CPCZurich2021 Tutorials – DYNAMIC CAUSAL MODELING

Practical Instructions

Authors:

Jakob Heinzle (heinzle@biomed.ee.ethz.ch)

Birte Toussaint (toussaint@biomed.ee.ethz.ch)

Revision and testing:

Ines Pereira

There are two sections to this practical: the first sets up and runs a single-subject DCM analysis; the second performs a group-level analysis.

There are several places where you can find more information about SPM (Statistical Parametric Mapping) and DCMs:

- https://www.fil.ion.ucl.ac.uk/spm/doc/manual.pdf
- https://www.fil.ion.ucl.ac.uk/spm/course
- https://en.wikibooks.org/wiki/SPM

There are also two excellent new step-by-step guides from Peter Zeidman et al.:

- https://doi.org/10.1016/j.neuroimage.2019.06.031
- https://doi.org/10.1016/j.neuroimage.2019.06.032

Section 1

Here, you will:

- Extract timeseries from a set of VOIs (Volume of Interest)
- Specify DCM via the GUI (Graphical User Interface)
- Inspect results
- Run a simple model comparison

First, we need to setup the environment. If you have not done so already follow the steps described in CPCZurich2021_Tutorials_DCM_InstallationGuide.pdf.

The last command you will run there is

```
cpc_glm_dcm_subject();
```

N.B. While interacting with the SPM GUI, it will frequently open new figures and/or print important information in the MATLAB main window. Be aware that the results of the commands you run can appear in all sorts of different places! Similarly, the GUI may well print error messages from time to time because it loses track of all these various figures. Do not worry unduly about these.

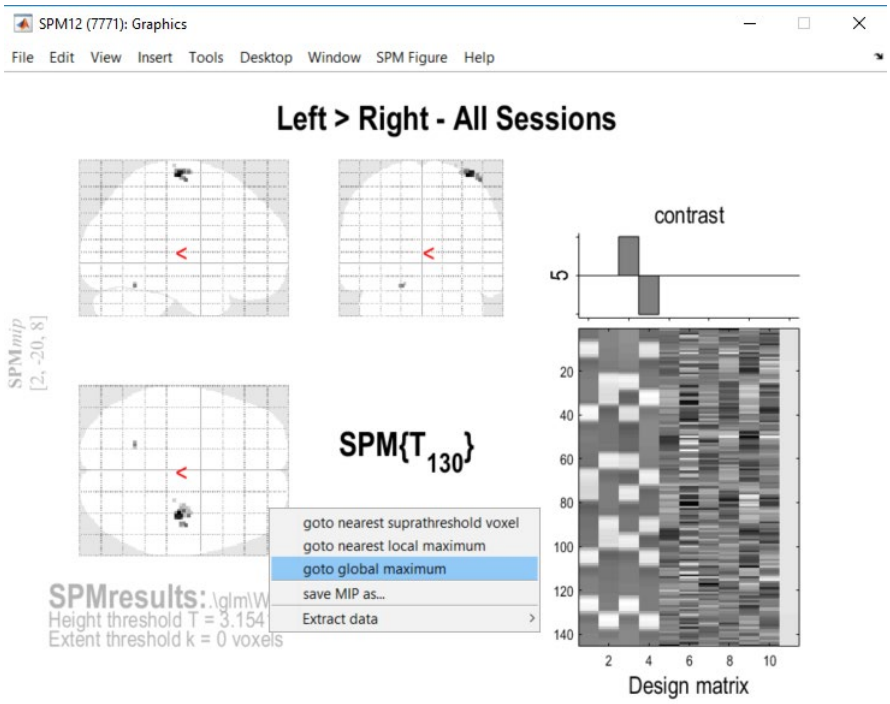
A) Timeseries Extraction

We will show you how to extract timeseries interactively via the Results browser, but then extract them automatically for you via the batch system. As this is not the focus of this tutorial we will not go into too much detail, but plenty of other information is available. For example:

- https://doi.org/10.1016/j.neuroimage.2019.06.031
- https://en.wikibooks.org/wiki/SPM/Timeseries_extraction

The setup command should have run the GLM (General Linear Model) and opened a results window. Have a quick look at the results, and make sure you understand the structure of the task. You can change contrasts under the Contrasts menu in the Results window. To extract a timeseries you can do the following.

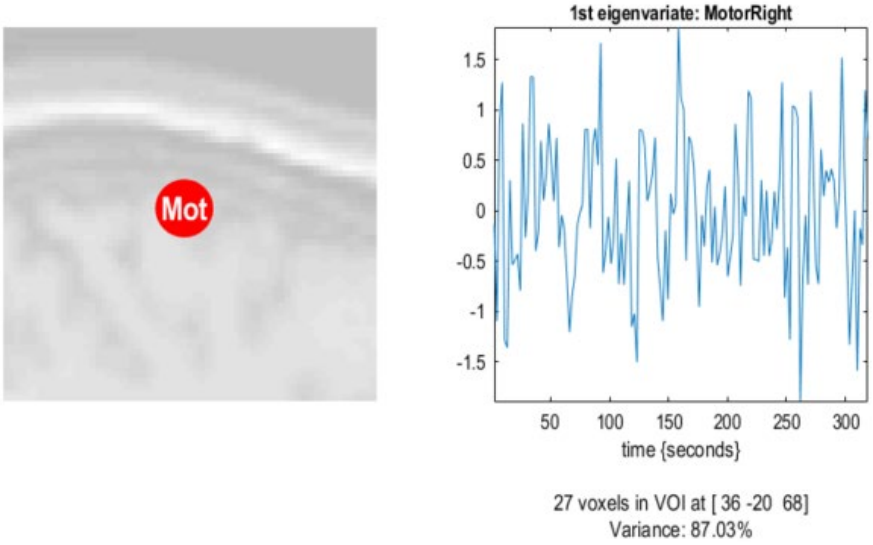
- In the Graphics window, click on top contrast in the statistics table, or right click and select '`goto global maximum`'. The red cursor should move.
-



Statistics: ***p-values adjusted for search volume***

set-level		cluster-level				peak-level					mm mm mm		
p	c	$p_{FWE\text{-}corr}$	$q_{FDR\text{-}corr}$	k_E	p_{uncorr}	$p_{FWE\text{-}corr}$	$q_{FDR\text{-}corr}$	T	(Z_E)	p_{uncorr}			
1.000	5	0.347	0.065	42	0.013	1.000	0.820	3.70	3.60	0.000	36	-20	68
						1.000	0.820	3.41	3.33	0.000	38	-12	70
		1.000	0.698	1	0.698	1.000	0.820	3.45	3.37	0.000	46	-14	64
		1.000	0.698	3	0.474	1.000	0.820	3.39	3.31	0.000	-18	-56	-20
		1.000	0.698	3	0.474	1.000	0.820	3.35	3.27	0.001	44	-18	68
		1.000	0.698	1	0.698	1.000	0.884	3.21	3.14	0.001	26	-24	76

- In the Results window, select '`eigenvariate`'. Fill in the various options (adjust for 'Main Effect – All Sessions).
- Check the output in the Graphics window looks sensible (we have included an example below).



Note that SPM will frequently change your working directory and/or clear your variables after running commands. Annoying huh? Now change back to the `code/` directory and run the following:

```
cpc_voi_dcm_subject();
```

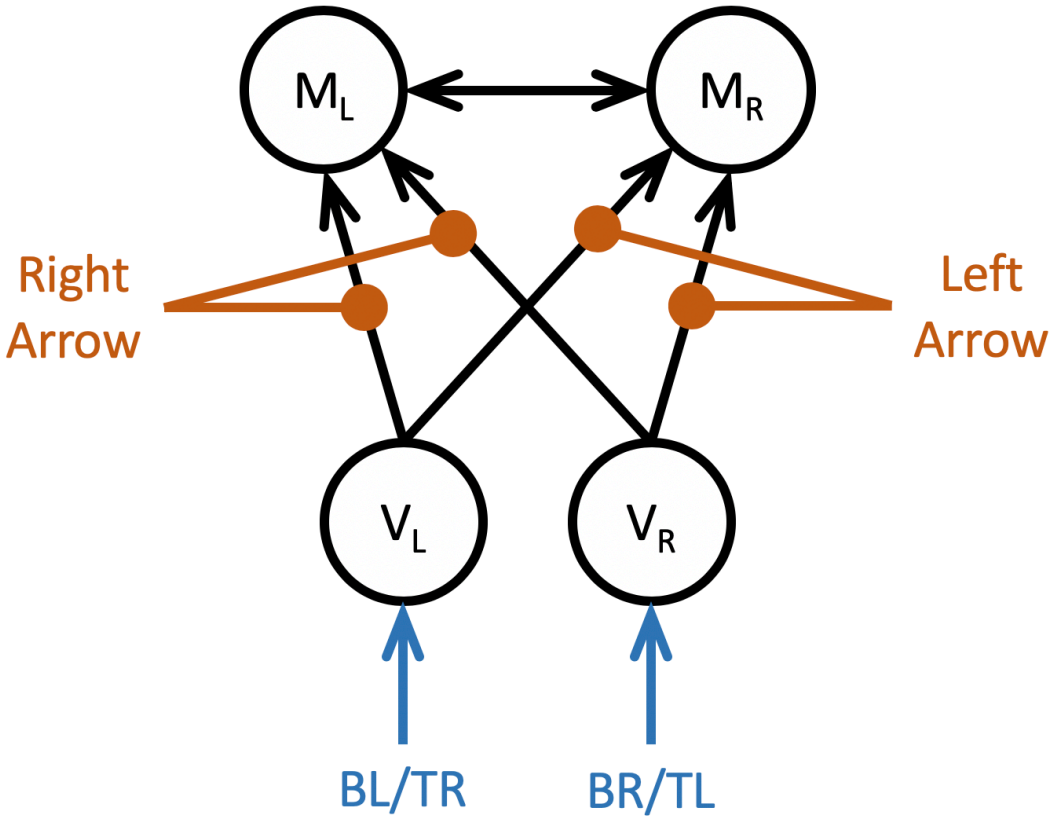
This extracts timeseries from two visual and two motor regions, and we will use these to specify the DCM in the next section.

B) Run a single-subject DCM

Go to (or reopen with `spm fmri`) the main SPM window and select 'Dynamic Causal Modelling'. Select `Specify`, and then load the `SPM.mat` file in the

`data/visuomotor/Sub01/glm/WedgeMotor_noderivs/` directory. Give this a name (e.g. 'number1', SPM will automatically add a prefix so that the full name will be 'DCM_number1'), load the four VOIs (`Vis_L`, `Vis_R`, `Mot_L`, `Mot_R`), and include all the inputs. Timing info can stay as per the defaults. Keep things simple and select a bilinear DCM with one state per region (no stochastic effects). Don't centre the inputs and choose fit timeseries.

Now specify the connectivity structure. If you have entered the VOIs in the same order, this will look like the below. Make sure you understand what each of the windows containing connectivity parameters are specifying!!



Specify endogenous (fixed) connections from

to	1	2	3	4
Vis_L 1	○	○	○	○
Vis_R 2	○	○	○	○
Mot_L 3	◉	◉	○	◉
Mot_R 4	◉	◉	◉	○

Effects of TopLeft-BottomRight on regions... and connect

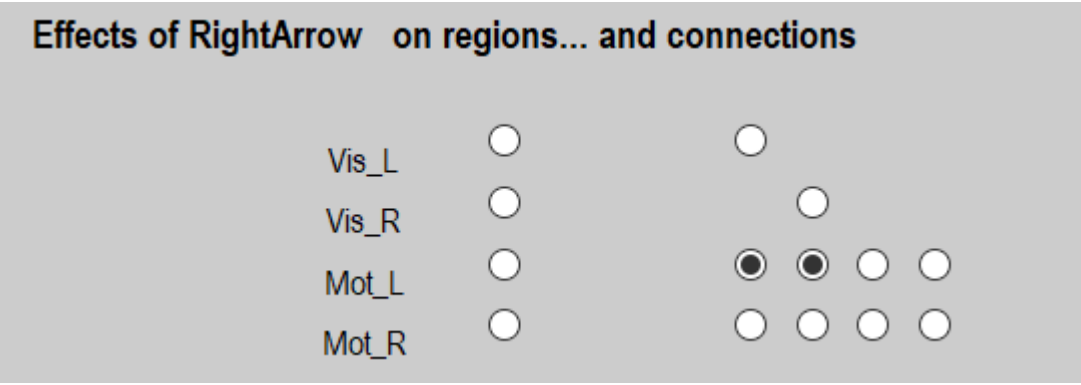
Vis_L	○		○		
Vis_R	◉			○	
Mot_L	○		○	○	○ ○
Mot_R	○		○	○	○ ○

Effects of TopRight-BottomLeft on regions... and connect

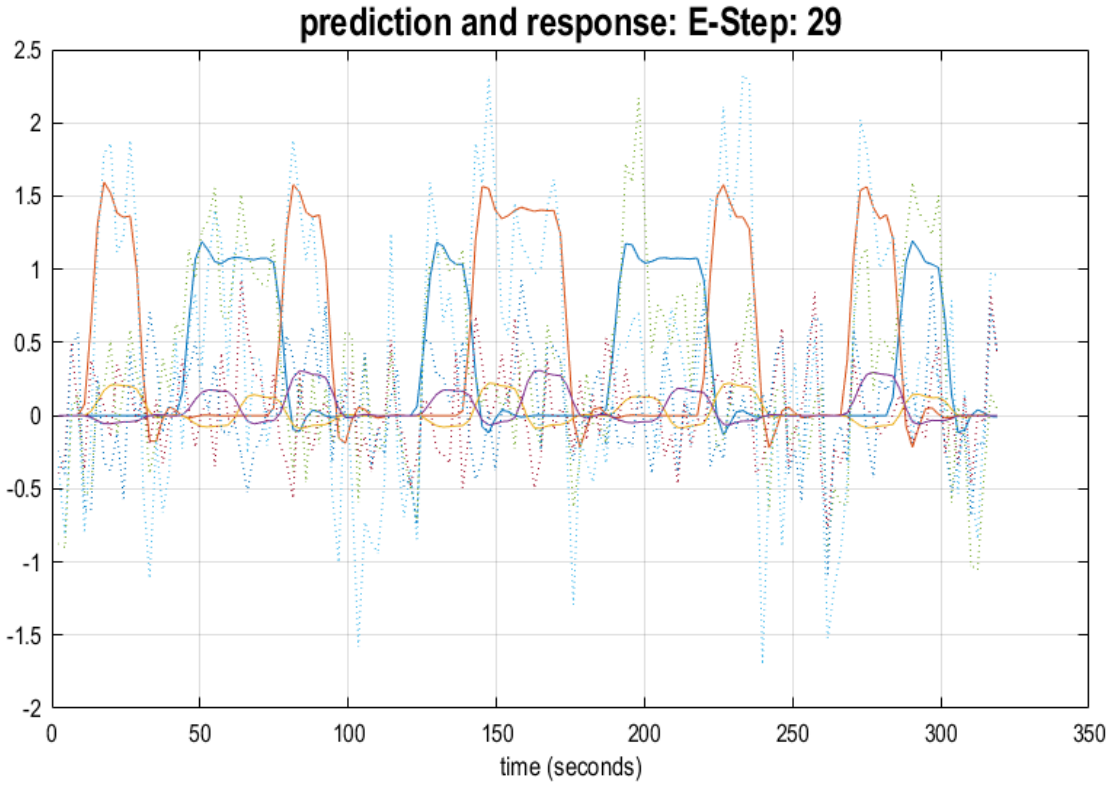
Vis_L	◉		○		
Vis_R	○			○	
Mot_L	○		○	○	○ ○
Mot_R	○		○	○	○ ○

Effects of LeftArrow on regions... and connections

Vis_L	○		○		
Vis_R	○			○	
Mot_L	○		○	○	○ ○
Mot_R	○		◉	◉	○ ○



Once this is done, reselect '`SPM > Dynamic Causal Modelling`' and now select '`estimate (time-series)`'. Choose the DCM you have specified, and the inference should run automatically for a minute or two. This should give something like the following after convergence:



Once this is done, reselect '`SPM > Dynamic Causal Modelling`' and now select '`review`'. Familiarise yourself with the behaviour of this model. You can also interact with the DCM structure directly (you can e.g. try the code below), but the GUI is usually sufficient.

```
load DCM_number1.mat
ylocs = linspace(0.0, 2.0 * (DCM.n - 1), DCM.n);
figure(); hold on
plot(DCM.Y.y + ylocs);
plot((DCM.Y.y - DCM.R) + ylocs, 'k');
yticks(ylocs); yticklabels(DCM.Y.name);
xlim([0, DCM.v + 1]); xlabel('Volume');
```

C) Run a model comparison

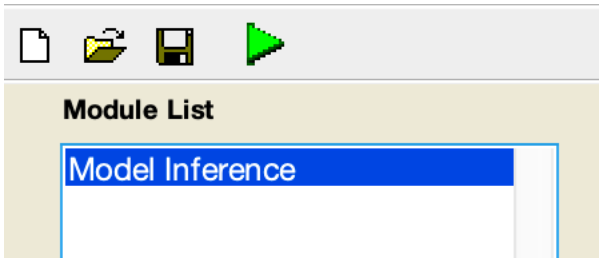
You can now run a different DCM with different modulation. To do so, you can execute the following lines. Make sure you fill in the correct filenames. First, go to your GLM folder, where the first DCM is saved.

```
load DCM_number1.mat % load your first DCM
DCM = rmfield(DCM, 'M'); % remove all model specifications
DCM.b(3,:,4)=[0 0 1 1]; % change which B entries are estimated.
DCM.b(4,:,3)=[0 0 1 1];
save DCM_number2.mat DCM % save with the name of the new DCM.
```

You can now estimate this DCM exactly as you did with the first one. Review this new DCM, and make sure you understand what is different.

Once this is done, reselect '`SPM > Dynamic Causal Modelling`' and now select '`compare`. Surprise! This has now opened the batch editor. You now need to specify `Directory`, `Data`, and `Inference method`. The directory is the GLM directory (which is by now probably the current directory), and you need to add a new subject and session under `Data`. Inference method can be set to fixed-effects for the beginning. Then select your `DCM_number1.mat` and `DCM_number2.mat` under `Models`. Press the green play button in the top left to run the analysis.

Which model wins? How does a fixed-effects analysis compare to a random-effects one?



Section 2

Here, you will:

- Run an analysis that looks for group-level differences in DCM parameters.

As it would take some time to analyse a whole group of subjects we have run a set of analyses for you. These are simulated based on the model structure we used in section 1, and the DCMs can be found in `GCM_prespec1_est.mat`.

This is the format that is needed to run a Parametric Empirical Bayes (PEB) analysis. Unfortunately, we cannot review this structure using the GUI, but it is easy to extract the individual DCMs and inspect them directly. More info about PEB can be found at the links below:

- https://doi.org/10.1016/j.neuroimage.2019.06.032
- https://en.wikibooks.org/wiki/SPM/Parametric_Empirical_Bayes_(PEB)

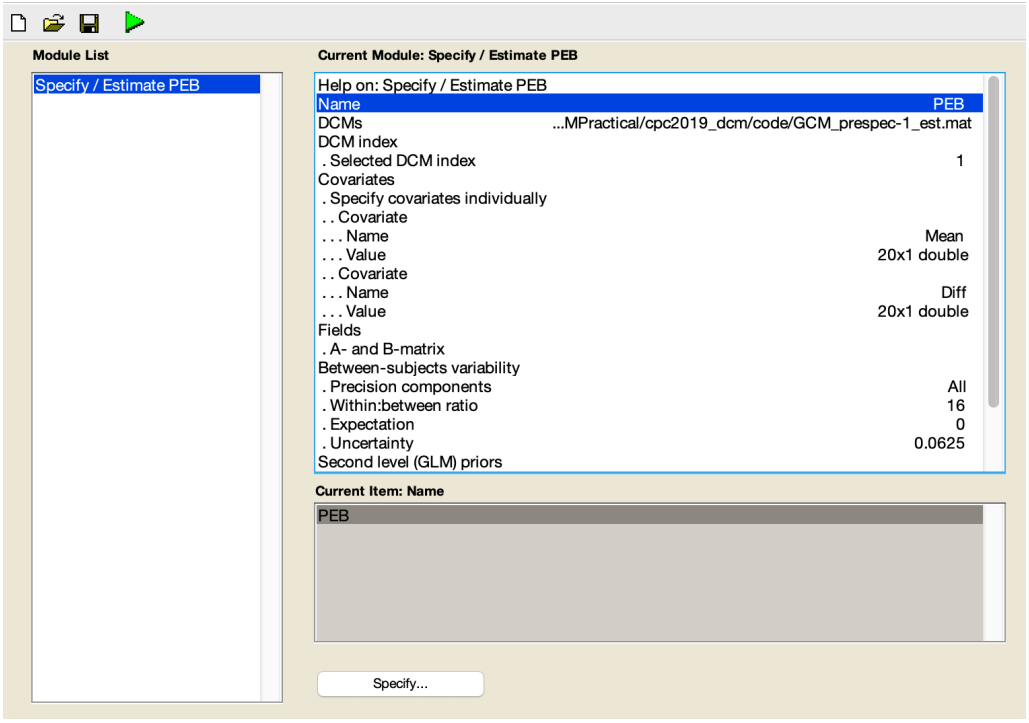
(Re)open SPM, and select '`Batch`' from the menu window (bottom panel). From the new Batch Editor select '`SPM > DCM > Second level > Specify / Estimate PEB`'. Choose a name, and select `GCM_prespec1_est.mat` for the DCMs.

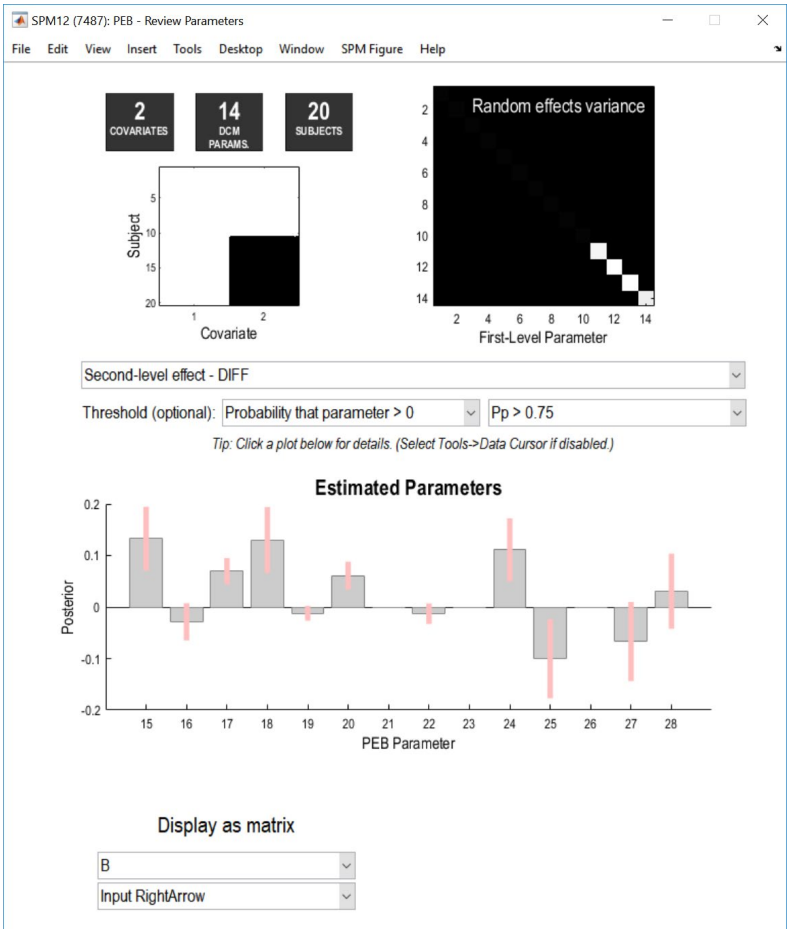
We now need to add our design. There are two groups of ten subjects each, and we want to model the group mean and group difference. Select '`Covariates > Specify covariates individually`' and add two new covariates. The first is the group mean, for which you can simply specify '`ones(20,1)`' under Value. For the difference, you can use '`[ones(10,1); -ones(10,1)]`'. This should have filled all the necessary fields, so press the green arrow to run the analysis!

Now select 'SPM > DCM > Second level > Specify / Review PEB' and add the necessary files. Press the green arrow again, and the PEB GUI should appear.

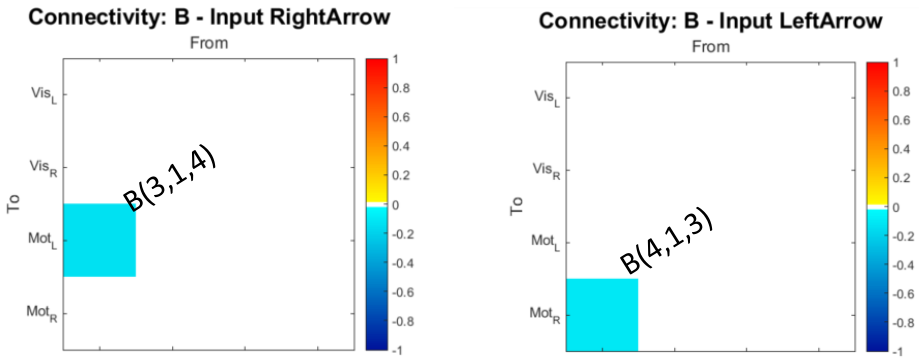
Explore the results. What are the key differences between groups?

Finally, you can see how we simulated the data in `cpc_gen_dcm_group.m`. Does this match what you see in the PEB results? Explore the changes we get when using PEB and what we have changed in the DCMs. What could be explanations for discrepancies, if there are any?





Finally, we would like to look at two specific parameters: the modulation from the left visual cortex (area 1) to the two motor regions (areas 3 and 4). In the DCM world these parameters are `B(3,1,4)` and `B(4,1,3)`. As a control, we also consider `B(3,2,4)`.



We first extract the parameters for all DCMs from the GCM struct. For this, go to the folder with the file `GCM_prespec1_est.mat` and load the file:

```
load('GCM_prespec1_est.mat');
```

The posterior parameter estimates for subject k are stored in `GCM{k}.Ep`. In there, you will find all the parameters of the Model. Have a look at this. Now, let's extract the parameters we are interested in. For this you can loop over subjects and extract the parameters (for didactical reasons, we will also extract `B(3,2,4);`).

```
for k=1:20;
    B314(k) = GCM{k}.Ep.B(3,1,4);
    B413(k) = GCM{k}.Ep.B(4,1,3);
    B324(k) = GCM{k}.Ep.B(3,2,4);
end
```

We can now plot these parameters for the two groups, for example, using the following lines. It's best to open a new figure, before.

```
figure; hold on;
plot(0.9, B314(1:10), 'k.', 'MarkerSize', 20);
plot(1.1, B314(11:20), 'r.', 'MarkerSize', 20);
plot(1.9, B413(1:10), 'k.', 'MarkerSize', 20);
plot(2.1, B413(11:20), 'r.', 'MarkerSize', 20);
```

```
plot(2.9, B324(1:10), 'k.', 'MarkerSize', 20);
plot(3.1, B324(11:20), 'r.', 'MarkerSize', 20);
xticks([1.0, 2.0, 3.0]);
xticklabels({'B314', 'B413', 'B324'});
```

Finally, you can run a t-test and compare the two groups:

```
[h,p314] = ttest2(B314(1:10),B314(11:20))
```

This will give you the p-value `(p314)` for the test of a difference (between the two groups) in the modulation (by right arrow input) of the connection from area 3 to 1. As a control you can test a connection that was not modulated:

```
[h,p324] = ttest2(B324(1:10),B324(11:20))
```

Here, we have demonstrated how to extract parameters and then use classical statistics (t-tests) to look for differences between groups. Of course, you could now also extract more parameters and for example train a classifier to distinguish the two groups. This would then be 'Generative Embedding'.