DCM PEB Example

Peter Zeidman

January 2019

Contents

1 Introduction

This tutorial introduces group-level connectivity analysis with DCM for fMRI and PEB. We'll use data from a previously published experiment (Seghier et al., Cerebral Cortex, 2011) which investigated the functional circuitry underlying individual differences in brain laterality.

1.1 The hypothesis and task

When people perform language tasks, there is asymmetry in the level of response of left and right frontal brain regions. This is often summarised by a score called the Laterality Index (LI). The LI varies across people, and the hypothesis of this experiment was that individual differences in LI could be explained by differences in the connectivity of specific frontal brain regions.

To address this, 60 subjects underwent fMRI scanning. On each trial of the experiment, subjects were presented with three visual stimuli, which were either pictures or words. Subjects had to decide whether one item was semantically related to the other two. There were also baseline trials, during which subjects had to decide whether one item was perceptually related to the other two. There were therefore two experimental factors at the within-subject level: stimulus type (words or pictures) and task (semantic or perceptual matching). The between-subject factor was LI, which is the focus of our analysis.

1.2 Model design

We will model these data using a four region Dynamic Causal model (DCM), shown in Figure 1. This will include four brain regions: left ventral Frontal cortex (lvF), left dorsal Frontal cortex (ldF), right ventral Frontal cortex (rvF) and right dorsal Frontal cortex (rdF). These regions were identified in an initial SPM analysis. At the onset of each trial, the network is 'pinged' by the driving input (blue dotted arrows labelled C in Figure 1) and activity flows around the network. Each region's sensitivity to inputs from the rest of network is modulated by Pictures and Words (red and green lines respectively). These modulatory effects are quantified by parameters in matrix B of the neural model. The neural model also has parameters which quantify the sensitivity of each region to driving input (i.e. the 'ping' of activity for each trial), quantified in parameter matrix C.

The initial steps - performing an SPM analysis and extracting timeseries - has already been done for you. We will start by specifying the DCM network model for every subject. We will then take the modulatory parameters (B) and driving input parameters (C) to the group level and test hypotheses about the commonalities and differences across subjects.

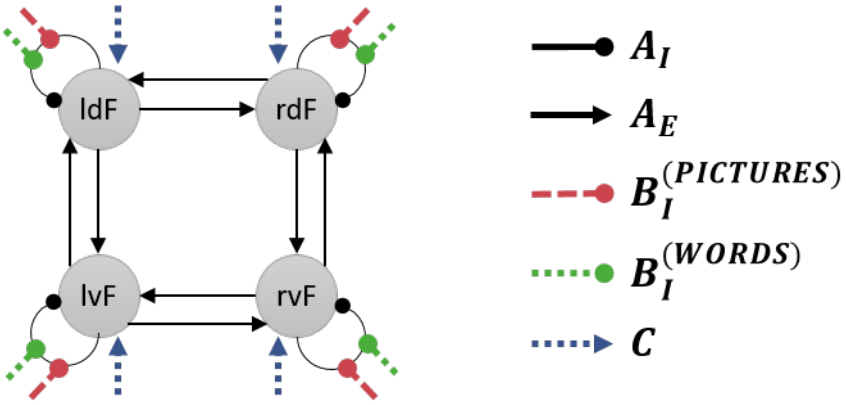

Figure 1: Design of the DCM model.

2 Getting the data

The data are available to download from Github. After downloading the dataset, save it in a convenient location, unzip it and change the Matlab working directory to the location of the files (you should see a file named design_matrix.mat on the left hand side of your Matlab window).

3 Analysis using the GUI

There are two ways to use DCM: via the Graphical User Interface (GUI), or using scripts. We will first walk through an entire analysis using the GUI, as this is a good place to start if you haven't used DCM before. We'll then walk through the scripting approach, which will also enable us to demonstrate working with large model spaces involving families of models.

3.1 First level analysis: DCM for fMRI

3.1.1 DCM Specification

We will now specify a DCM for the first subject, matching the design in Figure 1.

- In Matlab, use the file selector on the left hand side to change the working directory to the GLM folder for subject 1. i.e. \GLM\sub-01\
- Launch SPM by typing: spm fmri and press enter.
- Click the big Dynamic Causal Modelling button. In the grey window which appears, click 'Action' and then specify.
- In the file selector, click the SPM.mat file on the right hand side and then click Done. This provides DCM with the timing of the experimental conditions.
- You'll be asked for a name for the new DCM. Type: full and press enter.
- You'll be asked to select the VOIs (volumes of interest) for this subject. These are the timeseries for each brain region, which have already been prepared. The order is important - the same order will be used for the regions in the DCM. For consistency with this tutorial, select in order: lvF, ldF, rvF, rdF and click Done.
- Now you are asked which experimental conditions to include. For Task, Pictures and Words click yes.
- For VOI timings, type 3.6 3.6 3.6 3.6 and press enter. This configures DCM's slice timing model. Typically you would leave this on the default of half way through the volume (1.8s in this case), but here we set it to the end of the volume, for consistency with a previous publication using this dataset.
- For Echo Time (TE), type 0.05 and press enter.
- You are now asked to set certain options for the model. Select:
  - bilinear
  - states per region: one
  - stochastic effects: no
  - centre input: yes
  - fit timeseries or CSD: timeseries
- You are now asked which connectivity parameters you want switched on (free to be informed by the data) and which you want switched off (fixed at their prior expectation of zero). These are connections in matrix A from the DCM equation, which is the average connectivity over experimental conditions. The self-connections (on the leading diagonal) are always switched on, and you can select which between-region connections to include. Switch the connections on according to Figure 2 and then press done. (Tip: holding your mouse pointer over a button will identify the connection.)

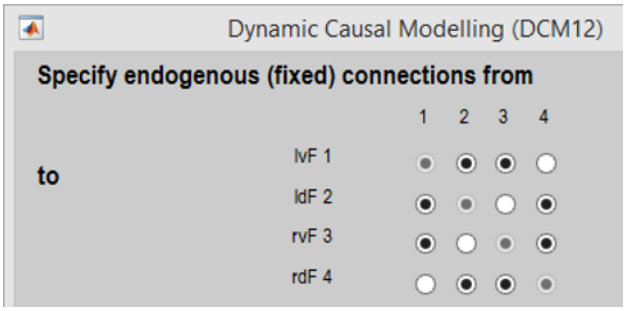

Figure 2: DCM specification for matrix A.

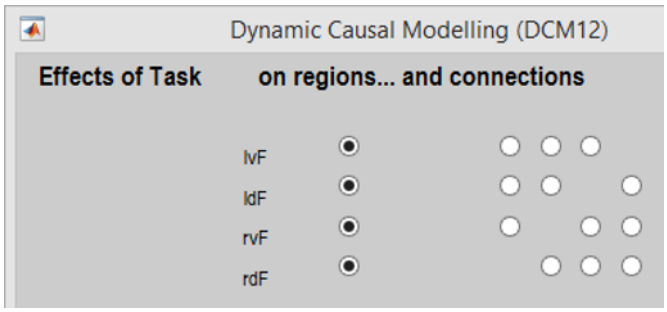

Figure 3: DCM specification for input Task.

- Next you are asked about each of the experimental inputs, starting with Task. The buttons on the left are the driving inputs (matrix C), and the buttons on the right are the modulatory inputs (matrix B) which increase or decrease the strength of particular connections. The connections you switched off in the previous step are hidden. Set Task to drive all regions by selecting the buttons on the left hand side, as shown in Figure 3 and then press done.
- Next you are asked about Pictures. We'll allow this experimental condition to modulate specific connections, rather than serve as driving inputs. Using the buttons on the right, click the four buttons on the leading diagonal. This will allow Pictures to modulate the self-connections of each region as shown in Figure 4 (top), then press done.
- Finally, you are asked about the Words condition, which we'll also allow to modulate each region's self-connection. Set the buttons as shown in Figure 4 (bottom), then press done. You will receive a polite 'thank you' and a file called DCM_full.mat will have been created in this subject's GLM directory.

3.1.2 Alternative DCM

Hypotheses are tested in DCM by switching off certain parameters (e.g. connections) and seeing how this effects the model evidence (approximated by the free energy). In the paper we had 112 alternative models, however, to specify large numbers of models it is best to write a Matlab script, which is detailed in the second part of this guide. To demonstrate DCM using only the GUI, we will specify just one alternative model so that we can ask the question: is left dorsal frontal cortex (ldF) needed to explain individual differences in laterality index (LI)? To test this, we'll specify a reduced model where modulatory parameters relating to region ldF have been switched off, as illustrated in Figure 5, right.

- In the main SPM window, click the big Dynamic Causal Modelling modelling then click specify.
- In the file selector, select the SPM.mat for subject 1 and click Done.

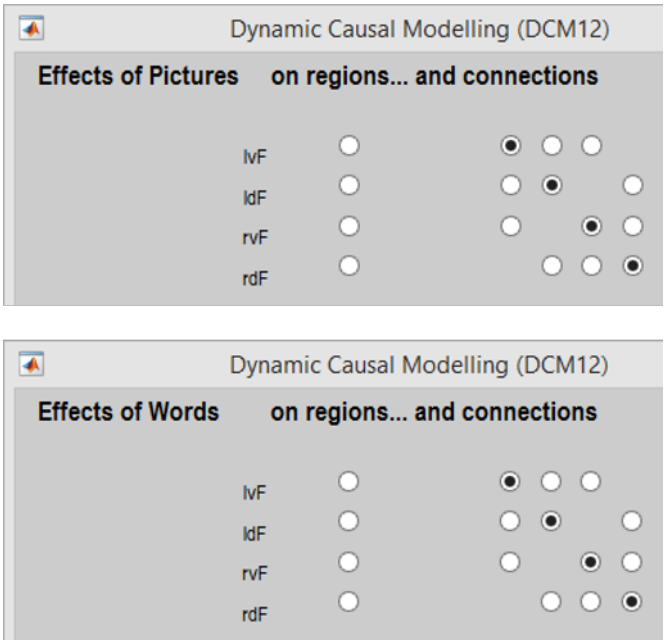

Figure 4: DCM specification for inputs Pictures (top) and Words (bottom).

- For name, type: no_ldF_modulation then press enter.
- Select the 4 ROIs in order: lvF, ldF, rvF, rdF then click Done.
- Click yes to include Task, Pictures and Words
- For VOI timings, type 3.6 3.6 3.6 3.6 and press enter.
- For Echo Time, type 0.05 and press enter.
- Click the buttons as follows. Modulatory effects: bilinear, states per region: one, stochastic effects: no, centre input: yes, fit timeseries or CSD: timeseries.
- Switch on A-matrix connections as per Figure 2.
- Set Task to be the driving input for all regions, as per Figure 3 and press done.
- Unlike before, set the effects of Pictures and Words to modulate the self-connections EXCEPT for region ldF, which we'll switch off, as per Figure 6. Then press done.

A second DCM will have been created for this subject named DCM_no_ldF_modulation.mat.

3.1.3 Replicating across subjects

Having specified two DCMs for a single subject, we can now use these as templates to specify the same two DCMs for every subject. All subjects' models will be the same, except the timeseries and timing of the experimental input, which will be customized for each subject.

- In the main SPM window, click Batch (if the main SPM windows isn't visisble, launch it by typing: spm fmri and press enter).
- From the menu at the top of the Batch Editor, click SPM → DCM → DCM specification → DCM for fMRI → Specify group. Fill in the batch as follows (also shown in Figure 7):
  - Output directory - double click Output directory to bring up the file selector. We will choose to store the group DCM file (GCM) in a folder called 'analyses'. Click the two little dots (..) on the left hand side twice, to go up two directories, and single click 'analyses'on the right hand side. Then press Done.

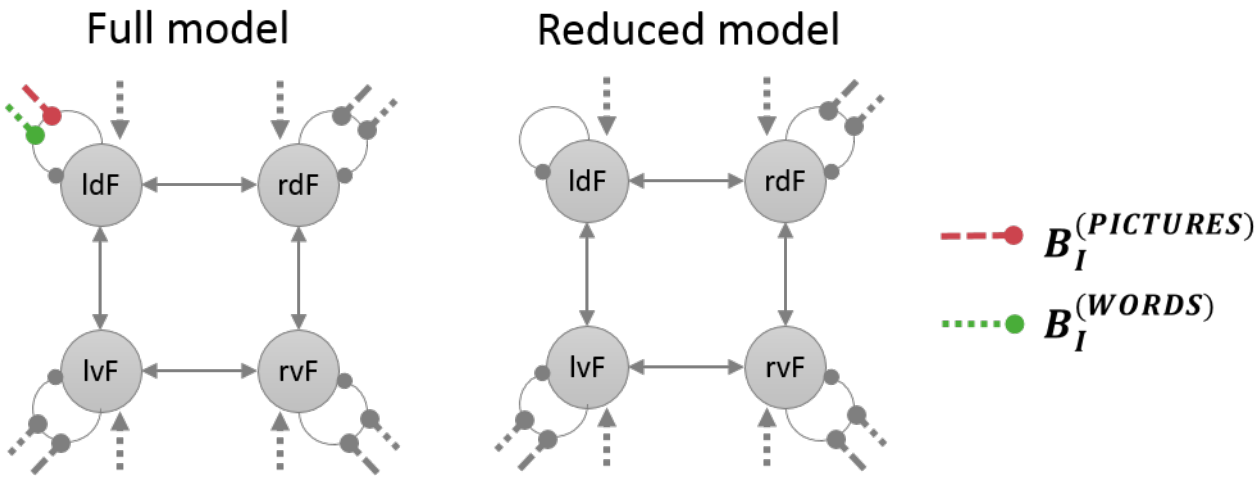

Figure 5: Two models, which differ in the modulation of region ldF.

- Output name - double click, then type two_models and press OK.
- Full DCM - double click, then in the file selector, select the full DCM we made earlier for subject 1. It is named DCM_full.mat . Select this on the right hand side, then press Done.
- Alternative DCMs - double click, then select the reduced DCM (DCM_no_ldF_modulation.mat) we created earlier and press Done.
- SPM.mat files - We will now select all subjects' SPM.mat files, which contain the timing information for each subject. Double click, then in the file selector, navigate to the GLM directory (click .. on the left hand side). You should now see a list of all subjects on the left hand side. Press the small 'Rec' button. This will search through all subjects and pick out their SPM.mat files. Check that 60 are selected at the bottom of the file selector window and press Done.
- Regions of interest - Now we'll select the timeseries (VOIs) for each subject. Click Regions of interest then click New: Region (VOI files). Do this four times, so you get four entries in the batch which say 'Region (VOI files)'. Double click the first one and navigate to the GLM folder which contains all the subjects. Do this by clicking the two small dots (..) on the left hand side of the file selector. We're going to select lvF timeseries for every subject. Delete the text in the Filter box on the right hand side (^VOI_.*\.mat$) and type: lvF. Then click Rec, which will search through all the subjects' folders selecting the 60 lvF files. Press Done.
- Next we'll select the VOI files for the second region (ldF). Double click the second 'Region (VOI files)' entry, navigate to the GLM folder by clicking .. on the left hand side, delete the contents of the filter box and type: VOI_ldF. Then click Rec. This should find the ldF timeseries from all 60 subjects - press done. Then repeat this process for regions rvF then rdF. You should now have timeseries for all 4 regions of interest, and the batch should look like Figure 7.

- Click File → Save batch and save it somewhere safe, with a name like replicate_dcm_batch.mat. Then press the green play button to run it.

If all has gone to plan, you'll see two DCMs in each subject's GLM folder named DCM_two_models_m0001.mat (the full model) and DCM_two_models_m0002.mat (the reduced model with no effect of pictures on region ldF). Their filenames will also be collated into a single cell array and saved in a file named 'GCM_two_models.mat' in the 'analyses' folder.

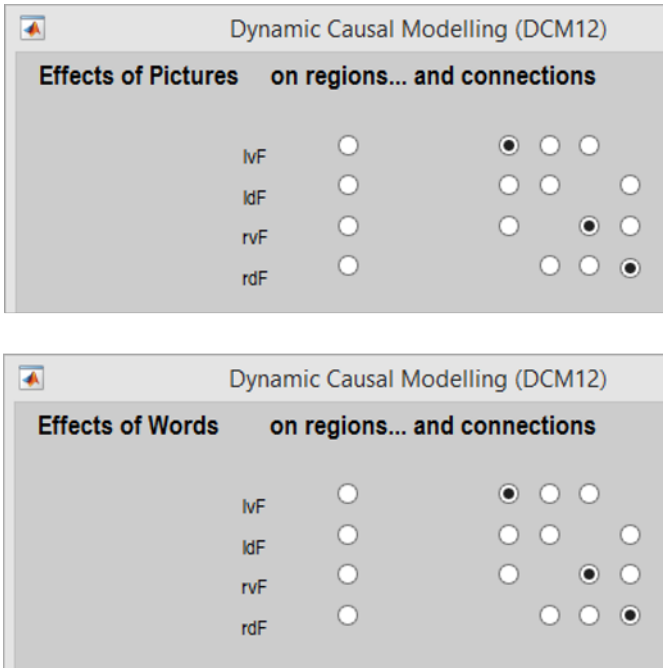

Figure 6: Specification of alternative model without modulation of region ldF.

3.1.4 Estimating

Having specified two DCMs for every subject, let's now fit them to the data. In the Batch editor, click File → New Batch. Then click SPM → DCM → DCM estimation. Fill out the batch as follows:

- Select GCM_*.mat - double click, then navigate to the analyses folder by using the two dots (..) on the left hand side. Select the file named GCM_two_models.mat we created earlier. Then press Done. The file we selected contains a cell array of DCM filenames.
- Output - single click Output, then click 'Only save individual DCM files'.

The other items can be left on their defaults. Save this batch by clicking File, Save Batch. If you want, you can estimate the models by pressing the green play button, which will take about 10 minutes on a high-spec PC with parallel computing enabled (see Tip below). This will fit each full model to the subject's data, and then analytically derive the evidence and parameters of each nested model (DCM_no_ldF_modulation.mat) using Bayesian Model Reduction (BMR). The results will overwrite the DCM files in each subject's folder.

Tip: depending on your computer, you might be able to speed up DCM estimation by using parallel computing. You can switch this on by editing spm_dcm_fit.m and changing the variable named 'use_parfor' on line 24 from false to true.

3.1.5 Diagnostics

Having completed the estimation of the first-level DCMs, it is a good time to perform some diagnostics on the models. First, in Matlab, change to the analyses directory and load the GCM file containing the filenames of all subjects' DCMs (GCM_two_models_pre_estimated.mat), by double clicking on it. Then type or paste the following code:

Listing 1: DCM for fMRI diagnostics

```
spm_dcm_fmri_check(GCM);
```

The output of this command is shown in Figure 8. This is a graphical representation of the GCM file, where the long coloured bar indicates the explained variance of each DCM, and the

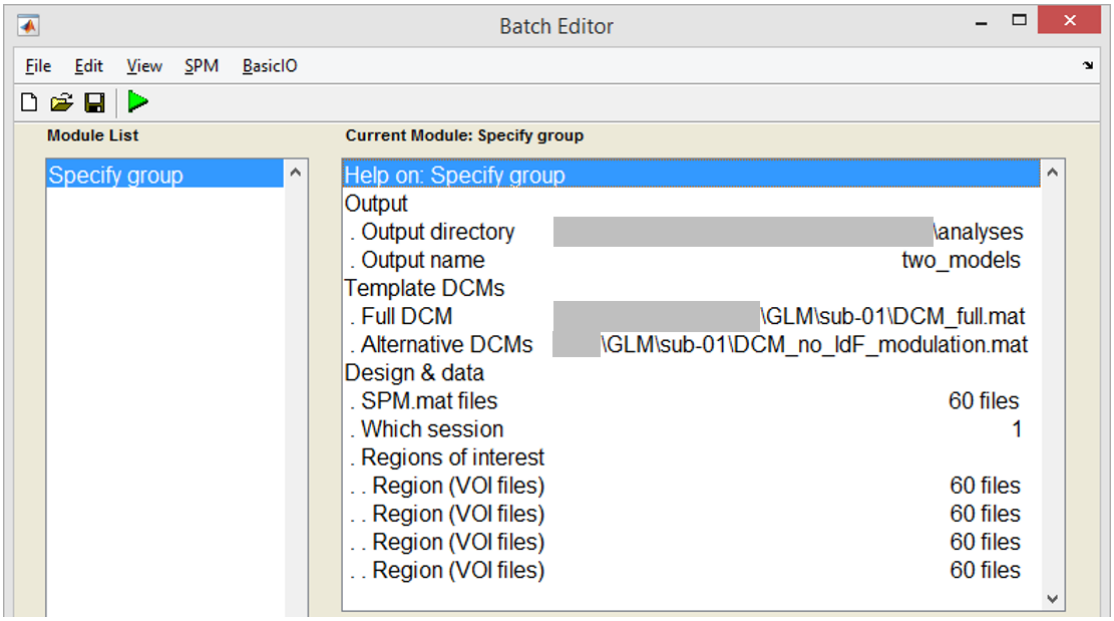

Figure 7: Batch for replicating an fMRI DCM over subjects.

two columns are the two models per subject. Only the first column (the full model) will have the explained variance calculated. Here we clicked on the model for Subject 37, who had explained variance 18.86%. Clicking the different subjects shows that some subjects' models explained more variance than others. In a real experimental situation, there would be a few options for how to deal with subjects with poor explained variance (less than 10%) - see Section 5 at the end of this guide for more detail.

After clicking on a subject's full model (left hand column of the figure), click the Diagnostics button, which shows the window in Figure 9. The top plot shows the predicted timeseries (solid lines) and the data (dotted lines). Here we check that the predicted timeseries are not a flat line - i.e. some dynamics were captured by the model. The bottom-left plot shows the estimated parameters from matrix A (the average or baseline connectivity over experimental conditions). This lets us check that the data have informed the model, such that the neural parameters have confidently deviated away from zero (the prior). The bottom-right plot shows the correlations between the parameters as well as the estimated number of parameters (degrees of freedom) in the model, after taking these correlations into account. This is a little harder to apply as a diagnostic, but may be useful for the development of new models.

This completes the first level analysis. The next step is to summarise the individual subjects' DCM parameters at the group level and test hypotheses.

3.2 Second level analysis: PEB

3.2.1 PEB model specification

The first level analysis provided estimates of the parameters for every subject. Of particular interest here are the B-matrix parameters (modulatory inputs). We will now form a Bayesian General Linear Model (GLM) of the individual subjects' DCM parameters and use this to test hypotheses.

- Load the between-subjects design matrix into the Matlab workspace. To do this, go to the main Matlab window and change to the directory where you downloaded the example dataset (if you're currently in the analyses folder, then it's one folder up). On the left hand side, find the file named design_matrix.mat and double click on it to load it into Matlab. You will get a variable in the workspace named X (the design matrix) and another variable called labels (the name of each column in the design matrix: Mean, LI, Handedness,

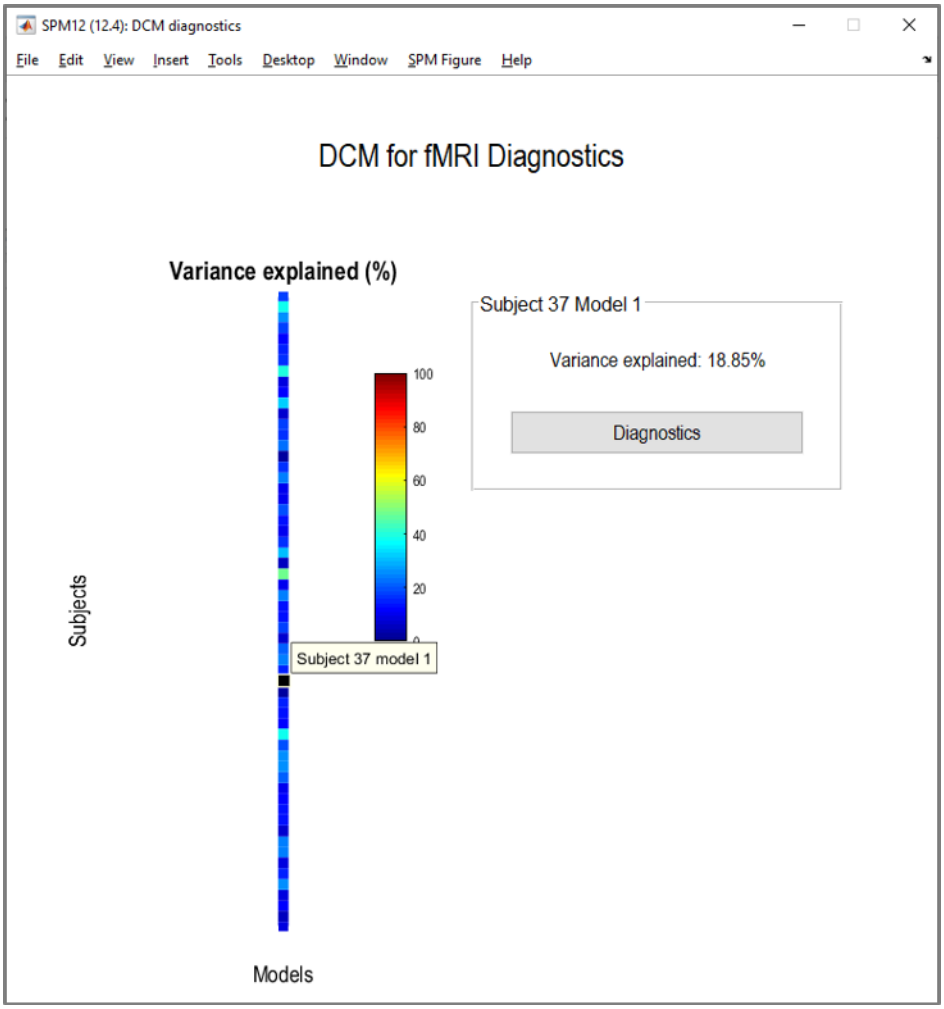

Figure 8: Output of spm_dcm_fmri_check(GCM)

Gender, Age).

- Go to the main SPM window and click Batch. Then click SPM → DCM → Second level → Specify / estimate PEB. Fill out the batch as follows (also shown in Figure 10):
  - Name - This is a name for the analysis. Call it: B. This is because we're going to include all parameters from each subject's DCM matrix B (modulatory inputs).
  - DCMs - Double click, then navigate to the analyses folder and select the file named GCM_two_models_pre_estimated.mat. This contains the filenames of all subjects' DCM files.
  - Selected DCM index - Recall that we specified 2 DCMs per subject, a full model and a reduced model. We are going to build the PEB model using the parameters from each subject's full model (model 1). So leave this on the default value of 1.
  - Covariates - This is where we specify the columns of the between-subjects design matrix. Click Covariates then 'Specify design matrix', then double click 'Design matrix'. Type: X then click OK. This will read the design matrix we loaded earlier. You should see the numbers from the design matrix, in 5 columns with 60 rows. Then single click on 'Covariate names', click 'New: Name', double click 'Name', type 'Mean' then click OK. Repeat this to give names for the remaining four covariates: LI, Handedness, Gender, Age.

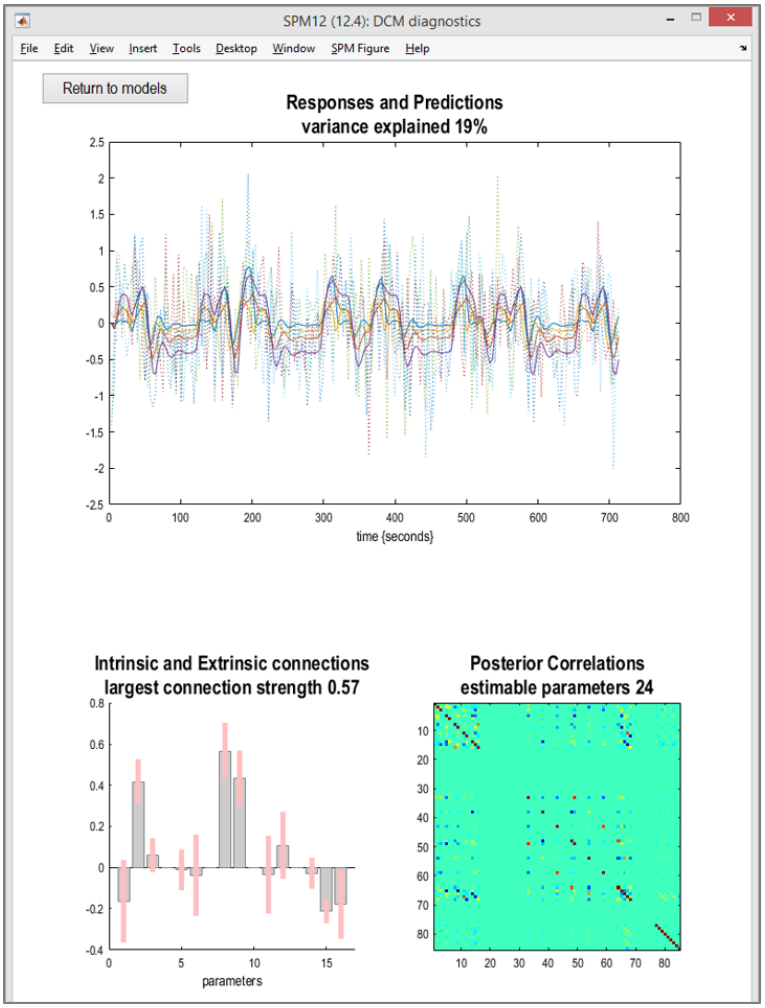

Figure 9: Output of spm_dcm_fmri_check(GCM) after selecting a subject and clicking the Diagnostics button.

  - ∗ **Design matrix tips**: The first covariate should be all ones, to represent the commonalities across subjects. If there are between-subjects differences (covariates) to include, these should appear in subsequent columns, with the most interesting between-subjects difference in the second column of the design matrix. Optionally, you might want to mean-centre columns 2 to K, where K is the number of columns. This will give the first column the interpretation of the mean connectivity across subjects. If they are not mean-centred, the first column will represent the baseline or intercept. The code for mean-centring is: X(:,2:K)=X(:,2:K)-mean(X(:,2:K)). Note that for every regressor in the design matrix, parameters will be added for every DCM connection. So try to keep the number of regressors to a minimum, to keep model estimation tractable.
  - Fields - We're only going to take parameters from DCM matrix B to the group level. Click Fields, then click 'Enter manually'. Double click 'Enter manually' and type: {'B'} including the curly brackets, then press OK.
  - Max iterations - Scroll to the bottom of the batch to find this option, which controls the maximum number of iterations of the model fitting procedure. This example dataset is quite large and we found that the default of 64 iterations was not enough. Set this to 256.

- Now save the batch by clicking File, Save Batch and give it a name like 'peb_spec_batch.mat'.

Then run the batch by pressing the green play button. The won't show any windows, but will create and estimate the group-level PEB model and store the results in a file called PEB_BC.mat within the analyses folder.

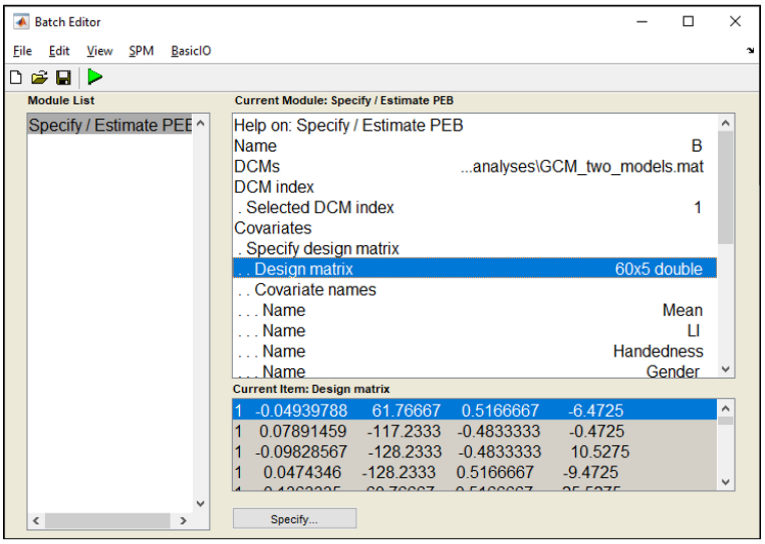

Figure 10: Batch to specify and estimate second level model (PEB).

The PEB model created above (PEB_BC.mat) contains parameters which quantify the effects of 5 covariates (group average, LI, handedness, gender and age) on each of 12 DCM connectivity parameters (Pictures, Words and Task for each of 4 regions). So that's 12 x 5 = 60 parameters in total in the PEB model. To test hypotheses, we will compare the evidence for the PEB model with different combinations of these 60 parameters switched on or off.

3.2.2 Model comparison: automatic search

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 with specify and compare hundreds of candidate reduced PEB models, in which different combinations of parameters have been switched off. This process is performed rapidly using a method called Bayesian Model Reduction (BMR). It will then average the parameters (connection strengths) from the best reduced models - this is called Bayesian Model Averaging (BMA).

- In the batch editor, click File → New Batch then click SPM → DCM → Second level → Search nested PEB models. Fill it out as follows:
  - Select PEB file - In the file selector, navigate to the analyses directory and choose the PEB_B.mat file we created earlier. This is the full PEB model which will be pruned by the search.
  - DCMs - Select the GCM_two_models.mat file we created earlier, from the analyses folder. (This is only used to make the graphical output more readable.)
  - Null prior variance - This determines the null hypothesis for each connectivity parameter - i.e. what prior variance constitutes a connection being 'switched off'. For consistency with the paper, set this to 0 (zero).
- Save the batch (e.g. with a name like 'search_batch.mat') and press the green play button. This will produce three windows as well as a file called BMA_PEB_B.mat in the analyses folder containing the results. Let's go through each of the windows.

The window titled 'BMR - all' (Figure 11) details the 256 candidate PEB models from the final iteration of the automatic search. The top left plot shows the log model evidence for each PEB model and the top right shows these values converted to posterior probabilities. The second row shows the parameters of the PEB model before the search (left) and after the search (right). Clearly, many parameters have been pruned away because they did not contribute to the model evidence (free energy). We will return to the identity of these parameters shortly. The bottom left plot shows the parameters that were switched on (white) and switched off (black) in each model from the final iteration of the search. For example, B(2,2,2) - the modulatory effect of Pictures on region ldF - was switched off in the first 128 models and switched on in the second 128 models. Finally, the bottom right plot shows the posterior probability for each PEB parameter. This is computed by comparing the evidence for all models (out of the final 256) which had the corresponding parameter switched on, versus all models which had that parameter switched off.

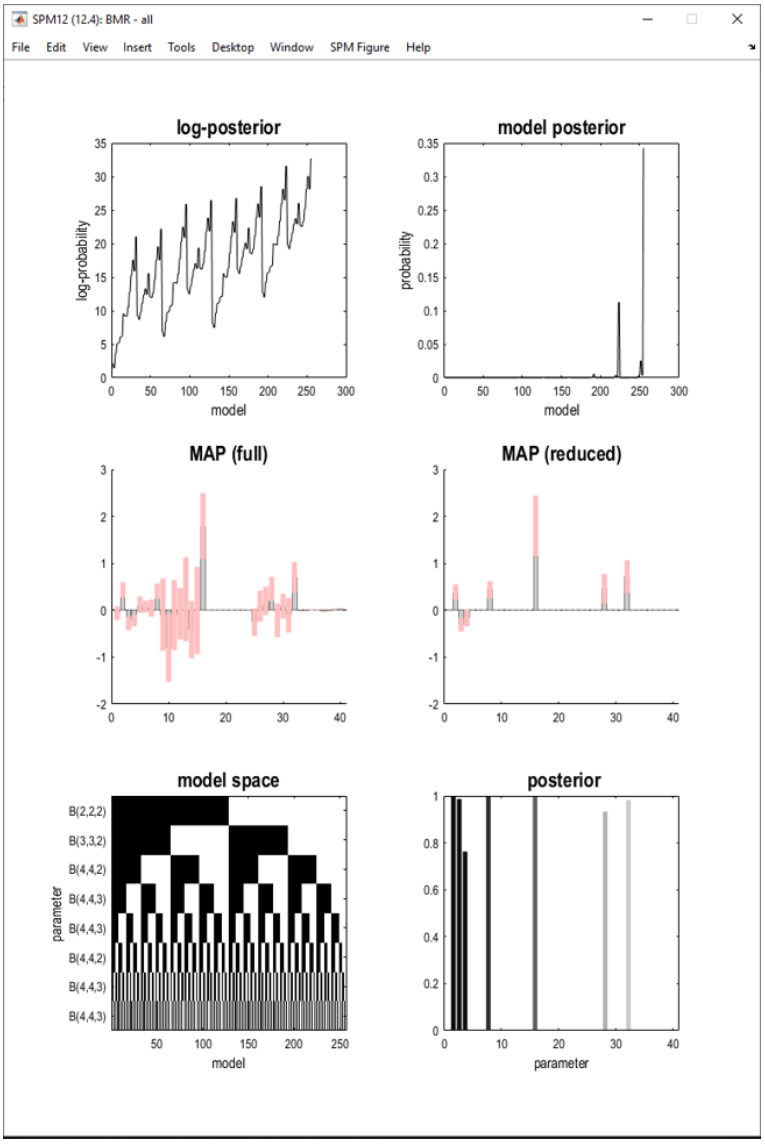

Figure 11: Model space display after performing an automatic search over reduced PEB models. (Generated by spm_dcm_bmr_all.m which is called by spm_dcm_peb_bmc.m)

With a large model comparison such as this, it is unusual to find one model that is the overall winner. Instead of considering the individual models, it is generally more informative to consider the BMA - the weighted average of the parameters over models. The window titled BMC (Bayesian Model Comparison, Figure 12) shows this average, with plots organised into three

columns. The first column relates to the first eight PEB parameters, which are the commonalities across subjects (group average). Each bar corresponds to one DCM connection. The second column shows the PEB parameters relating to the group difference, LI. The third column shows parameters relating to the next group difference, Handedness. The first row shows the parameters from the PEB model, and the second row shows the parameters after the model search and the BMA. It is clear from the middle plot that only one effect of LI has survived (parameter eight) and the plot to its right shows that no effect of handedness has survived. The bottom row shows the posterior probability for each parameter (as described above). The legend at the bottom right identifies the eight parameters used in all the plots, and it shows that the only surviving LI parameter is B(4,4,3). The first number is the target region, the second number is the source region, and the third number is the experimental condition. So that's the modulation of the self-connection on region 4 (rdF) by experimental condition 3 (Words).

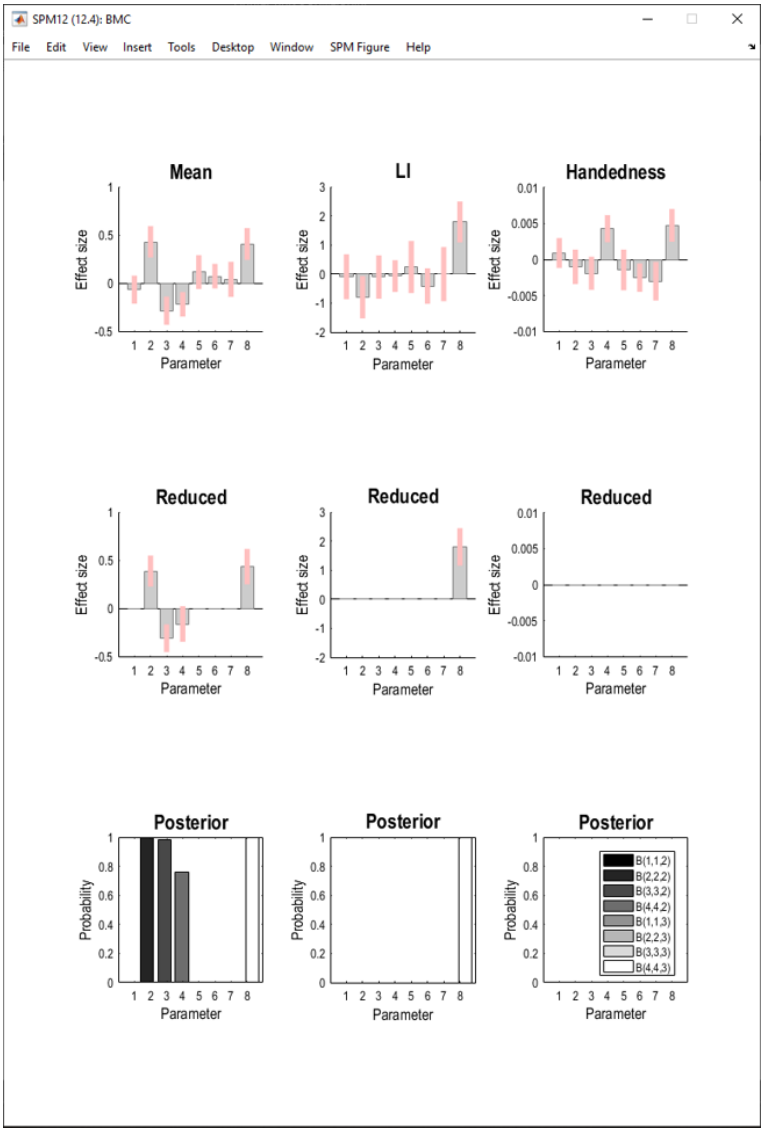

Figure 12: Bayesian Model Average (BMA) after calling spm_dcm_peb_bmc in automatic search mode.

An interactive tool provides an easier way to explore the results of this analysis. This is the third window, titled 'PEB - Review Parameters', shown in Figure 13. Should you need to open this tool yourself at a later stage, then the command is:

Listing 2: PEB review tool

```
spm_dcm_peb_review(BMA,GCM);
```

We'll first walk through each part of this tool, labelled in Figure 13, and then we'll use it to address our experimental question:

A) The boxes give the number of regressors (covariates) in the between-subjects design matrix, the number of DCM parameters taken from the first to the second level and the number of subjects. The between-subjects design matrix is shown below, with one subject per row and one covariate (regressor) per column.

B) The estimated between-subject covariance matrix. The diagonal is the estimated between-subjects variance for each parameter, where the more white they are, the greater the between-subjects variability. Clicking on each item identifies the corresponding parameter.

C) The parameters are grouped by covariate. Use this selector to choose which covariate you'd like to review.

D) Optionally, the parameters can be thresholded to just focus on the most probable effects. The first drop-down menu switches between thresholding based on the free energy (model comparisons with/without each parameter) and thresholding based on the posterior variance (the pink error bars). Where possible, we recommend selecting free energy. The menu on the right is used to select the threshold.

E) The bars are the parameters relating to the selected covariate. Pink error bars are 90% credible intervals. Clicking on a bar shows the name of the parameter, its expected value, and its probability calculated using the option selected above.

F) Use this selector to display the parameters as a connectivity matrix.

We'll now use this screen to see where Laterality Index (LI) had an effect:

- Click 'Please select...' and choose 'Second-level effect - LI' (using part B of the interface). This will display the LI parameters.
- In part D of the interface, next to 'Threshold (optional):' ensure 'Free energy (with vs without)' is selected. In the right-hand box, select to 'Strong evidence (Pp > 95%)' in part D of the interface.
- Click on the single surviving bar, which identifies this parameter as the 'B-matrix from rDF to rDF (Input Words)'. This means it is the modulatory input on rDF's self-connection due to word stimuli.

This result shows that the effect of word stimuli on rDF inhibitory self-connection increased with LI. The strength of the self-connection increased by 1.80 times the LI score. (NB if the bar had been negative, it would have meant a decrease in the inhibitory self-connection with LI). We may therefore conclude that recurrent activity in region rdF is sufficient to explain individual differences in LI across subjects.

This concludes an analysis based on automatically searching over reduced PEB models. Next, we will try comparing specific PEB models based on pre-defined hypotheses.

3.2.3 Model comparison: specific models

We are going to compare the evidence for two specific hypotheses:

- H1: there was modulation of region ldF by Pictures and Words
- H2: there was NO modulation of region ldF by Pictures and Words

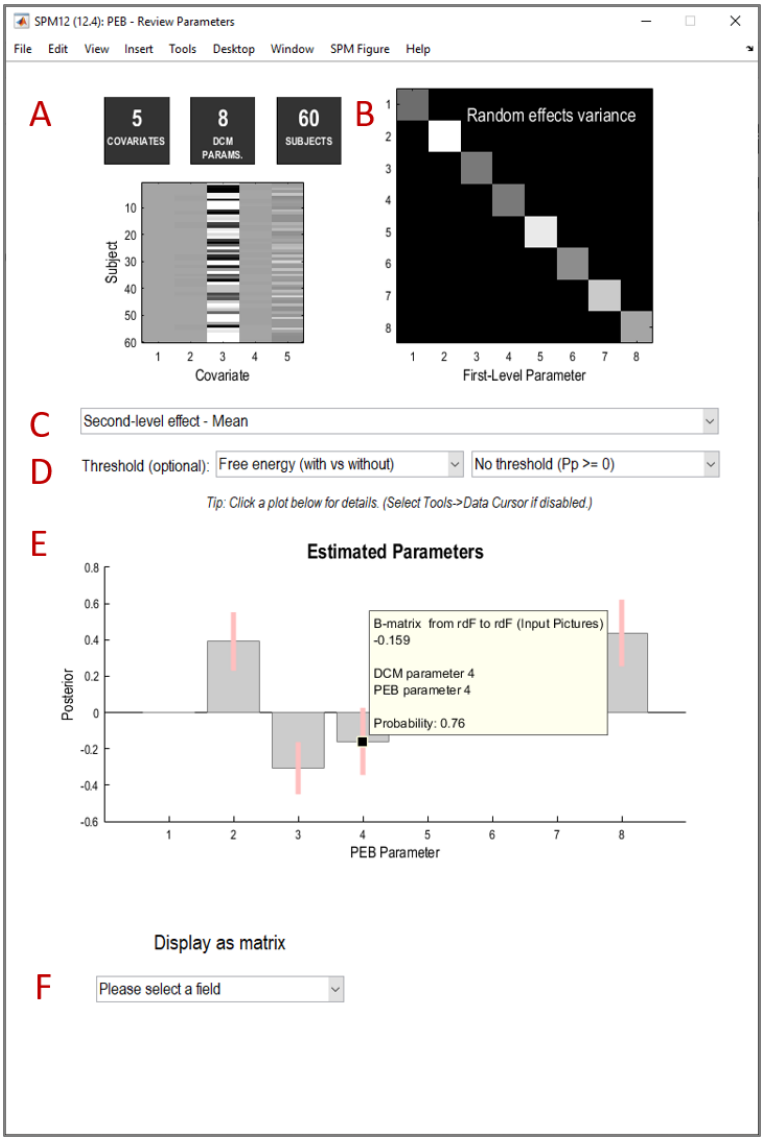

Figure 13: The PEB review tool (spm_dcm_peb_review) applied to the results of the automatic search.

We will test these two hypotheses as explanations for the commonalities across subjects (i.e. the group mean), and the differences across subjects due to LI. This will require comparing the evidence for four alternative PEB models, with parameters switched on or off according to each combination of hypotheses, listed in the table below:

To perform this analysis, follow these steps:

- Open the batch editor, click File → New Batch. Then select SPM → DCM → Second level → Compare / Average PEB models. Fill it out as follows:
  - Select PEB file - Double click and choose the file PEB_B.mat we created earlier.
  - DCMs - Double click and choose the file GCM_two_models.mat we created earlier.
- Save the batch, then click the green play button.

The software will read in the GCM file and look at which connections are switched on or off in each DCM of an example subject. It will use this to determine which parameters should be switched on and off for each candidate PEB model. Two windows are generated. The window

Model	Commonalities	Differences (LI)	Description
1	H1	H1	Common effects (group mean $\neq$ 0) and LI differences
2	H1	H2	Common effects (group mean $\neq$ 0) without LI differences
3	H2	H1	No common effects (group mean=0) but LI differences
4	H2	H2	No common effects (group mean=0) nor LI differences

titled BMC shows the results of the model comparison, shown in Figure 14. The panels in this window are as follows:

- Top left - the model space. This shows which connections were switched on (white) and switched off (black) in each of the two models. The first column is the full model (H1), and the second column is the reduced model (H2) with the effect of Pictures (parameter B(2,2,2)) and Words (parameter B(2,2,3)) on region ldF switched off.
- Top right - the posterior probability for each of the four PEB models described above. The best PEB model had the commonalities across subjects set according to H1 (common effects on ldF) and the LI differences set according to H2 (no LI effects on ldF).
- Middle left - The same result as the top right of the figure, but summed over columns and re-normalized. This is the posterior probability of each hypothesis as an explanation for the commonalities across subjects.
- Middle right - The same result as the top right, but summed over rows and re-normalized. This is the posterior probability of each hypothesis as an explanation for the LI differences across subjects.
- Bottom left - Probability for each of the parameters which varied across models. Computed by comparing the evidence for all models which had each parameter switched on vs all models which had the same parameter switched off.
- Bottom right - Probability for each of the LI parameters that varied across models. Computed by comparing the evidence for all models which had each parameter switched on vs all models which had the same parameter switched off.

We can conclude from this result with 98% probability that there was an effect of Picture and Word stimuli on region ldF across subjects (middle left of the figure), however there was no effect of LI on this connection (with 93% probability). Therefore, hypothesis H1 was the best explanation for the commonalities across subjects and H2 was the best explanation for the LI differences.

The window titled 'PEB - Review Parameters' can be used to review the connectivity parameters averaged across candidate PEB models (the Bayesian Model Average). Select 'Second-level effect - Mean' and for clarity, threshold to only show parameters with strong evidence (95% probability) based on the free energy. **Only parameters which varied across models will be displayed. Any parameters which did not vary across models will have probability set to not a number (NaN) and so will be hidden by the thresholding step.** This plot shows the two effects - pictures and words - on region ldF. It is clear that pictures increased the strength of the inhibitory self-connection on region ldF. Changing to the effect of LI (using the drop-down menu above) shows that no connections in this model comparison expressed an effect of LI.

3.2.4 Prediction: cross-validation

The model search we conducted in Section 3.2.2 identified that the modulation of region rdF's self-connection by word stimuli depended on the subject's LI score. We will now ask: was the effect size large enough to predict a left-out subject's LI score from their connectivity? In other

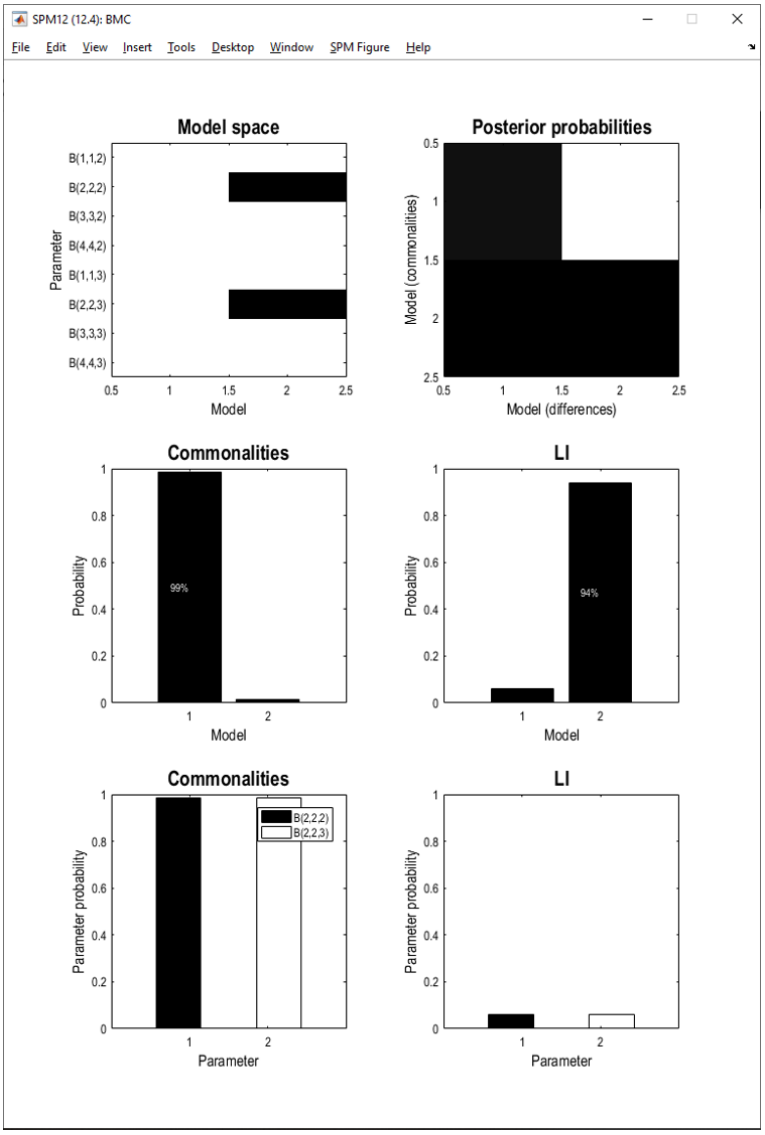

Figure 14: Comparison of two PEB models: a full model and a reduced model with modulations by word stimuli switched off.

words, does this parameter have predictive validity? To assess this, we'll use Leave One Out (LOO) cross-validation.

- Ensure the design matrix is still loaded into the Matlab workspace. You can load this again by double clicking on the file named design_matrix.mat on the left hand side. The design matrix is a variable called X and the names of the columns are in a cell array called labels.
- Open the batch editor by clicking Batch from the main SPM window (or click File, New Batch if its already open). From the menu at the top, select SPM → DCM → Second level → Predict (cross-validation). The batch now asks all the same questions as when you specified the PEB model above. Fill it out as follows:
  - Name - type rdF and click OK.
  - DCMs - select the file named GCM_two_models_pre_estimated.mat from the analyses folder we created earlier.
  - Design matrix - double click and type: X. Then press enter.

  - Covariates - Click on 'Covariate names', click 'New: Name', double click 'Name', then type 'Mean' then click OK. Repeat this to give names for the remaining four covariates: LI, Handedness, Gender, Age.
  - Fields - We will only include the effect of words on the self-connection of region rdF. Click Fields, then click 'Enter manually'. Double click 'Enter manually' in the main box and type: {'B(4,4,3)'} including the curly brackets, then press OK. (The three numbers are: target region, source region and experimental condition. The target and source regions are region number 4 (rdF), and Words is condition number 3.)
  - Max iterations - Scroll to the bottom and set this to 256.

- Save the batch and click the green play button to run it.

This will take a little time to run, as it will loop over all 60 subjects, creating a PEB model on the remaining 59 and trying to predict the LI of the left-out subject. The resulting window is shown in Figure 15. The top left plot shows the estimated (mean-centred) LI score as red lines with 90% confidence interval as the shaded area. The actual scores are shown as faint dotted lines. The top right plot gives the out-of-samples Pearson's correlation between the predicted (expected value) LI score and the actual LI score. This demonstrates a significant positive correlation. The design of the bottom plot depends on whether the effect to be predicted is continuous or discrete. Because LI is continuous, the plot shows the predicted LI scores as circles, and the probability density over the prediction as very faint shaded areas. In this example, this is rather hard to see because the shaded areas are small enough to be hidden by the circles.

We may conclude from this analysis that the effect of words on rdF was sufficiently large to significantly predict subjects' LI scores.

3.3 Summary

This concludes the tutorial on DCM and PEB using the GUI. To recap the steps we took:

1. We specified two DCMs for an example subject: a full model and a reduced model.
2. We replicated these DCMs over subjects.
3. We estimated the DCMs, which provided estimates of the connectivity parameters.
4. We took the estimated modulatory (B) parameters to the group level by specifying a PEB model.
5. We compared the full PEB model to hundreds of reduced PEB models using an automatic search, which revealed a specific effect of LI on region rdF.
6. We also tested a specific hypothesis about region ldF by comparing the full PEB model against a specific reduced model.
7. Finally, we evaluated the predictive validity of a parameter (the influence of words on region rdF) using Leave-One-Out cross-validation.

The remainder of this document repeats and extends this analysis using scripting rather than the GUI.

4 Analysis using scripting

DCM analyses can also be scripted in Matlab, rather than just using the SPM GUI. This is useful for generating multiple model structures per subject without having to click lots of buttons in the GUI, enabling the creation of more interesting model spaces. The pipeline we will implement here using scripts will be similar to that performed above using the GUI, but we will additionally demonstrate having a much larger model space (112 pre-defined models) and the use of 'family analysis' to compare groups of models. The scripts we will work through can be found in the 'code' folder in the accompanying dataset.

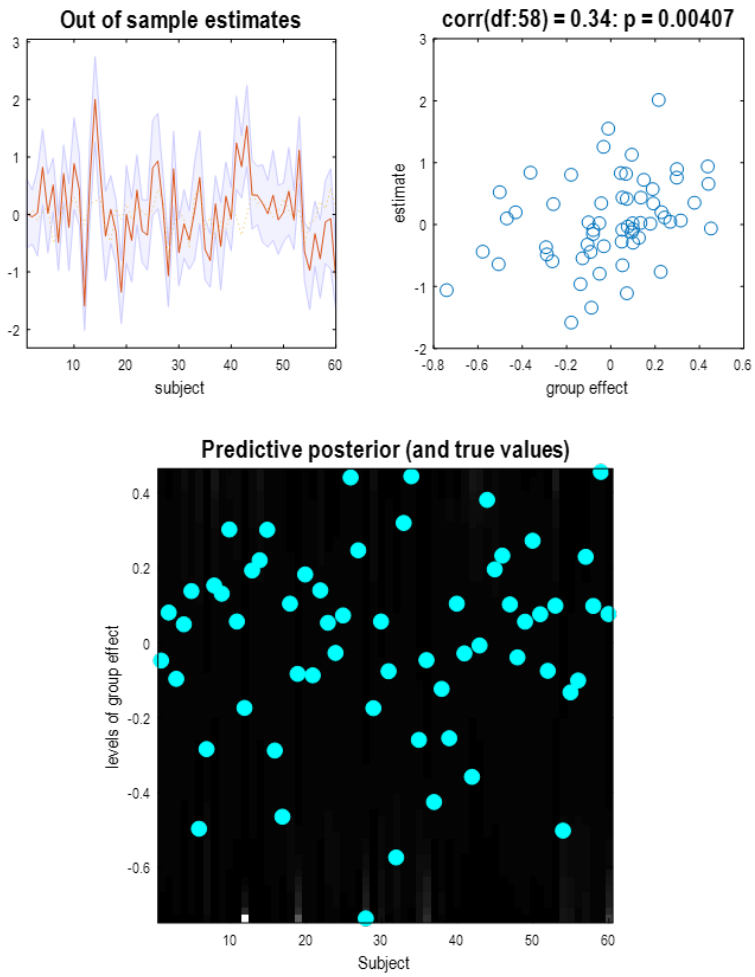

Figure 15: Result of Leave-One-Out (LOO) cross-validation on the self-connection of region rdF to predict the subjects' LI scores.

4.1 First level analysis: DCM for fMRI

We'll start with the within subject (first level) analysis, which can be found in 'code Run_first_level.m'. The script begins with some settings:

Listing 3: Settings

```
% MRI scanner settings
TR = 3.6;    % Repetition time (secs)
TE = 0.05;   % Echo time (secs)

% Experiment settings
nsubjects   = 60;
nregions    = 4;
nconditions = 3;

% Index of each condition in the DCM
TASK=1; PICTURES=2; WORDS=3;

% Index of each region in the DCM
lvF=1; ldF=2; rvF=3; rdF=4;
```

By defining constants like PICTURES=2, the code which follows will be more readable.

4.1.1 Selecting connections

Next we'll configure the connectivity by selecting which connections we want switched on (informed by the data) and which we want switched off (fixed at zero). Matrix A in the DCM neural model is the average connectivity across experimental conditions. We'll form a matrix a which we will use to select which A-matrix connections should be switched on (1) and off (0):

$$a = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

The columns are outgoing connections and the rows are incoming connections, in the order: lvF, ldF, rvF and rdF. For example, the zero in the bottom left corner of the matrix switches off the connection from lvF (region 1) to rdF (region 4).

Matrix B specifies which connections are modulated by each of the experimental conditions. We'll form a matrix b to select which of these parameters should be switched on (1) and off (0):

$$b(:,:,1) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} b(:,:,2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} b(:,:,3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Again, the columns are the outgoing connections and the rows are the incoming connections. The third dimension - shown here as three separate matrices - indexes the experimental condition, ordered: Task=1, Pictures=2 and Words=3. So in this example, the first b-matrix says that Task does not modulate any connections, whereas Pictures and Words modulate the self-connection of all regions (in the second and third matrices respectively).

Finally, matrix c determines which regions are driven by each experimental condition:

$$c = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

The columns are the conditions and the rows are the brain regions. This says that all four regions can be driven by Task, whereas Pictures and Words are only used used as modulatory inputs.

4.1.2 DCM specification

Let's write the connectivity matrices above in Matlab code:

Listing 4: Connectivity matrices

```
% A-matrix (on / off)
a = ones(nregions,nregions);
a(lvF,rdF) = 0;
a(rdF,lvF) = 0;
a(ldF,rvF) = 0;
a(rvF,ldF) = 0;

% B-matrix
b(:,:,TASK)     = zeros(nregions); % Task
b(:,:,PICTURES) = eye(nregions);   % Pictures
b(:,:,WORDS)    = eye(nregions);   % Words

% C-matrix
c = zeros(nregions,nconditions);
```

```
c(:,TASK) = 1;

% D-matrix (disabled but must be specified)
d = zeros(nregions,nregions,0);
```

We can now specify the DCMs. For each subject we'll need their SPM.mat file (which provides the experimental timing) and their Volume of Interest (VOI) files, which provide the timeseries. We'll give these to the spm_dcm_specify.m function to create each subject's DCM:

Listing 5: DCM specification

```
start_dir = pwd;
for subject = 1:nsubjects

    name = sprintf('sub-%02d',subject);

    % Load SPM
    glm_dir = fullfile('..','GLM',name);
    SPM     = load(fullfile(glm_dir,'SPM.mat'));
    SPM     = SPM.SPM;

    % Load ROIs
    f = {fullfile(glm_dir,'VOI_lvF_1.mat');
         fullfile(glm_dir,'VOI_ldF_1.mat');
         fullfile(glm_dir,'VOI_rvF_1.mat');
         fullfile(glm_dir,'VOI_rdF_1.mat')};
    for r = 1:length(f)
        XY = load(f{r});
        xY(r) = XY.xY;
    end

    % Move to output directory
    cd(glm_dir);

    % Select whether to include each condition from the SPM.mat
    % (Task, Pictures, Words)
    include = [1 1 1]';

    % Specify the DCM
    s = struct();
    s.name       = 'full';
    s.u          = include;
    s.delays     = repmat(TR,1,nregions);
    s.TE         = TE;
    s.nonlinear  = false;
    s.two_state  = false;
    s.stochastic = false;
    s.centre     = true;
    s.induced    = 0;
    s.a          = a;
    s.b          = b;
    s.c          = c;
    s.d          = d;
    DCM = spm_dcm_specify(SPM,xY,s);

    % Return to script directory
    cd(start_dir);
end
```

4.1.3 Estimating a GCM file

We have now created a DCM for each subject named DCM_full.mat. Next we'll collate these into single group DCM file (GCM), which is a cell array with one row per subject and one column per model, which we'll then fit to the data:

Listing 6: Estimating models

```
% Find all DCM files
dcms=spm_select('FPListRec','../GLM','DCM_full.mat');

% Character array -> cell array
GCM = cellstr(dcms);

% DCM filenames -> DCM structures
GCM = spm_dcm_load(GCM);

% Estimate DCMs (this won't effect original DCM files)
GCM = spm_dcm_fit(GCM);

% Save estimated GCM
save('../analyses/GCM_full.mat','GCM');
```

In the code above, we used the spm_select function to search for all subjects' DCMs (within the GLM directory, located one directory up) and list their filenames. It returned a character array, which the second line of code converted to a cell array, which is easier to work with. The third line loaded the DCMs into memory and the fourth line fitted each subject's DCM to their data. This doesn't update the original DCM files, and in the last line we saved the estimated GCM.

Tip: you can speed up DCM estimation by using parallel computing. As of SPM revision 7365, you can switch this on by replacing the penultimate line in the previous listing with the following:

Listing 7: Enabling parallel DCM estimation

```
use_parfor = true;
GCM = spm_dcm_fit(GCM,use_parfor);
```

4.1.4 Diagnostics

Having completed the estimation of the first-level DCMs, it is a good time to perform some diagnostics on the models. To inspect the models, run:

Listing 8: DCM for fMRI diagnostics

```
spm_dcm_fmri_check(GCM);
```

For interpretation of these results, please see section 3.1.5.

4.1.5 Specifying reduced DCMs

Following the paper, we will test whether the commonalities across subjects, and the differences across subjects due to LI, could be explained by:

1. picture stimuli and / or word stimuli,
2. dorsal and / or ventral regions,
3. left and / or right regions.

To try all combinations of these options, we will need 3x3x3=27 PEB models with different sets of parameters switched on or off. Additionally, we'll include a 'null' model, with no modulatory inputs by task. That gives a total of 28 alternative models to compare, based on our hypotheses. These candidate DCMs will not need to be estimated - we will just use them to tell the software which connections to switch on and off for each candidate PEB model. The script supplied with this tutorial creates 28 minimal DCMs containing only the required fields: a,b,c,d,name and options, and it puts these DCMs in a cell array with a single row (saved in analyses/GCM_templates.mat).

To recap, we have now specified DCMs for each subject, collated them into a GCM file, fitted them to each subject's data and performed some basic diagnostics. We have also prepared templates models defining our model space. Next, we will use these models to test hypotheses at the group level using PEB.

4.2 Second level analysis: PEB

4.2.1 PEB model specification

There are two prerequisites for a PEB analysis: a between-subject design matrix and the estimated DCMs from each subject (collated in a GCM file). Let's load the ready-made design matrix for this data as well as the GCM we made earlier. We'll create a structure called M to specify the settings for the PEB model:

Listing 9: PEB specification

```
% Load design matrix
dm = load('../design_matrix.mat');
X        = dm.X;
X_labels = dm.labels;

% Load GCM
GCM=load('../analyses/GCM_full.mat');
GCM=GCM.GCM;

% PEB settings
M = struct();
M.Q      = 'all';
M.X      = X;
M.Xnames = X_labels;
```

M.Q is the choice of precision components to use. By setting this to 'all', the between-subject variability for each DCM connection will be individually estimated. M.X is the design matrix and M.Xnames is a cell array containing the names of the regressors (columns) in the design matrix. These are: Mean, LI, Handedness, Gender, Age. For a full list of the available PEB settings, see the help text in the function spm_dcm_peb.m.

4.2.2 PEB model estimation

The PEB model is estimated as follows:

Listing 10: PEB estimation

```
[PEB_B,RCM_B] = spm_dcm_peb(GCM,M,{'B'});
save('../analyses/PEB_B.mat','PEB_B','RCM_B');
```

The last input to spm_dcm_peb selects which DCM parameters to take to the group level, i.e. to treat as random effects. Here we have selected all parameters in matrix B (the modulatory inputs). This will be a total of eight parameters per subject: the modulatory effect of Pictures and the modulatory effect of Words on each of the four regions. The first output from spm_dcm_peb is the PEB model (PEB_B). The second output (RCM_B) is the array of DCMs, where each subject's model has been finessed by the group-level analysis. More specifically, in

the array RCM_B, the individual subjects' parameters and evidence have been updated using the group-level connection strengths as empirical priors, potentially saving any subjects with noisy data.

4.2.3 Model comparison: automatic search

Having estimated the PEB model, the next step is to test hypotheses. We'll start with the simplest model comparison procedure, which is an automated search over reduced PEB models. This will evaluate many nested models, in which combinations of parameters are switched off. The parameters of the best candidate models will be averaged, returning a Bayesian Model Average (BMA):

Listing 11: Automatic PEB search

```
BMA_B = spm_dcm_peb_bmc(PEB_B);
save('../analyses/BMA_search_B.mat','BMA_B');
```

This produces two windows, the interpretation of which is described in section 3.2.2.

4.2.4 Model comparison: specific models

Instead of performing an automatic search over reduced models, we can instead compare specific models according to our hypotheses. For this example we'll compare 28 models which will differ in the combination of connectivity parameters switched on and off. Recall that the PEB model includes parameters quantifying the group average strength of each connection (the commonalities across subjects), and parameters quantifying the difference between subjects due to laterality index (LI). The software will evaluate all of the 28 hypotheses for the commonalities parameters, crossed with all of the 28 hypotheses for the LI parameters, necessitating 28x28=784 candidate PEB models. Thanks to Bayesian Model Reduction, these can be evaluated in seconds.

To tell the software which connections should be enabled for each of the 28 options, we'll give it the template DCMs we defined earlier:

Listing 12: Compare specific PEB models

```
% Load estimated PEB
load('../analyses/PEB_B.mat');

% Load template models
templates = load('../analyses/GCM_templates.mat');

% Run model comparison
[BMA,BMR] = spm_dcm_peb_bmc(PEB_B, templates.GCM);
```

Here we have used the same function as for the automatic search (spm_dcm_peb_bmc) but with a second input - a cell array of template DCMs, organised into a cell array of dimension 1 x 28. This produces a figure window titled BMC, shown in Figure 16. The top left shows the model space - the connections switched on (white) and off (black) in each model. It's a good idea to use this part of the figure to check that the models were correctly specified. The top right shows the joint probability for the 28 models as explanations for the commonalities across subjects (i.e. the group average connectivity) and the differences across subjects encoded in the second column of the design matrix (LI). This shows that one combination of models has done particularly well - model four of the commonalities and model fifteen of the LI differences. The second row shows the same result, but summed over the columns (left) and over the rows (right) and re-normalized, to give the probability of each model as explanations for the commonalities and LI. The bottom row shows the result of model comparisons with and without each parameter. Only one parameter has strong evidence as an explanation for LI - B(4,4,3), the modulatory input on rdF's self-connection due to word stimuli, reproducing our finding above using the automatic search.

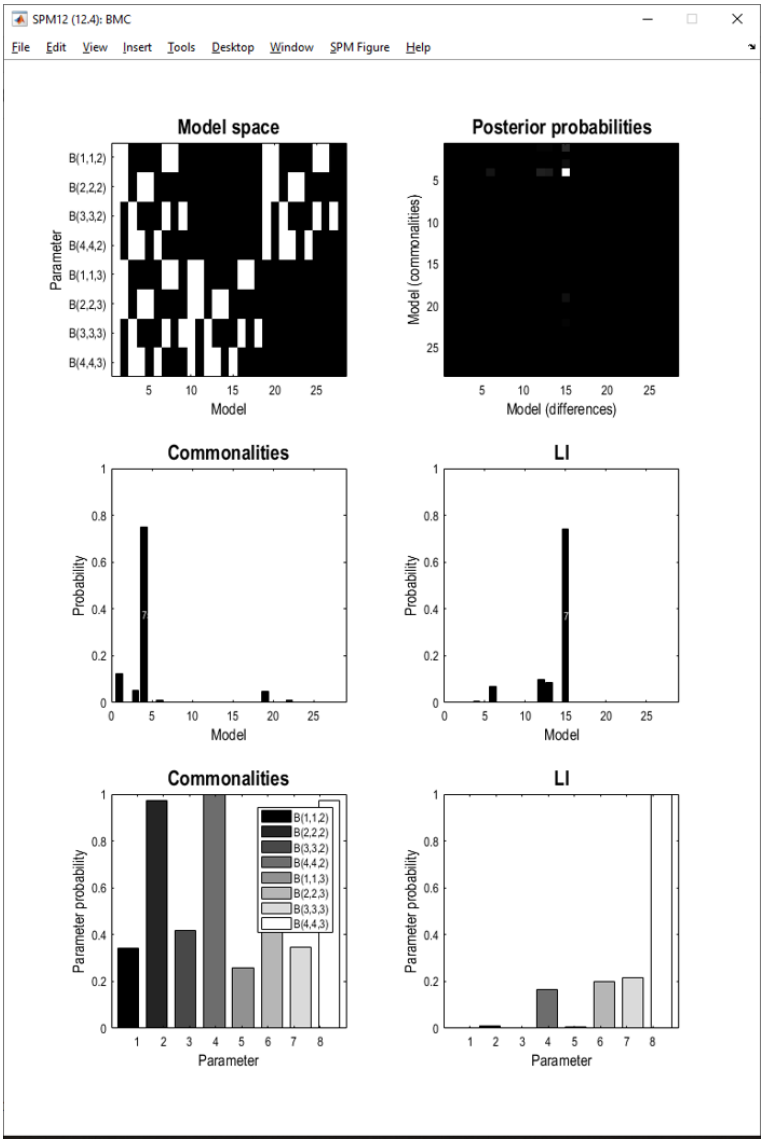

Figure 16: Output from spm_dcm_peb_bmc when comparing specific pre-defined models.

It is also clear from this result that several models had non-trivial levels of evidence, and so the best way to summarise the connection strengths is to look at the Bayesian Model Average. As for the automatic search, we can review the BMA using the review tool:

Listing 13: PEB review tool

```
% Review the BMA
spm_dcm_peb_review(BMA,GCM);
```

Which gives similar results as described for the automatic search.

4.2.5 Model comparison: families

As described in the accompanying paper, the 28 models were specified in order to address a series of specific questions. To address these, we can group the models into families and compare the evidence for different families. For example, to ask which experimental condition was the best explanation for the commonalities and differences across subjects, we'll split the models into four families:

1. Both words and pictures modulating
2. Words modulating
3. Pictures modulating
4. No modulation

To define which of the 28 models should be in which family, we will create a vector called task_family where element j is the family for the j'th model. For example, the tenth element of this vector - task_family(10) - equals 2, meaning that the 10th model is in family number 2. We can then run the family analysis:

Listing 14: Family comparison

```
% Load the result from the comparison of 28 reduced models
load('../analyses/BMA_BC_112models.mat');

% Group models into families and compare
[BMA_fam_task,fam_task] = spm_dcm_peb_bmc_fam(BMA, BMR, templates.
    task_family, 'ALL');
```

This function (spm_dcm_peb_bmc_fam) computes probabilities for each family of models, as well as an updated BMA under the prior that each family is equally likely. The outputs of the model comparison in the previous section (BMA,BMR) form the inputs to spm_dcm_peb_bmc_fam, as well as the definition of which model should be assigned to which family (templates.task_family). The final input 'ALL' selects how to generate the BMA. The options are 'ALL' (compute the BMA over all models), 'WINNING' (compute the BMA over models in the winning family) or 'NONE' (don't average).

The results are shown in Figure 17. The top left shows the posterior probability for each family as explanations for the commonalities across subjects (the four rows) and the differences among subjects due to LI (the four columns). This shows that the best combination, with 86% probability, is Family 1 for the commonalities (both words and pictures) and Family 2 for the LI differences (words only). Therefore, we may conclude that this brain network was modulated by both words and pictures, but only the modulation by words is needed to explain LI differences. The top right plot shows the updated posterior probabilities over models given the family definitions, and this is summed over rows and columns in the middle plots. The bottom left plot shows the assignment of each model to each family (where the first family is black and the last is white). The bottom right plot shows the models which contributed to the BMA, which as above, can be viewed using the review tool (spm_dcm_peb_review).

4.2.6 Prediction: Cross-validation

The consistent result across these analyses is that the modulation of self-inhibition in region rdF by word stimuli was correlated with subjects' LI score. Our final question is whether the effect size was large enough to predict a left-out subject's LI, based on their connectivity. To test this we will build a PEB model on all subjects bar one, try to predict the left-out subject's LI, and then repeat with the next subject. This is done as follows:

Listing 15: Cross-validation

```
[qE,qC,Q] = spm_dcm_loo(GCM,M,{'B(4,4,3)'});
save('../analyses/LOO_rdF_words.mat','qE','qC','Q');
```

The spm_dcm_loo (leave-one-out) function takes as input the GCM array and the PEB settings M we defined earlier. The final input is a cell array of the DCM connections we want to use for prediction. It's a good idea to only include parameters which show strong experimental effects - here we include only the modulation of region 4 (rdF) by condition 3 (words). This produces the plot shown in Figure 15, described in detail in Section 3.2.4. We conclude from this result that the effect size was large enough for statistically significant prediction of the LI score. The predicted posteriors are returned by the function in the parameters qE and qC.

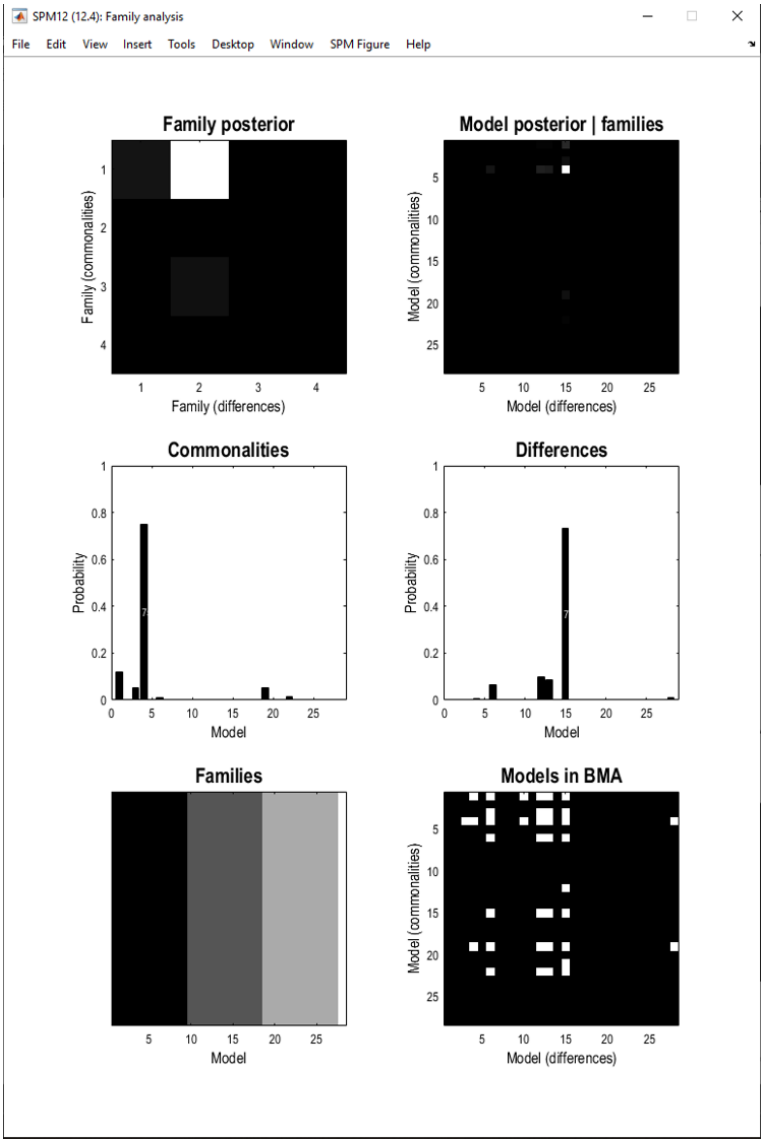

Figure 17: Output from spm_dcm_peb_bmc_fam when comparing families of models.

4.3 Summary

We have shown how to test hypotheses about connectivity at the group level using PEB. With this example dataset, we found that individual differences in Laterality Index (LI) can be explained by the level of response of region rdF to word stimuli. The key software functions for group-level analysis were as follows:

- spm_dcm_peb - Estimates a PEB model.
- spm_dcm_peb_bmc - Performs Bayesian model comparison and averaging (BMA) over reduced PEB models, where combinations of connectivity parameters have been switched off at the group level.
- spm_dcm_peb_review - Interactive GUI for reviewing the parameters of a PEB or BMA from any of the functions above.
- spm_dcm_loo - Performs leave-one-out cross-validation, to evaluate whether effect sizes were large enough to predict left-out subjects' parameters.

5 Troubleshooting

5.1 Dealing with very low explained variance in DCM for fMRI

How much explained variance is enough to be useful has no clear answer. As a rule of thumb, if you're getting less than 10 percent explained variance in some or many subjects, you may wish to investigate this further. You have a few options:

1. Do nothing. We usually draw conclusions at the group level, and it is quite typical that some subjects don't show experimental effects. By including these subjects in the group-level analysis, the between-subject variability is properly represented in the analysis.

2. Revisit timeseries extraction. It may be that some subjects had their experimental effects located in different brain areas than expected - for example, because they were using different cognitive strategies, or due to individual differences in anatomy. This can be checked by viewing the results of their first level SPM analysis, to see if the coordinates selected for timeseries extraction overlapped with experimental effects.

3. Revisit DCM model specification. It may be that a better model of the data would do better for these subjects. One strategy is to simplify the model - include the minimum number of brain regions and connections to see if you can get a model to fit the data.

4. Re-estimate with empirical priors. Getting the parameters into the right range before estimation can help to rescue troublesome subjects. You can try re-starting model estimation, using the estimated group-average connection strengths as priors. Replacing spm_dcm_fit with spm_dcm_peb_fit will do this for you - but note that this can take a long time (every subject's DCM will be estimated up to 4 times).