V pyramidal cell populations; and 'lateral' ones terminate in all three layers – layer II/III inhibitory interneuronal, layer IV spiny stellate cell and layer V pyramidal cell populations.
- The B-matrices encode gain modulations of the connections defined by the A-matrices, with the modulations capturing the difference between evoked responses across conditions. Like in the accompanying fMRI tutorial, DCM will explain the response to 'scrambled faces' by using the A-matrix only, while the response to 'famous faces' and 'unfamiliar faces' is modelled by modulating the connections in the A-matrix by their corresponding weights in the B-matrix. Further, there are as many B-matrices as there are effects (rows) specified in the design matrix in Section 4.2.1.2. Since we have only one effect of 'Face Perception' in our design matrix, we have only one B-matrix.
- Lastly, the C-matrix allows us to specify which ROIs receive inputs, modelled as stimulus impulses (Gaussian functions with a mean of around 64ms and standard deviation of 16ms, to resemble thalamic input to cortex).
- The rows and columns of each of the A and B matrices (only rows for C-matrix) correspond to the sources in the same order as specified in the electromagnetic model. The column index corresponds to the source "from" which a connection originates, while the row index corresponds "to" the source at which a connection terminates. A connection from a source to another is switched 'on' in a matrix by pressing the radio button located at that index.
- Based on these rules, set up the A matrices such that OFA projects forward connections to FFA and FFA projects backward connections to OFA bilaterally. Further, both OFA and FFA are connected across hemispheres by bidirectional lateral connections. In the C matrix, specify bilateral OFAs as the only two regions receiving direct inputs.
- Set up a B matrix by switching on all the between-region connections as specified in the 'forward' and 'backward' A matrices as well as self-connections that correspond to the diagonal of the B matrix. This corresponds to the 'Full' model containing all relevant sources, connections and modulations, except for inter-hemispheric lateral connections.
- Lastly, enable the 'dipolar symmetry' radio button in the bottom-left corner of this section. This imposes symmetry constraints on the orientations of dipoles modelled at homologous sources, based on the fact that the cerebral hemispheres are organised symmetrically.



● dipolar symm... ○ optimise source loca... ○ lock trial-specific eff... ○ trial-specific in...

Figure 4: Connectivity of neuronal model and modulation by Face Perception

Save this 'full' model as 'DCM_Full' in the 'DCMs' subdirectory located in the 'templates' folder under 'fits/batch_gui/meg' by pressing the 'save' button on the top-left corner of the DCM interface. This model is now ready for inversion.

4.2.2 Estimating the full model for single subject

Press the 'invert DCM' button located in the top-right corner of the 'neuronal model' section to invert this model. The estimation process will proceed until a reasonable fit is obtained, up to a fixed maximum number of iterations of the underlying expectation-maximization algorithm (which is 64 by default, but we have increased this to 512 steps to ensure that all models may converge). Once complete in 53 steps for this subject, the 'invert DCM' button will change to 'Estimated', and produce a figure like that in Figure 5.

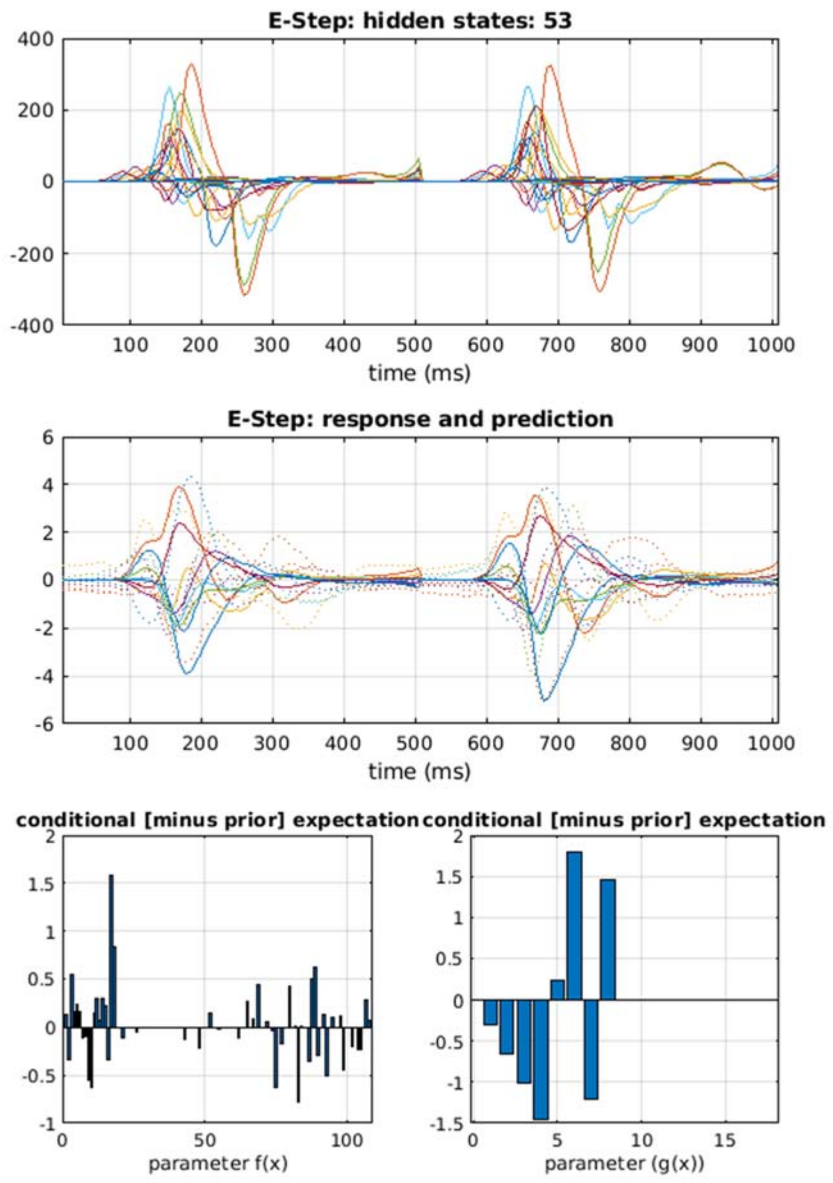


Figure 5: Completed estimation of specified DCM

The inverted DCM can be inspected using the drop-down list at the bottom-left of the DCM interface. Estimates of various parameters as well as fits to observed data can be evaluated, such as modelled activity at sources (select 'ERPs (sources)'; see Figure 6) and scalp (select 'Response'; see Figure 7) for each condition. Modulation of coupling, i.e. estimates of B-matrix parameters can be inspected in two ways, firstly, by selecting 'coupling (B)' – this shows estimates as well as posterior probabilities in a matrix format (Figure 8), and secondly, by selecting 'trial-specific effects' – which shows bar graphs of the change in strength (or modulation) expressed as a percentage change for Faces relative to Scrambled, which is held at 100% (Figure 9). The layout of this latter option mimics that of the B-matrix.

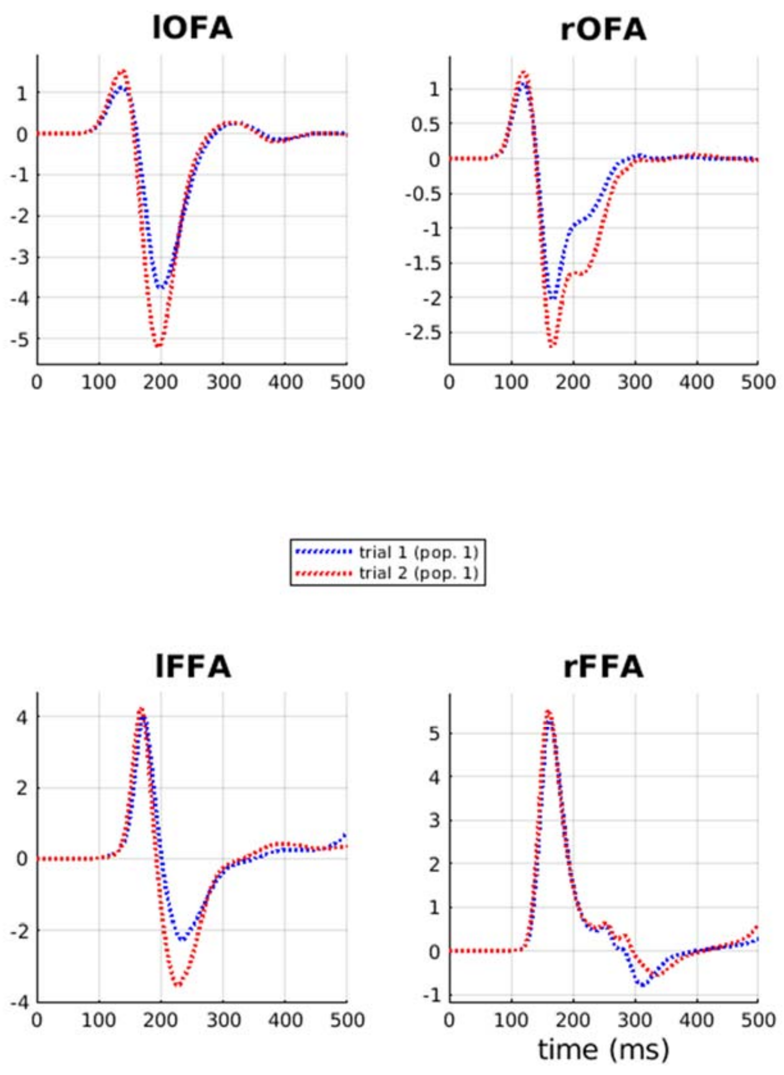


Figure 6: Estimated activity for each trial condition at each source. Trial 1 (blue) is scrambled faces; Trial 2 (red) is faces; populations 1 (pop. 1) corresponds to the 'output' population i.e. pyramidal cells in each source.

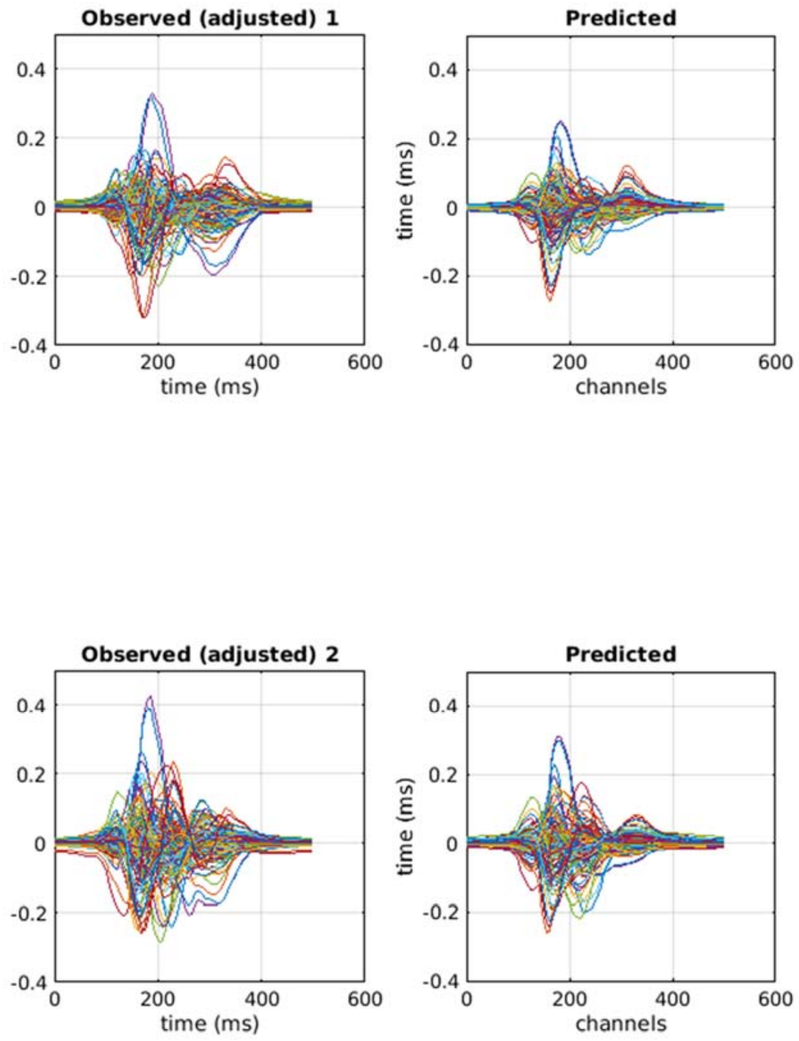
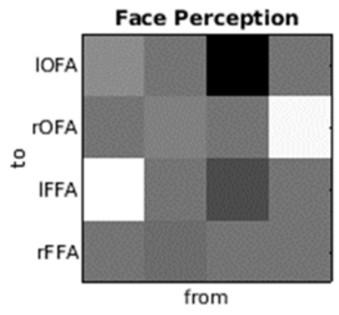
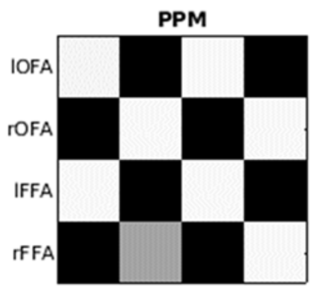


Figure 7: Fitted time courses at sensors



```
1.09 1.00 0.54 1.00
1.00 1.05 1.00 1.52
1.55 1.00 0.85 1.00
1.00 0.97 1.00 1.01
```



```
0.98  NaN 0.99  NaN
 NaN 1.00  NaN 1.00
1.00  NaN 1.00  NaN
 NaN 0.67  NaN 1.00
```

Figure 8: Estimated B-matrix parameters (top) & their respective posterior probabilities (bottom)

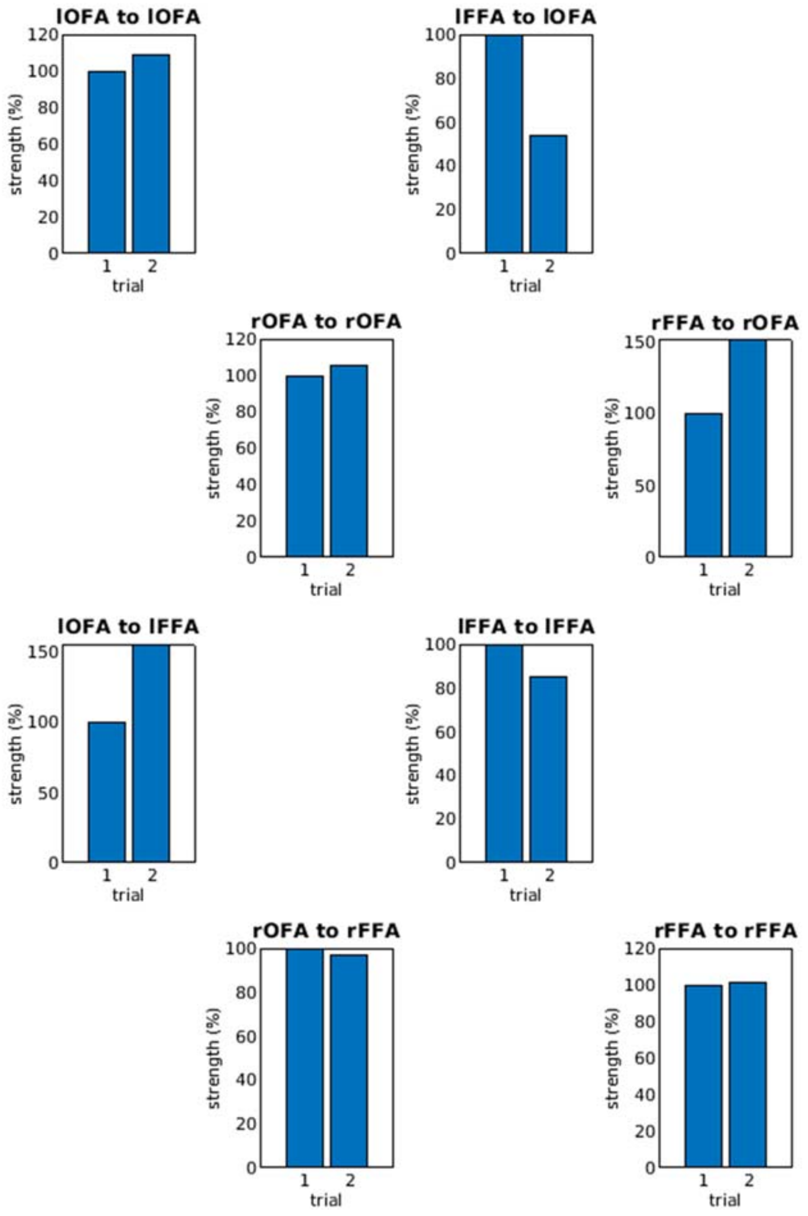
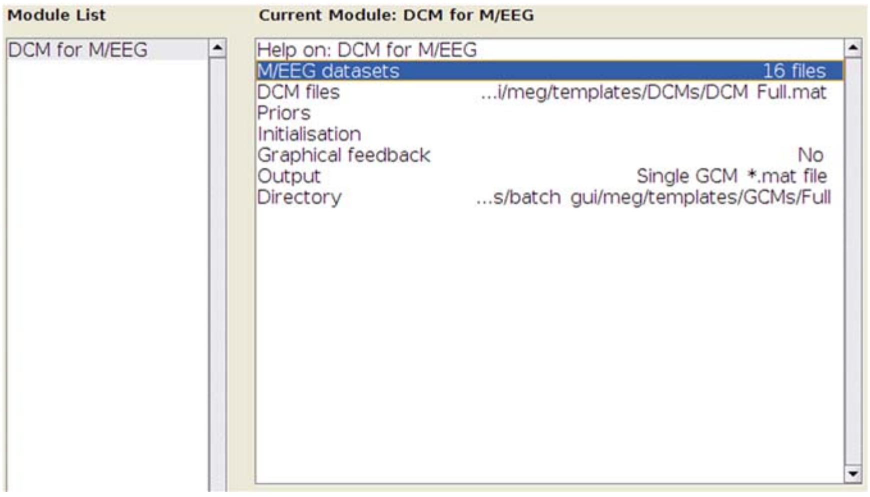


Figure 9: Trial-specific modulation of connections. This figure shows change in connection strengths for Trial 2 (Faces) relative to Trial 1 (Scrambled), where the latter is 100% (since only Faces are modulated in this DCM).

4.3 Group DCM (or GCM)

We have specified a DCM for a single subject, which we can use as a template model for every subject. This way, all subjects' models will be identical, except for their M/EEG time series and forward models, which will be customized for each subject. While there are several ways of doing this, we demonstrate an approach using SPM's batch GUI interface, which replicates the 'Full' template model over all subjects, like we did in accompanying fMRI tutorial. However, rather than running this step and further steps one-by-one, we illustrate construction of a pipeline, using "dependencies" in the batch interface, to chain together multiple steps for 1) replication of template models over subjects, 2) estimating these models as well as 3) fitting a PEB model for estimating group level effects.



```
Module List        Current Module: DCM for M/EEG
DCM for M/EEG      Help on: DCM for M/EEG
                   M/EEG datasets                                      16 files
                   DCM files                ...i/meg/templates/DCMs/DCM Full.mat
                   Priors
                   Initialisation
                   Graphical feedback                                        No
                   Output                                Single GCM *.mat file
                   Directory           ...s/batch gui/meg/templates/GCMs/Full
```

Figure 10: Specification step: DCM for M/EEG module in the batch interface

4.3.1 Replicating model specification across multiple subjects

The first step involves preparing and setting up the batch module for specifying DCMs at a group level. Start the 'Batch Editor' by pressing the 'Batch' button located in the bottom row of the SPM EEG window. This will launch the batch interface with no modules loaded.

- Add the first module by opening the 'SPM' menu from the top row, and selecting 'DCM' → 'DCM specification' → 'DCM for M/EEG' (Figure 10).
- In module options, select 'M/EEG datasets' and press the 'Specify…' button. In the file selector interface that opens up, navigate to the 'data' folder containing the processed files in the 'derivatives' sub-folder. Although we could select each subject's processed data files one at a time, by navigating to their respective sub-directories and manually clicking on each file, we make use of the recursive regular-expression-based filename matching filter in this interface. This is especially useful since all subjects' data underwent the same pre-processing steps, therefore each file has the same string of letters prepended to its name by SPM, where each letter corresponds to one step of processing. In the current dataset, this prepended string is 'wmaMceffdspmeg'. In the text box next to the 'Filter' option, enter the regular expression: '^wmaMc.*'. This expression will match all files whose names begin with 'wmaMc' (indicated by the '^') and can have any set of letters after this string. Now press the 'Rec' button towards the left in the same row as the text box. This will recursively search all sub-directories for files that match the pattern specified. Once the search is complete, the box at the bottom will be populated with 16 files corresponding to the processed SPM header files (*.mat) for all subjects. You can view this list and edit if needed by pressing the 'Ed' button, which launches a text editor with a populated list of all files. Press 'Done' to specify these 16 files for the option 'M/EEG datasets'.
- Select the option 'DCM files' and press the 'Specify…' button for indicating the full template model. In the file selector interface, navigate to the 'DCMs' subdirectory in the 'templates' folder and select the file 'DCM_Full.mat' from the panel on the right. Press 'Done' to confirm and return to the batch editor.
- Since we are using the default priors for our DCM parameters, we will leave the options 'Priors' and 'Initialisation' empty as is. Select 'Graphical feedback' and change it to 'No', as this will slow-down execution and more importantly would not produce anything if you run on multiple cores using MATLAB's parallel computing, as here.
- Leave the 'Output' as a 'Single GCM *.mat file'. This is not only a compact, efficient way of representing group DCMs across subjects (as rows) and models (as columns), but subsequent

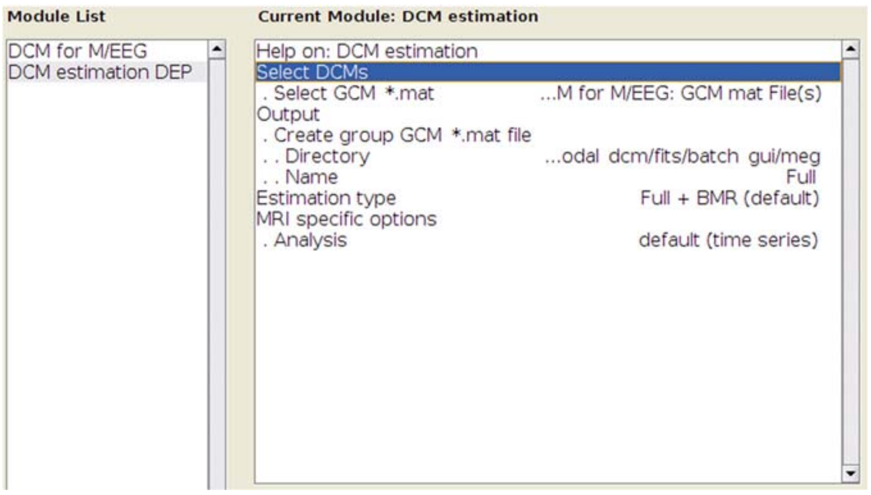
functions for fitting PEBs and model comparison also rely on this 'GCM' and the way it is structured.

- Point the output 'Directory' to the 'fits/batch_gui/meg/templates/GCMs/Full' folder. Note that the GCM specified this way will be implicitly named after the first file specified in 'DCM files' above, with prefix 'GCM_'. In our case, it will be named 'GCM_DCM_Full.mat'.

With this configuration, the 'DCM for M/EEG' module for specifying DCMs for all subjects is ready and can be run by either pressing the green play button or by selecting 'File' → 'Run Batch' in the batch interface. Doing so will generate a 'GCM' file with the '.mat' extension in the 'fits' directory. This can be inspected by loading it in MATLAB, or can be used for estimation of the specified models. However, here we will wait to specify some subsequent steps before running.

4.3.2 Estimating GCM

We will now add the estimation step and chain it with the specification step we just completed to build our pipeline (Figure 11).



```
Select DCMs
 . Select GCM *.mat               ...M for M/EEG: GCM mat File(s)
Output
 . Create group GCM *.mat file
 . . Directory                    ...odal dcm/fits/batch gui/meg
 . . Name                                                   Full
Estimation type                               Full + BMR (default)
MRI specific options
 . Analysis                                default (time series)
```

Figure 11: Estimation step: DCM estimation module in batch interface

- Add the estimation module by opening the 'SPM' menu from the top row, and selecting 'DCM' → 'DCM estimation'
- In the option 'Select DCMs', you could specify DCMs per subject (= columns of GCM), or per model (= rows of GCM), or you could specify a GCM directly. This is the default option. If you have an existing GCM specified manually, you could specify a path to the file containing this GCM. Since we will use the GCM produced by the preceding step, we can specify the output of the previous module as a dependency. With the 'Select GCM_*.mat' option selected, press the 'Dependency' button. This will open a pop-up dialog box with only one option 'DCM for M/EEG: GCM mat File(s)' selected by default. Press 'OK' to continue with this selection.
- In the 'Output' option, select 'Create group GCM_*.mat file'. This will create an array of DCMs with subjects as rows of the array and columns corresponding to models (though only one in our case, at the moment). Selecting this option will also prevent overwriting the GCM 'specification' created in the previous step. This may be useful for testing different ways of estimation (see 'Estimation type' below). In the selected branch 'Create group GCM_*.mat file' of 'Output' option, specify 'Directory' as the 'meg' subdirectory in the 'fits/batch_gui' folder and 'Name' as 'Full'. The prefix 'GCM_' will be appended to the name, making 'GCM_Full.mat' the final filename of the estimated GCM.

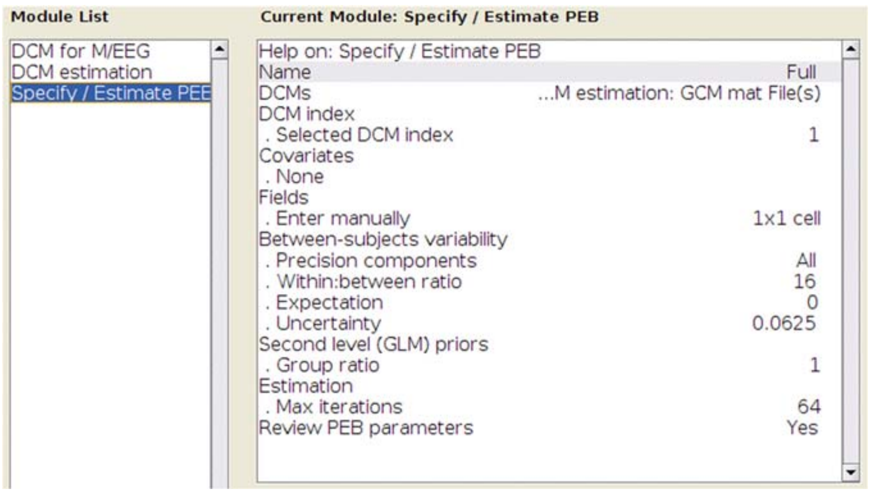
- In the option ‘Estimation type’, select ‘Full + BMR’. This option will estimate the ‘full’ model for each subject (first column of GCM) and then uses Bayesian Model Reduction or BMR to infer the parameters and model evidence for any subsequent nested models. If we had multiple models in our GCM (i.e. more than one column), then only the first column of GCM corresponding to the full model for each subject will be estimated, while the remaining columns of GCM corresponding to the nested reduced model will be subjected to BMR for inferring parameters from the full model (Zeidman et al 2019).

  Since we only have a single model per subject in our GCM, BMR will not be performed. Note that the alternative of ‘Full + BMR PEB’ can be used for iterative estimation of the full model for each subject. This sets the priors on each parameter to the group mean from a PEB model and then re-estimates the full model for each subject. Doing so improves estimation by rescuing parameters stuck in local optima, but due to the iterative nature of fitting, can take much longer to estimate, and we do not use here.

The estimation module is now ready and chained to the previous specification step as a dependency. If you ran this batch pipeline by pressing the green ‘play’ button, SPM will sequentially run both modules to first create a GCM specification (‘fits/batch_gui/meg/templates/GCMs/Full /GCM_DCM_Full.mat’), and then estimate this specified GCM (‘fits/batch_gui/meg/GCM_Full.mat’). The estimated GCM could then be investigated by loading it in MATLAB and calling various SPM functions to review DCMs in the GCM array, or can be used for estimating a group-level PEB model. However, here we will continue with next step of PEB to estimate the 2nd level group-level model.

4.5.3 Second level analysis: PEB

We continue and extend our pipeline to lastly estimate a group-level PEB model from the estimated GCM by chaining another module (Figure 12).



Module List

DCM for M/EEG
DCM estimation
Specify / Estimate PEB

Current Module: Specify / Estimate PEB

```
Help on: Specify / Estimate PEB
Name                                          Full
DCMs                ...M estimation: GCM mat File(s)
DCM index
 . Selected DCM index                            1
Covariates
 . None
Fields
 . Enter manually                         1x1 cell
Between-subjects variability
 . Precision components                        All
 . Within:between ratio                         16
 . Expectation                                   0
 . Uncertainty                              0.0625
Second level (GLM) priors
 . Group ratio                                   1
Estimation
 . Max iterations                               64
Review PEB parameters                          Yes
```

Figure 12: PEB step: Specify / Estimate PEB module in batch interface

- Add the PEB estimation module by opening the ‘SPM’ menu from the top row, and selecting ‘DCM’ → ‘Second level’ → ‘Specify / Estimate PEB’
- Specify ‘Name’ as ‘Full’. The prefix ‘PEB_’ will be appended to this name and the estimated PEB model will have the name ‘PEB_Full.mat’, written in the same directory as the estimated GCM i.e. the ‘batch_gui/meg’ subdirectory in the ‘fits’ folder.
- Leave the ‘DCM index’ option to ‘Selected DCM index’ with the value of 1 (corresponding to the first column of GCM with the full model for each subject).

- Leave the ‘Covariates’ option set to ‘None’ as we illustrate this functionality later in 4.5.5.
- Set the option ‘Fields’ to ‘Enter manually’ and then select this option and specify “{‘B’}”, including the curly braces. This option specifies which parameters of our model should be treated as random effects across subjects (other parameters are treated as fixed effects). Since our models differ in connections being modulated by the effect of face perception, we specify B-matrix parameters as the ones we want to model at the group level, accounting for between-subject variability over these parameters.
- Leave the options ‘Between-subjects variability’, ‘Second level (GLM) priors’ and ‘Estimation’ to their default values. Set ‘Review PEB parameters’ to ‘Yes’ as we would like to examine the group-level estimates at the end of the pipeline.

With this module, our pipeline is now ready to be run. Press the green play button, or select ‘File’ → ‘Run Batch’ to run this batch pipeline. This may take a while to run, and on completion, the following outputs will be produced:

1. In ‘fits/batch_gui/meg/templates/GCMs/Full’: one GCM specification file (‘GCM_DCM_Full.mat’)
2. In ‘fits/batch_gui/meg/templates/GCMs/Full’: one full DCM fitted per subject for a total of 16 files.
3. In ‘fits/batch_gui’: one estimated GCM (‘GCM_Full.mat’) & PEB (‘PEB_Full.mat’) file each.

Once the pipeline has finished, you will be presented with a window for reviewing the group PEB structure (as in Figure 13). This interface can be used for examining estimated group-level parameters (by selecting ‘Second-level effect – Commonalities’ in the main drop down menu) as well as quality of fits (for example, selecting ‘Diagnostics’ will show correlations between parameters). The group-level parameter estimates can be thresholded based on their posterior probabilities for evaluating which modelled parameters were consistent across subjects.

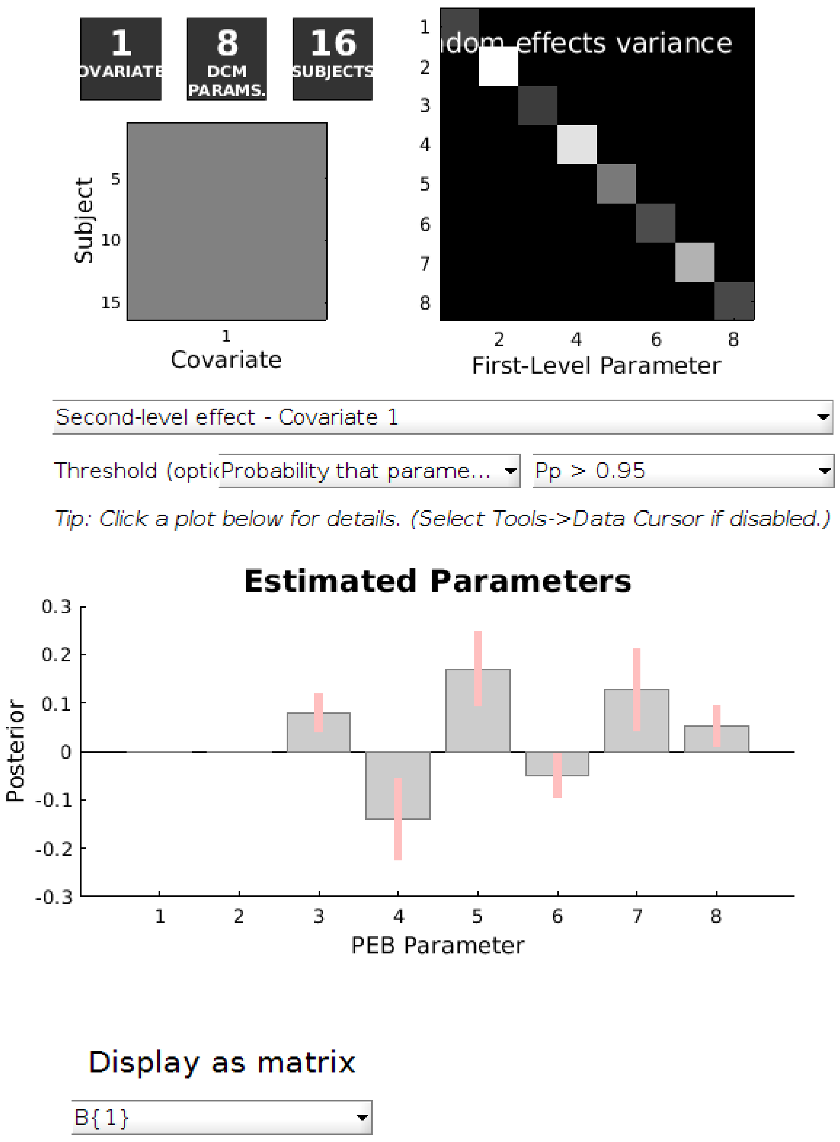


Figure 13: Review of estimated PEB parameters

In the batch editor, select 'File' → 'Save Batch and Script' and save the pipeline as 'batch_specify_estimate_peb' in the folder 'code/meg/saved_from_batch_interface'. We will later modify this batch file for use through scripts. Create a new batch by selecting 'File' → 'New Batch' before proceeding further.

4.4 Model Comparison

To make use of the PEB model, we need to perform a model comparison. The simplest form of model comparison to run is an automatic search, which will prune parameters from the PEB model that do not contribute to the model evidence. The software will specify and compare hundreds of candidate reduced PEB models, in which different combinations of parameters have been switched off. This search can be performed quickly owing to a method called Bayesian Model Reduction (BMR), in which the parameters and model evidence for any nested model can be estimated from the full model fit by simple equations, without needing to re-fit each nested model to the data. Moreover, we can also average the parameters (connection strengths) across the whole model space, weighted by the model evidence for each model; a process called Bayesian Model Averaging (BMA).

4.4.1 Automatic Search

- In a new empty batch, add the module for automatic nested model search by opening the ‘SPM’ menu from the top row, and selecting ‘DCM’ → ‘Second level’ → ‘Search nested PEB models’.
- In the module options, for ‘Select PEB file’, specify the ‘PEB_Full.mat’ file in the ‘fits/batch_gui/meg’ directory, and similarly, for ‘DCMs’, specify the ‘GCM_Full.mat’ file located in the same folder.
- Set the ‘Null prior variance’ to 0 instead of the default 0.0625. This will switch ‘off’ a parameter completely instead of setting a low value, and tests the null hypothesis that parameters are present vs absent. Lastly, select ‘Yes’ for ‘Review PEB parameters’.

Press the green play button to execute this module and run an automatic search for nested PEB models. This should be quick, and once completed, you will be presented with three windows – one titled ‘BMR – all’ showing the models in the BMR model space that contribute to model evidence, another titled ‘BMC’ showing estimates from the averaged model from BMA, and the third one titled ‘PEB – Review Parameters’ showing parameters and quality of fits from the BMA (as in Figures 14-17 below).

This module will also automatically save the BMA as ‘BMA_search_PEB_Full.mat’ in the ‘fits’ folder. Save the batch file and script as ‘batch_bma_search’ in the folder ‘code/saved_from_batch_interface’.

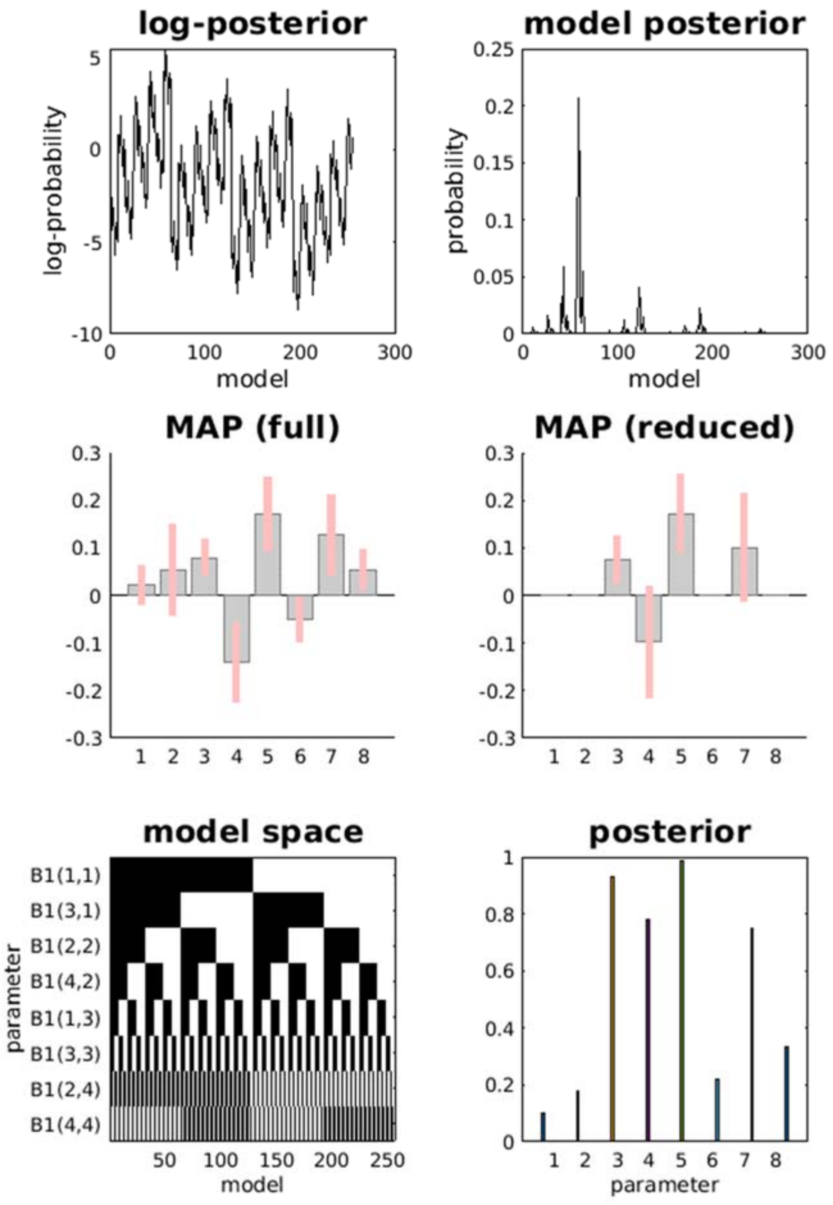


Figure 14: BMR window after searching nested models

The figure titled 'BMR –all' (Figure 14) shows the 256 reduced candidate models that were compared during the automatic search (the full number of models searched is 2^8 since 8 'B' matrix parameters were estimated during the PEB stage). This model space is shown in the bottom row, left plot, and for each model, the parameters that were 'on' are shown in white, while the ones that were 'off' are in black For instance, the first 128 models have the parameter B1(1,1), corresponding to modulation of the self-connection for left OFA, switched off while the remaining 128 models have it switched on.

The log of model evidence is shown in the left plot of top row, while the right plot shows these values converted to posterior probabilities. Note that no single model is 'best', although a few models appear more likely. The second row shows parameters of the PEB model before search (left) and those after search (right). Four parameters have been pruned away: 1, 2, 6 and 8 which have near-zero estimates after search. The right plot in the bottom row shows the posterior probability for each PEB parameter obtained by comparing evidence for all reduced models which had that corresponding parameter switched on, versus all reduced models which had that same parameter switched off.

When comparing a large set of models, it is unusual to find a single winning model. Instead of comparing individual models, it is more informative to consider the weighted average of parameters of models, called the BMA. The window title 'BMC' (Figure 15) shows this average, with plots organised into three rows. The top row shows parameters from the estimated PEB, while the middle row shows parameters from the BMA, with their respective posterior probabilities in the bottom row. The bottom row shows posterior probabilities for each parameter as described above. After pruning parameters, other parameters have become less strong, eg credible interval for parameter 5 now overlaps 0. The bottom row shows the posterior probability for each parameter (as described above), for instance parameter 5 – B{1}(1, 3), corresponding to modulation of the backward connection from left FFA to left OFA has a probability close to 1, suggesting that it is needed.

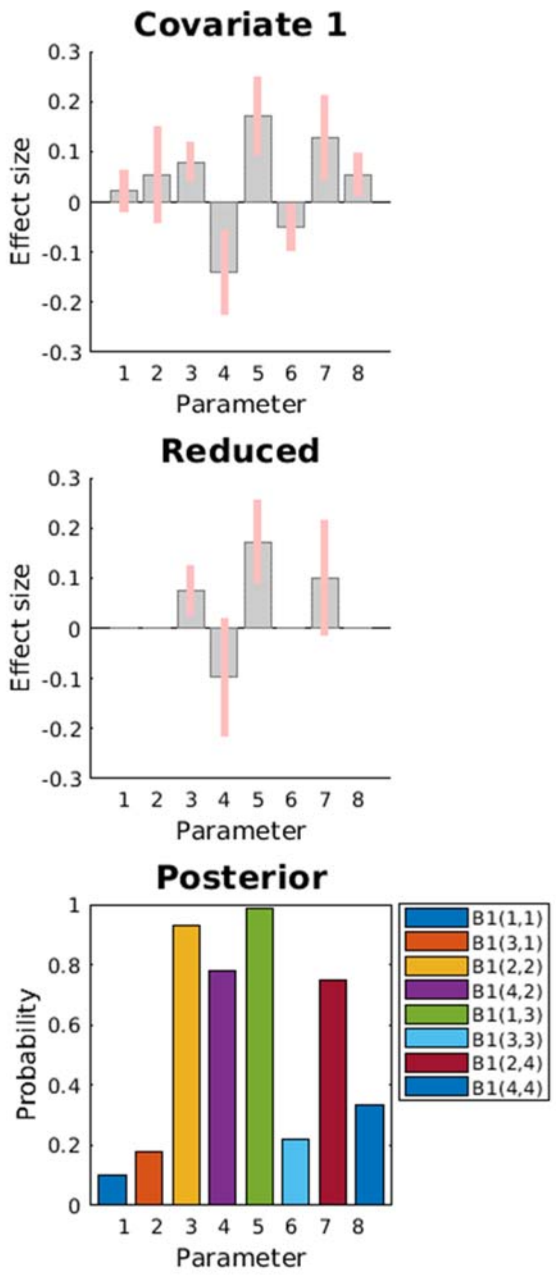
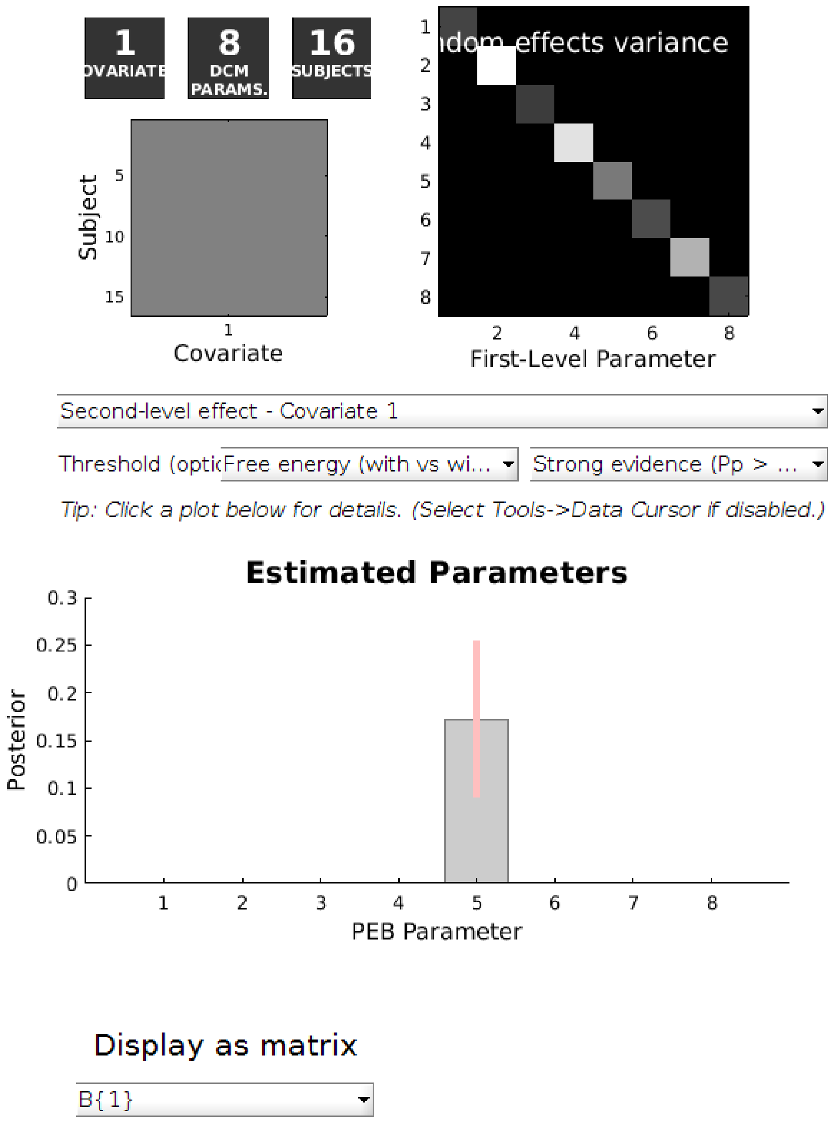


Figure 15: BMC window showing parameter estimates in reduced BMA

The final window, titled ‘PEB – Review Parameters’ (Figure 16), allows an interactive way to explore the results of this analysis. Using the drop-down menu to view ‘Second-level effect - Commonalities’ and thresholding parameters based on model evidence shows that only one parameter had strong evidence (posterior probability > 0.95) in the BMA. Clicking on the bar plots shows this to be the modulatory backward connection from lFFA to lOFA (parameter 5). Alternatively, these can be viewed in a matrix representation by selecting ‘B{1}’ from the drop-down list under ‘Display as matrix’ (Figure 17).



Second-level effect - Covariate 1

Threshold (opti Free energy (with vs wi... Strong evidence (Pp > ...

Tip: Click a plot below for details. (Select Tools->Data Cursor if disabled.)

B{1}

Figure 16: Review of parameter estimates in the Bayesian Model Average

One can see that the backward connection from lFFA to lOFA is positively modulated, implying that an increased backflow of information from left FFA to left OFA accounts for the difference in processing ‘Faces’ over ‘Scrambled’ images.

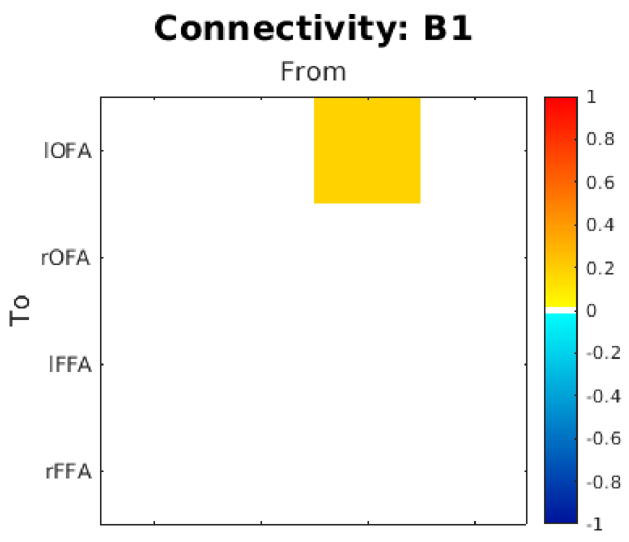


Figure 17: Connectivity matrix showing significant parameters of the B-matrix after thresholding based on model evidence (F > 0.95)

4.4.2 Specific Models

The automatic search explores all possible reduced models from the full model, and this may not be sufficient for testing specific hypotheses about parameters of interest. While the BMA consists of connections that are needed, more general questions like ‘Are between-region connections modulated by faces?’ are not easily addressable. There are two ways to test such hypotheses, firstly, by directly comparing pairs of models – for instance, one with between-region parameters switched on and another with between-region parameters switched off; and secondly, by comparing ‘families’ of connections – for instance, one family with all possible nested models with at least one between-region connection and another family without any between-region connections.

We first perform a simpler comparison of two models, and illustrate how we can use this approach for sequential tests of hypothesis on groups of connections.

4.4.2.1 Specify a reduced (“self-only”) model

We wish to compare the 'full’ model specified above to a nested ‘reduced’ model that only has self-connections but no between-region connections modulated by faces. We will then compare the full model against this reduced model to infer whether modulations of between-region connections are needed to fit the data ‘best’ (in terms of model evidence).

To create this reduced model, load the existing ‘Full’ DCM by launching the DCM GUI interface, pressing the ‘load’ button and selecting the file ‘DCM_Full.mat’ you specified earlier. Edit this DCM by switching off all radio buttons in the B-matrix which are not on the diagonal. Leave all the other options as before, and save this model separately with the name ‘DCM_Self’ in the ‘DCMs’ subdirectory located in the ‘fits/batch_gui/meg/templates’ folder.

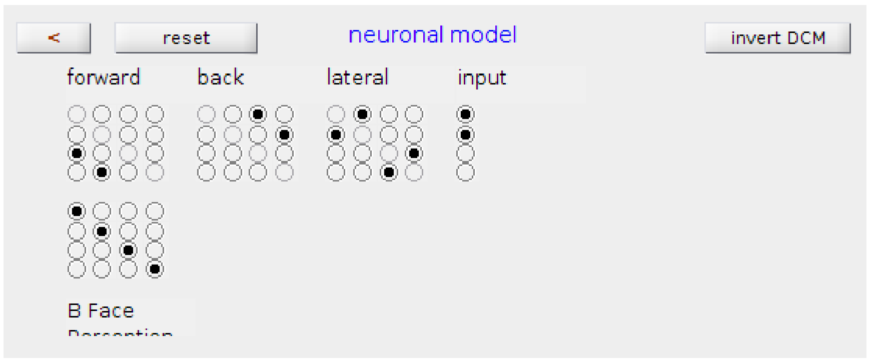


Figure 18: Self-only model: connectivity of neuronal model

4.4.2.2 Construct model space

We will now construct a model space by specifying a GCM in which the first column for each subject corresponds to the full model with both between and within region connections, and the second column for each subject corresponds to the reduced self-only model. In order to do this, start a new batch and add the module for specifying a group DCM by repeating the steps described in 4.3.1 with two changes.

Firstly, instead of specifying a single DCM in the option 'DCM files', specify two DCM template models: 'DCM_Full.mat' and 'DCM_Self.mat' from the 'fits/batch_gui/meg/templates/DCMs' folder. The order of specification of these DCM template models is used for ordering the columns of the GCM structure. It is therefore recommended to the keep the full model as the first template model in this option.

Secondly, point the output 'Directory' to the 'fits/batch_gui/meg/templates/GCMs/Full_vs_Self' folder. Keeping the model-space GCMs in a separate folder this way prevents overwriting previous GCMs. This happens because the specification module implicitly names the GCM after the first DCM template, which is always the full model called 'DCM_Full.mat' in our case. As a result, all GCMs specified using this module share the same name 'GCM_DCM_Full.mat'.

4.4.2.3 Setup BMC under PEB

Next, to compare the models, add the corresponding module by selecting 'SPM' → 'DCM' → 'Second level' → 'Compare / Average PEB models'. In the module settings, specify the 'PEB_Full.mat' PEB file and specify 'DCMs' as a dependency on the GCM produced from the previous step. Leave the 'Review PEB parameters' option to 'Yes' and run the module. This will now create a GCM and perform model comparison, producing two windows. Save this batch file and script as 'batch_specify_bmc' in the folder 'code/saved_from_batch_interface'.

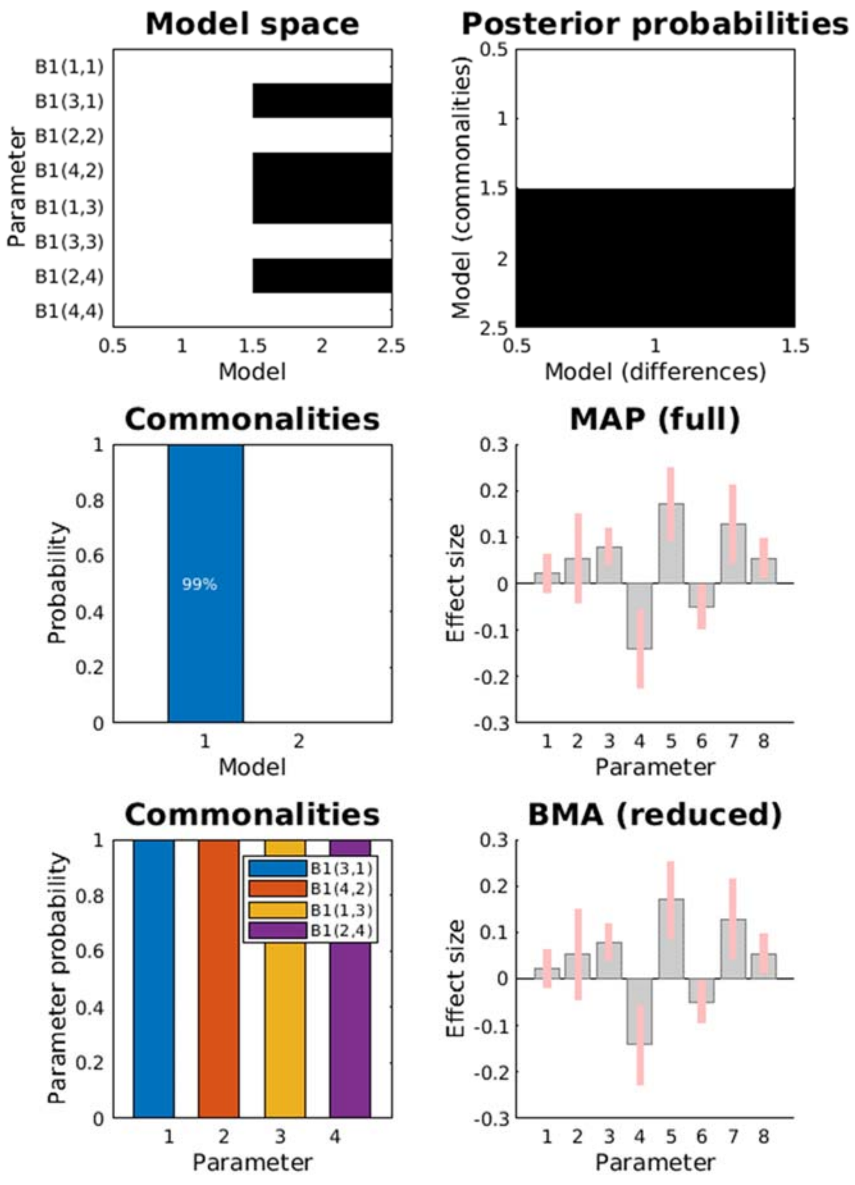
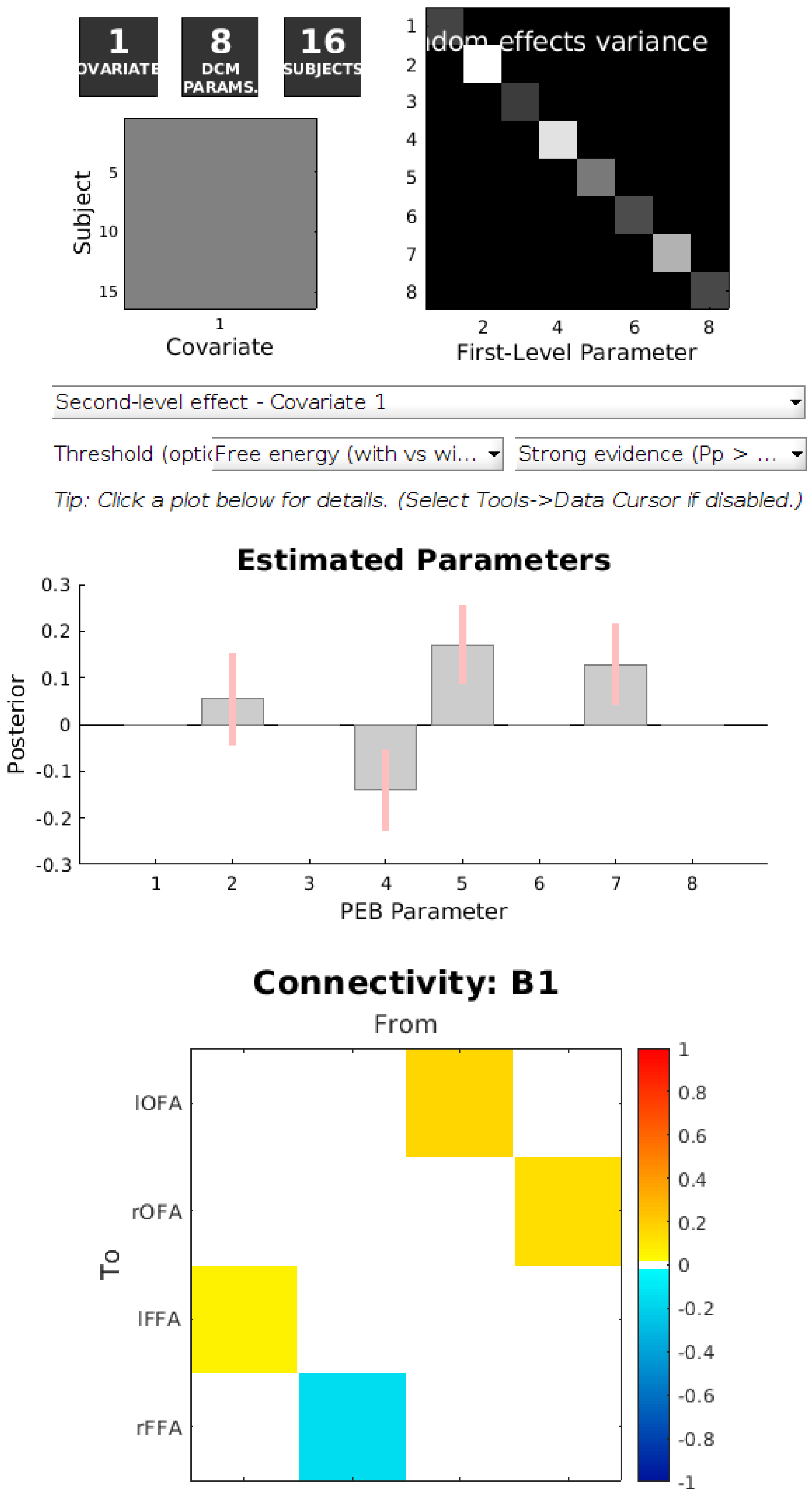


Figure 19: Binary BMC: Full vs Self-only model

Figure 19 shows results of the model comparison with 99% probability for the full model (and 1 for the self-only model), implying very high evidence that modulation of between-region connections is needed. The middle and bottom right panels show the parameter estimates for all B parameters in the full model, before and after BMA respectively. Since the probability of the full model is close to 1, these two estimates are virtually identical.

The second window titled 'PEB – Review Parameters' is similar to Figure 16, except that only parameters that varied across models are shown i.e. self-connections are not shown. These parameters are thresholded based on model evidence and their matrix-representations are shown in Figure 20.



Second-level effect - Covariate 1

Threshold (opti Free energy (with vs wi… Strong evidence (Pp > …

Tip: Click a plot below for details. (Select Tools->Data Cursor if disabled.)

Figure 20: Review of estimated parameters in BMA and thresholded connectivity estimates

The estimated BMA is stored by the module in 'fits/batch_gui/meg' with the name 'BMA_PEB_Full.mat'. The name of the PEB file being used by the module is prefixed with 'BMA_' implicitly. If we run the same module with a different model-space (GCM), this file will be overwritten. To prevent this and to make the naming scheme more descriptive, rename this file to 'BMA_PEB_Full_vs_Self.mat'. The model-space itself is stored in 'fits/batch_gui/meg/templates/GCMs/Full_vs_Self' as we had specified earlier.

4.6 Model Families

Instead of testing for modulation of all between-region connections versus none, we could alternatively ask whether one or more combinations of between-region connections are modulated. This involves multiple models with different combinations of between-region connections being modulated, i.e. switched 'off' or 'on' in the B-matrix. The model space therefore expands from just 2 models, like the previous BMC, to a larger number of models, which can be divided into different "families" (sets) according to which types of modulation are enabled. Below, we will distinguish models according to whether they contain self-modulations, modulations of "forward" connections (from OFA to FFA) or modulations of "backward" connections (from FFA to OFA).[2] This entails 8 models in total:

1. Modulation of forward, backward and self-connections
2. Modulation of forward and self, but not backward connections
3. Modulation of backward and self, but not forward connections
4. Modulation of self, but neither forward nor backward connections
5. Modulation of forward and backward, but not self-connections
6. Modulation of forward, but neither backward nor self-connections
7. Modulation of backward, but neither forward or self-connections
8. Modulation of neither forward, backward, or self-connections

By comparing the family of 6 models (models 1, 2, 3, 5, 6, 7) that include modulation of forward and/or backward connections with the family of 2 models (models 4 and 8) that do not contain forward or backward modulations (only with or without self-modulations), we can test a similar hypothesis to the previous section i.e., whether between-region connections are needed. The subtle difference is that the precise question now being asked is no longer whether all between-region connections are needed, but whether at least one type of between-region connection (forward and/or backward) is needed, and furthermore, whether this holds regardless of whether self-connections are also modulated.

4.6.1 Family-BMC for between-region connections

The first step involves creation of the eight template models for each model in the list above. This can be done by loading the full model, switching B-matrix connections off, and saving through the DCM GUI interface as shown previously. The same can be done more efficiently using simple MATLAB functions through the scripting interface. Steps:

1. Define model space as a GCM (a cell array of DCMs) by switching off relevant connections as done in section 4.4.2.1 and shown in Figure 21.

[2] Note we are always combining across hemispheres, though one could of course expand the model space to ask whether forward, backward or self-connections are needed in specific hemispheres. However, if one does not care about hemispheric differences, we are reducing the problem of model dilution by not considering models that differ in modulations between hemispheres.

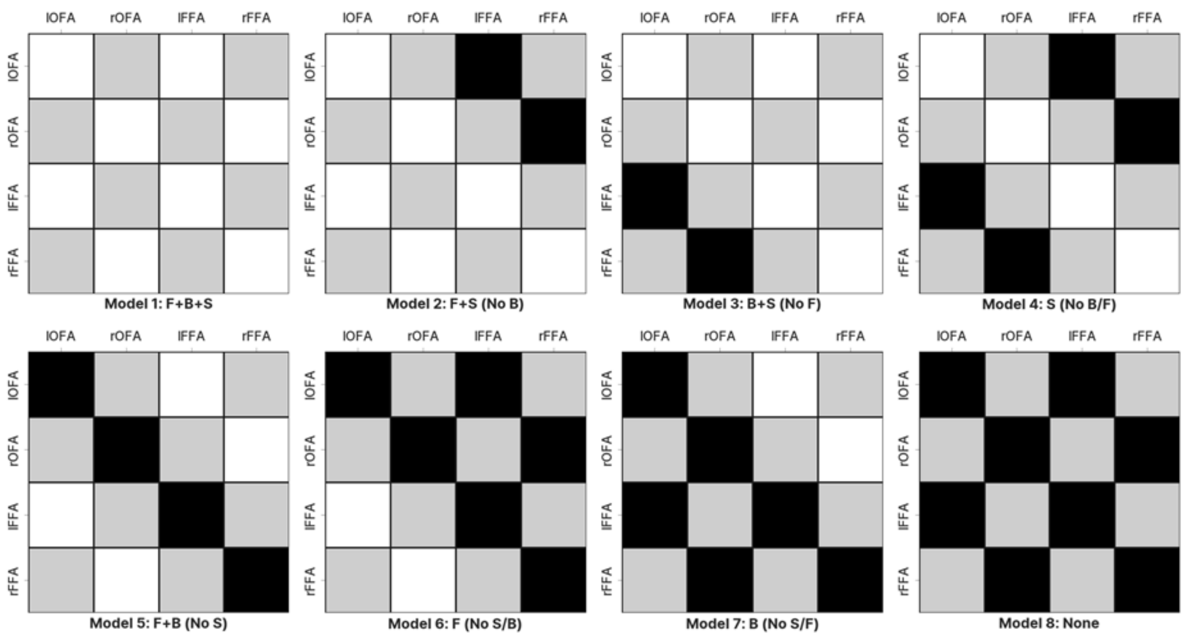


Figure 21: Model space for family-wise comparisons. White squares are connections that can be modulated (have value 1); grey and black squares cannot be modulated (both have value 0), but are distinguished here simply according to whether they never existed in the full model (grey) or need to be explicitly turned off (black).

2. Perform BMR using the estimated PEB to reduce this model space.

```
load('fits/batch_gui/meg/PEB_Full.mat');
[BMA, BMR] = spm_dcm_peb_bmc(PEB, GCM);
```

This BMR step only needs to be done once after the model space (GCM) has been specified. Running this will show a window with BMC for all 8 models along with parameter estimates. The model space in the top-left panel reflects the 8 models we defined earlier. Based on the middle left panel, model 5 with modulation of only backward connections but not forward or self-connections appears to have higher evidence (~53%) than others. This is also reflected in the parameter estimates in the BMA shown in panel ‘Commonalities’, which has ~1 posterior probability for both backward connections (parameters 5 and 7).

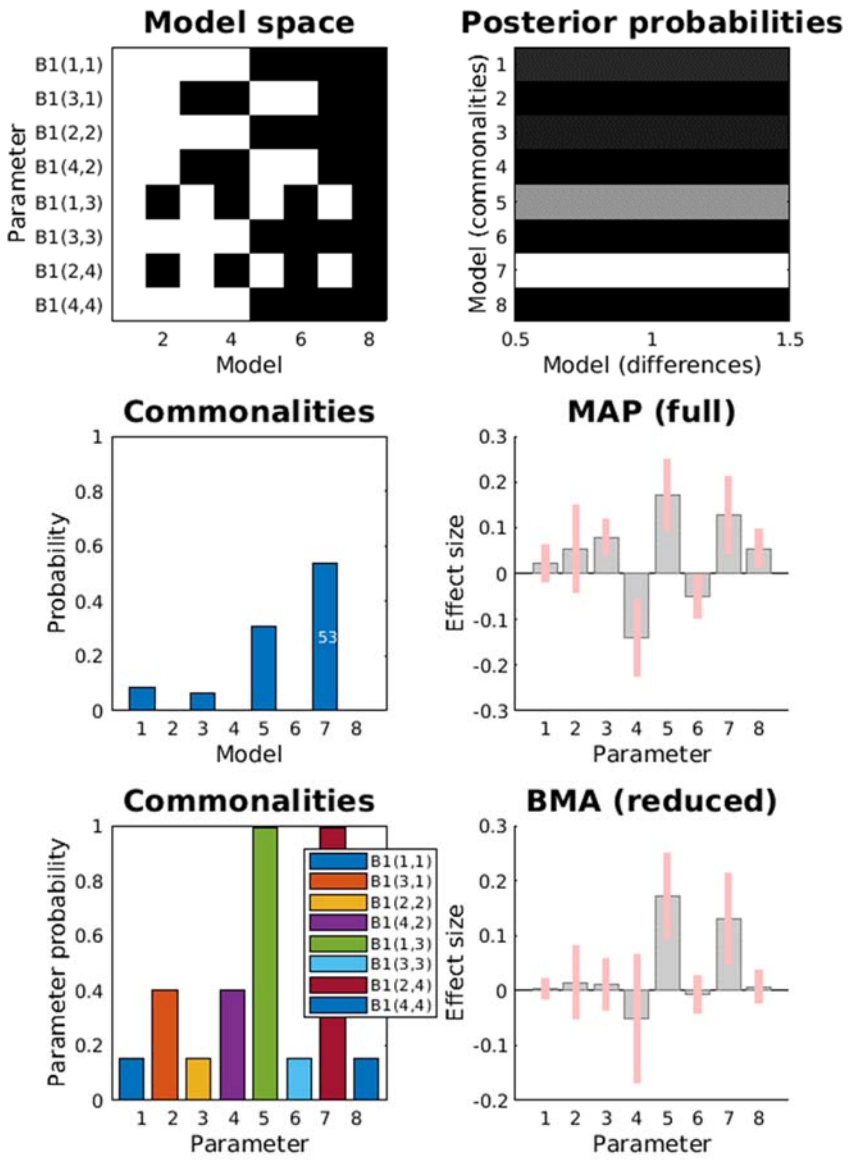


Figure 22: BMR for 8 models in model space

3. Assign models to families and perform family-wise BMC.
   Lastly, we group these models into families by specifying an integer for each of the models in our model space defined in the GCM above. Models with the same integer belong to the same family, which is identified by that integer. Run this line of code to create a variable defining families.
   ```
   families = [1, 1, 1, 2, 1, 1, 1, 2];
   ```
   This lets us specify which models to pool evidence from by grouping them under a family. We consider each family as equally likely, and perform inference at the level of families by running:
   ```
   [BMAf, fam] = spm_dcm_peb_bmc_fam(BMA, BMR, families, 'NONE');
   ```

where, BMA and BMR are variables obtained from the BMR step earlier. 'NONE' specifies that we do not want an averaged model. Running this line produces a figure with three panels.

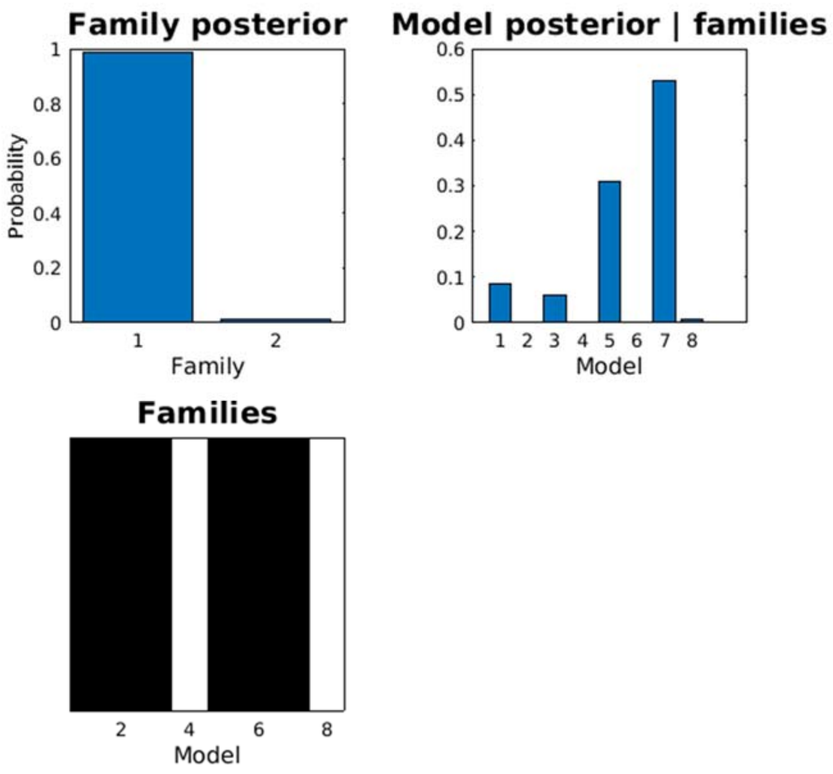


Figure 23: Family-wise comparison for modulation of between-region connections

The top left panel shows posterior probability of each family, where the family with modulation of at least one between-region connection has overwhelming evidence (~0.99) compared to the family with no modulation of between-region connections. The top right panel shows posterior probabilities of each model conditioned by the probability of each family. Comparing this panel to the middle left panel from the previous figure shows that the probabilities for each model have not changed after conditioning. This is due to the fact that all models in our model space with a non-zero posterior probability (i.e. models 1, 3, 5, and 7) belong to the winning family of models. The bottom left plot shows the grouping of models under each family.

4.6.2 Family-BMC for forward connections

Since we have very strong evidence for modulation of between-region connections, we now test whether a sub-group of between-region connections is modulated by Faces. We do this for both forward connections and backward connections, by specifying families for each of them.

Repeat step 3, with the 'families' variable set to [1, 1, 2, 2, 1, 1, 2, 2]; and run `[BMAf, fam] = spm_dcm_peb_bmc_fam(BMA, BMR, families, 'NONE');`

This will produce results for the family-wise comparison showing weak evidence (~0.6) for the family without any forward connections:

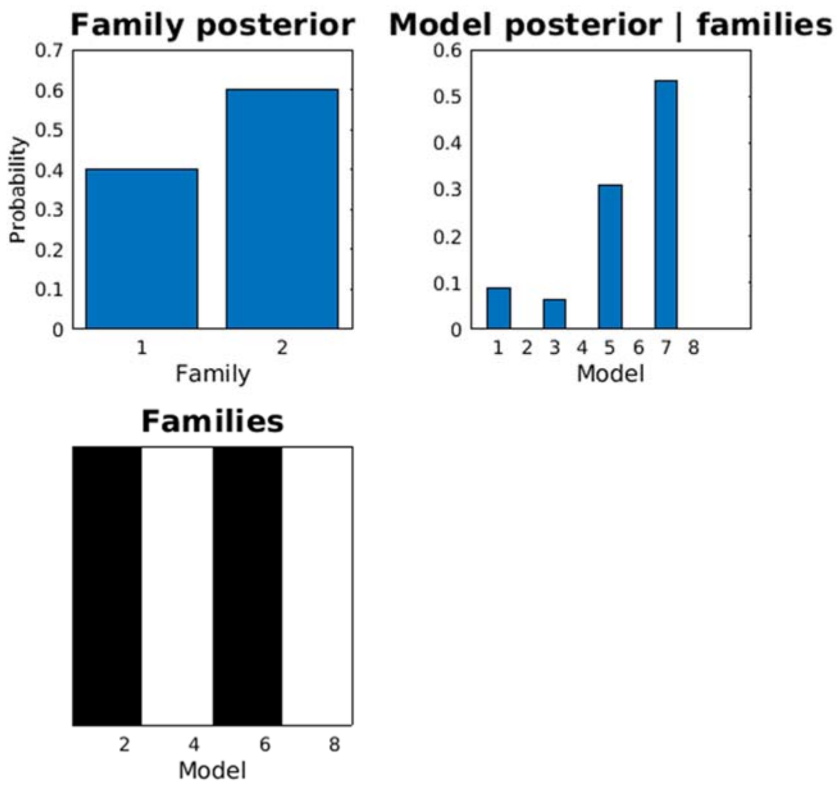


Figure 24: Family-wise comparison for modulation of forward connections

4.6.3 Family-BMC for backward connections

Similarly, for backward connections set the 'families' variable set to [1, 2, 1, 2, 1, 2, 1, 2]; and run

```
[BMAf, fam] = spm_dcm_peb_bmc_fam(BMA, BMR, families, 'NONE');
```

This will produce results for the family-wise comparison showing overwhelming evidence (~1) in favour of the family with modulation of backward connections:

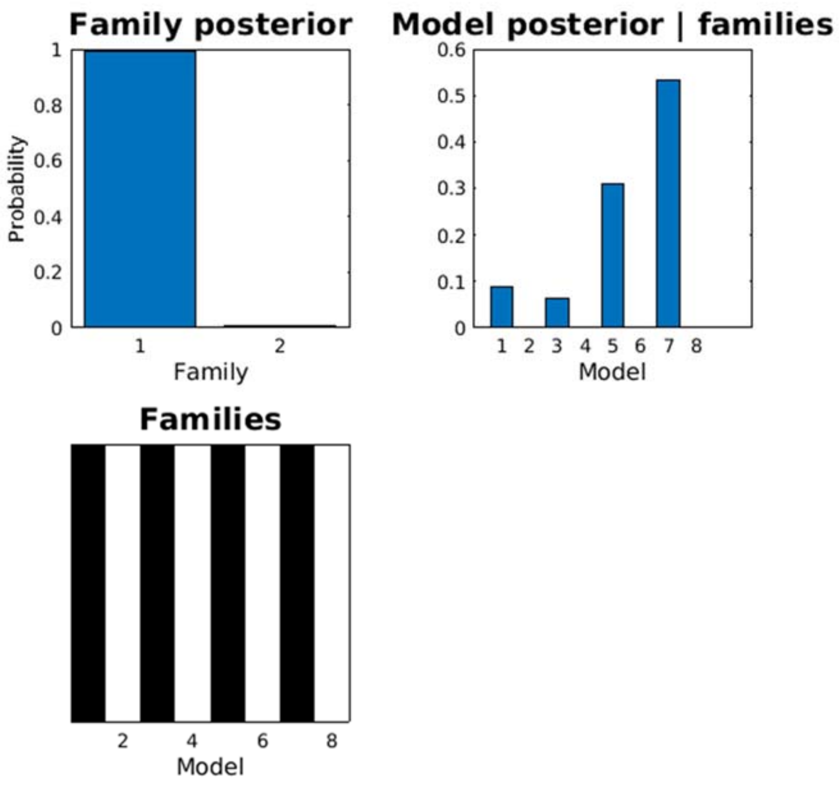


Figure 25: Family-wise comparison for modulation of backward connections

4.6.4 Family-BMC for self-connections

Lastly, in addition to modulation of between-region connections, we can also test whether self-connections are modulated by faces. To do so, set the 'families' variable set to [1, 1, 1, 1, 2, 2, 2, 2]; and run

```
[BMAf, fam] = spm_dcm_peb_bmc_fam(BMA, BMR, families, 'NONE');
```

This will produce results for the family-wise comparison showing moderate evidence (~0.85) in favour of family with no modulation of self-connections:

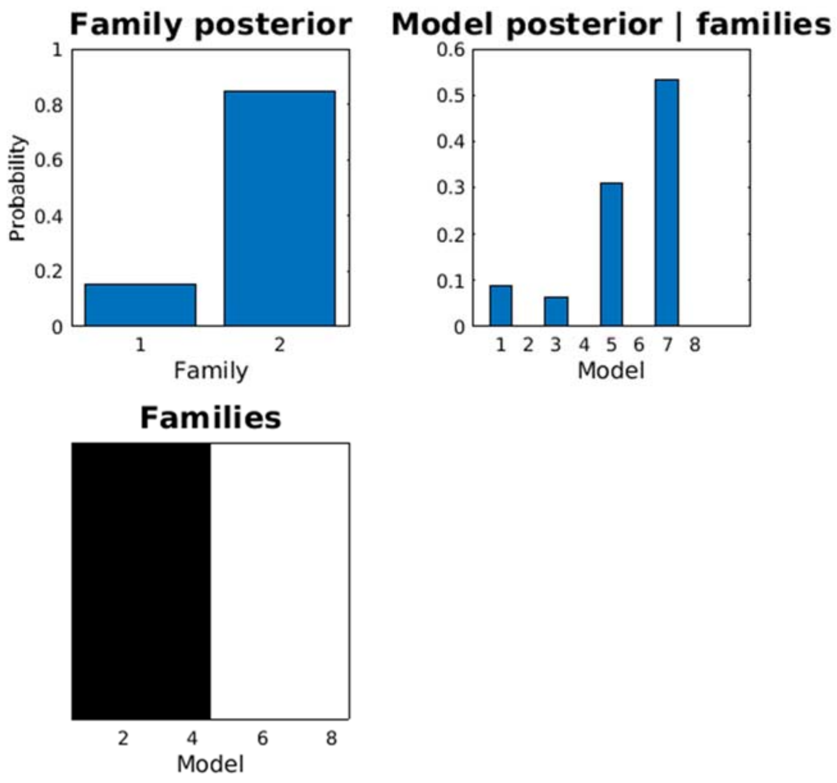


Figure 26: Family-wise comparison for modulation of self-connections

4.7 Covariates

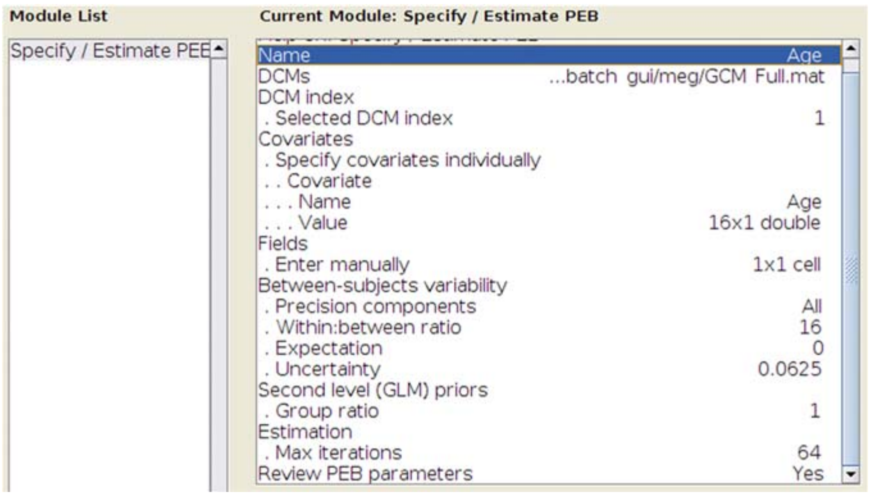
In this section, we demonstrate the addition of an age covariate for 2nd-level PEB estimation. Although we do not expect any effect of age on modulation of connections in these data (given the narrow range of adult participants from 23 to 31 years of age, with an average of 26 years), we conduct this exercise to highlight the key steps involved, since PEB was designed for testing differences between subjects (e.g, patients versus controls).

In a new batch, add a module to specify PEB by selecting 'SPM' → 'DCM' → 'Second level' → 'Specify / Estimate PEB'. In the options for this module, set 'Name' to 'Age' and select 'GCM_Full.mat' estimated earlier from 'fits/batch_gui/fmri' for 'DCMs'. Leave the 'DCM index' option as is.

For 'Covariates', select the option 'Specify covariates individually' from the grey box below. (Alternatively, a full design matrix with all covariates can be passed via this option). Then click on 'New: Covariate' in the grey box to create a pair of options – 'Name' and 'Value' for the covariate. Set 'Name' to 'Age'. For 'Value', enter the following numbers, one on each line in the text box that pops open on clicking 'Specify': `4.6, -1.4, 3.6, -0.4, -3.4, -0.4, 4.6, -0.4, 2.6, -3.4, -2.4, -2.4, -1.4, -2.4, 3.6, -1.4`

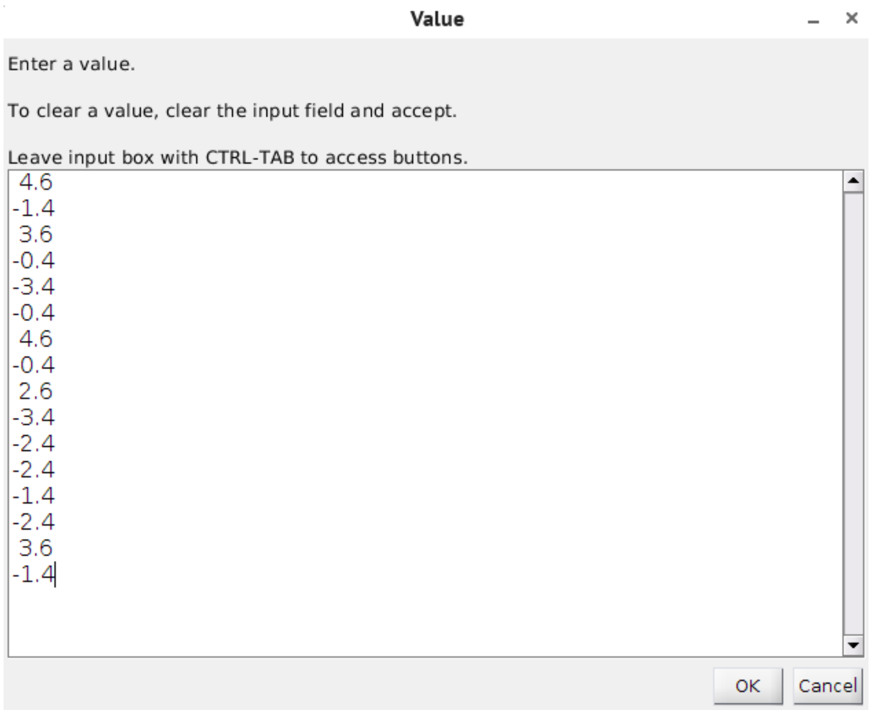
These numbers are the ages of the 16 subjects taken from the BIDS 'participants.tsv' file (available here: https://openneuro.org/datasets/ds000117/versions/1.0.5/file-display/participants.tsv), after subtracting the mean age (and to one decimal place).

This age covariate adds a second column to the design matrix (with the default first column still representing the group mean, and the mean correction of ages ensuring the two regressors are orthogonal, to ease interpretation).



Name	Age
DCMs	...batch gui/meg/GCM Full.mat
DCM index	
. Selected DCM index	1
Covariates	
. Specify covariates individually	
. . Covariate	
. . . Name	Age
. . . Value	16x1 double
Fields	
. Enter manually	1x1 cell
Between-subjects variability	
. Precision components	All
. Within:between ratio	16
. Expectation	0
. Uncertainty	0.0625
Second level (GLM) priors	
. Group ratio	1
Estimation	
. Max iterations	64
Review PEB parameters	Yes

Figure 27: Specification of Covariates in PEB



Enter a value.

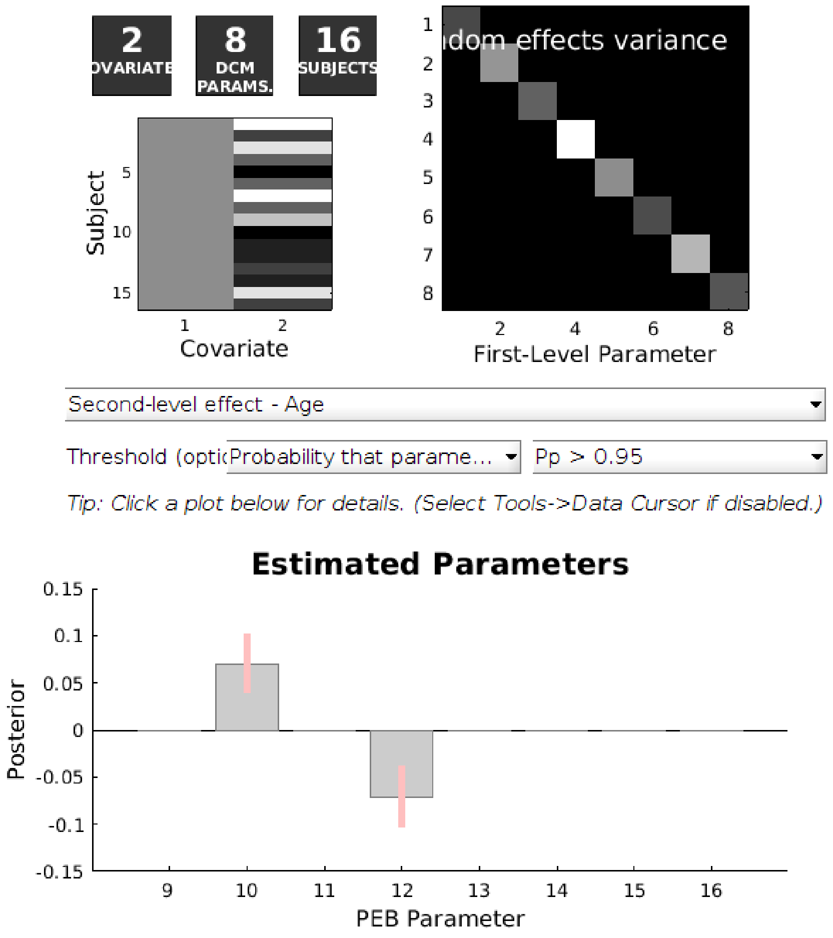
To clear a value, clear the input field and accept.

Leave input box with CTRL-TAB to access buttons.

```
 4.6
-1.4
 3.6
-0.4
-3.4
-0.4
 4.6
-0.4
 2.6
-3.4
-2.4
-2.4
-1.4
-2.4
 3.6
-1.4
```

Figure 28: Entry of Age covariate values

The review window now shows our design matrix in the top-left corner, with age as the second covariate. Since age was mean-centred, the first covariate represents mean modulation of connections across subjects. In addition to these common effects estimated at the group level ('Second-level effect – Commonalities'), the review window now has an additional option of viewing the effect of age by selecting ‘Second-level effect – Age’ from the drop-down menu. Thresholding these parameters based on posterior probabilities shows which of the connections modulated by faces were influenced by the age of participants. Although we do not demonstrate here, in order to carry out further inference and principled hypothesis testing based on model comparisons, any of the approaches illustrated earlier in section 4.4 can be applied to this group-level PEB estimate.



Second-level effect - Age

Threshold (optic Probability that parame... Pp > 0.95

Tip: Click a plot below for details. (Select Tools->Data Cursor if disabled.)

Figure 29: Review of PEB Parameters: 2nd level effect of Age

5. Discussion

We have demonstrated workflows for estimating group DCMs and PEB models for inference on connectivity parameters from MEG data. These workflows make use of SPM's graphical batch interface, and illustrate a systematic pipeline that begins with processed multimodal data for multiple subjects and ends with model-comparison based group-level inference about modulation of connections. We also show how to translate all steps of this pipeline from the graphical interface to batch scripts which allow for greater flexibility and easier chaining of multiple dependent steps. This tutorial accompanies a similar one for fMRI, for which notable differences exist in specifying individual DCMs, but the workflow for group-level PEB inference (i.e, once a GCM array has been specified) is identical.

Using our pipeline, we performed model comparisons at the group level on DCMs estimated from 16 subjects to test our hypotheses about effective connectivity of OFA and FFA during face perception. We first carried out a binary model comparison between a 'Full' model, in which we modelled the modulation of both between-region connections and self-connections due to faces, and a 'Self-only' model in which only self-connections but not between-region connections were modulated by faces. We observed greater evidence for the 'Full' model, suggesting that modulation of between-region connections is needed to explain the data better.

Next, instead of testing for modulation of all between-region connections versus none, we focused on specific groups of connections and tested whether one or more combinations of between-region connections are modulated. We created multiple models with different combinations of connections, grouped them under 'families' and then compared families of models. Each way of grouping models into families corresponded to a hypothesis. For the hypothesis of at least one between-region connection modulated by faces, we grouped all models with at least one forward or backward connection being modulated by faces into one family, and all models without into another. On comparison, we observed overwhelming evidence in favour of the family with at least one between-region connection, in agreement with the binary comparison of 'all' between-region connections versus 'none'. We then zoomed in, and reassigned models into families for testing whether at least one forward connection was modulated, and found no evidence in favour of modulation of at least one forward connection, regardless of other connections. Similarly, we reassigned models into different families for testing modulation of backward connections and self-connections. Based on these comparisons, we observed overwhelming evidence favouring the modulation of backward connections, but found little evidence in favour of modulation of self-connections.

The findings from our DCM analysis suggest that a backward flow of information from FFA to OFA drives the increased response to faces over scrambled images. This preliminary evidence favouring a role of backward connections from FFA to OFA during face processing is in agreement with extensive experimental findings of OFA-FFA connectivity in humans and non-human primates. The results differ somewhat from the companion tutorial on DCM for fMRI, where modulation of all connections (forward, backward and self) were needed, but the underlying neuronal models and timescales are quite different, such that there are good reasons why the connectivity captures different aspects of true neuronal interactions. Further fine-grained hypotheses could be tested using the demonstrated family-wise Bayesian model comparison approach to test for specific connections, including the role of lateral connections, as well as including other face-sensitive regions sensitive to faces such as the early visual cortex or the superior temporal sulcus (Lee et al., 2022).

6. References

David, O., Kiebel, S. J., Harrison, L. M., Mattout, J., Kilner, J. M., & Friston, K. J. (2006). Dynamic causal modeling of evoked responses in EEG and MEG. *NeuroImage*, *30*(4), 1255–1272. https://doi.org/10.1016/j.neuroimage.2005.10.045

Henson, R. N., Abdulrahman, H., Flandin, G., & Litvak, V. (2019). Multimodal Integration of M/EEG and f/MRI Data in SPM12. *Frontiers in Neuroscience*, *13*. https://www.frontiersin.org/article/10.3389/fnins.2019.00300

Lee, S.-M., Tibon, R., Zeidman, P., Yadav, P. S., & Henson, R. (2022). Effects of Face Repetition on Ventral Visual Stream Connectivity using Dynamic Causal Modelling of fMRI data. *BioRxiv*, 2022.07.05.498907. https://doi.org/10.1101/2022.07.05.498907

Litvak, V., Mattout, J., Kiebel, S., Phillips, C., Henson, R., Kilner, J., Barnes, G., Oostenveld, R., Daunizeau, J., Flandin, G., Penny, W., & Friston, K. (2011). EEG and MEG Data Analysis in SPM8. *Computational Intelligence and Neuroscience*, *2011*, 1–32. https://doi.org/10.1155/2011/852961

Moran, R., Pinotsis, D. A., & Friston, K. (2013). Neural masses and fields in dynamic causal modelling. *Frontiers in Computational Neuroscience*, *APR 2013*. https://doi.org/10.3389/fncom.2013.00057

Pereira, I., Frässle, S., Heinzle, J., Schöbi, D., Do, C. T., Gruber, M., & Stephan, K. E. (2021). Conductance-based dynamic causal modeling: A mathematical review of its application to cross-power spectral densities. In *NeuroImage* (Vol. 245). https://doi.org/10.1016/j.neuroimage.2021.118662

Spiegler, A., Kiebel, S. J., Atay, F. M., & Knösche, T. R. (2010). Bifurcation analysis of neural mass models: Impact of extrinsic inputs and dendritic time constants. *NeuroImage*, *52*(3). https://doi.org/10.1016/j.neuroimage.2009.12.081

The MathWorks Inc. (2018). *MATLAB R2018a* (version 9.4.0.813654 (R2018a)). The MathWorks Inc.

Wakeman, D. G., & Henson, R. N. (2015). A multi-subject, multi-modal human neuroimaging dataset. *Scientific Data*, *2*(1), 150001. https://doi.org/10.1038/sdata.2015.1

Zeidman, P., Jafarian, A., Corbin, N., Seghier, M. L., Razi, A., Price, C. J., & Friston, K. J. (2019). A guide to group effective connectivity analysis, part 1: First level analysis with DCM for fMRI. *NeuroImage*, *200*, 174–190. https://doi.org/10.1016/J.NEUROIMAGE.2019.06.031

Zeidman, P., Jafarian, A., Seghier, M. L., Litvak, V., Cagnan, H., Price, C. J., & Friston, K. J. (2019). A guide to group effective connectivity analysis, part 2: Second level analysis with PEB. *NeuroImage*, *200*, 12–25. https://doi.org/10.1016/J.NEUROIMAGE.2019.06.032