PPI in FSL
NITP Summer Course 2011

Psychophysiological Interaction (PPI) is a type of fMRI functional connectivity analysis that is specifically useful for looking at the statistical dependence of multiple brain regions as modulated by another factor, e.g. a task. This following describes a process for executing a PPI with FSL, based on a block designed experiment.

Main PPI reference

Friston, KJ, Buechel, C, et al. (1997).
Psychophysiological and Modulatory Interactions in Neuroimaging. Neuroimage, 6, 218-229.
http://dx.doi.org/10.1006/nimg.1997.0291

This paper introduced the concept of PPI. It used the dataset in which subjects are asked to either pay attention to a set of moving dots or to simply look without paying attention. The purpose of this lab will be to see if there are any regions which V1 (primary visual cortex) is more functionally connected to during the attention condition vs. the non-attention condition.

Steps

All the relevant materials are in the ppi_lab directory so open terminal and cd into it:
cd ppi_lab

The GLM analysis for the active condition (moving dots) vs. the baseline condition (a fixation cross) has already been run, the result are in: vis_vs_fix.feat
1) Look at the activation map to see which regions were activated by the task:
fslview vis_vs_fix.feat/example_func.nii.gz vis_vs_fix.feat/thresh_zstat1.nii.gz &
We are going to chose an area that we know is involved in vision and test which regions it becomes more functionally connected to when the subject is paying attention to the stimuli vs. passively viewing them. Notice the point 21, 11, 14 – the center of a nice big activation cluster.

PPI - Preprocessing

This dataset has already been preprocessed for you, with the exception of high-pass filtering.
Normally, standard fMRI pre-processing applies - motion correction, skull stripping, spatial smoothing, and high-pass filtering.
Optionally, some choose to remove other covariates of no interest, such as the global signal or mean signals in white matter and CSF.
1) We're going to use FEAT to run the PPI. Launch the FEAT gui from the command line:
Feat & (make sure to use capital 'F')
2) Select 4D data: choose the snff_convert.nii.gz file in ppi_lab/
3) Set the output directory to whatever you want. Total volumes should show as 360, TR is 3.22, leave Delete volumes at 0, and set a high pass filter cutoff of 192; note that this is in seconds, not Hz. Turn high pass filtering on, leave everything else off. This is larger than average (ie, allowing lower frequencies) because we're interested in low frequency functional connections.

Seed ROI selection and time series extraction

Leave the Feat window open for now but move it aside. The goal here is to decide what your region of interest will be. Your ROI might be a voxel (based on peak activation from your fMRI analysis or from a meta-analysis), a sphere around a voxel, or an anatomical region (from a cortical/subcortical parcellation or an atlas). In case any of these commands take to long to run or you have problems, all of the output files are already in the outputs/ directory.

1) To start, we'll extract the timeseries from the voxel that was at the center of the activation cluster from the GLM analysis. To extract the timeseries from the fMRI file for the voxel of interest in V1:
fslmeants -i snff_convert.nii.gz -o v1_point_ts.txt -c 21 11 14
fslmeants is short for fsl mean timeseries and will extract the timeseries from a BOLD file for a chosen voxel, or extract the mean timeseries from the BOLD file for a set of voxels from a given mask.
2) Another common strategy is to form a small sphere around a voxel of interest. To form a sphere mask around a voxel of interest, use the following the fslmaths commands. Note the may take too long to run, the outputs have already been created in the outputs/ directory.
create a mask file that covers a single voxel:
fslmaths snff_convert.nii.gz -mul 0 -add 1 -roi 21 1 11 1 14 1 0 1 21_11_14_point
create a sphere mask around the voxel mask:
fslmaths 21_11_14_point -kernel sphere 6 -fmean 21_11_14_sphere_mask
extract the average timeseries from the fMRI file for all voxels inside the sphere mask:
fslmeants -i snff_convert.nii.gz -o 21_11_14_seed_ts.txt -m 21_11_14_sphere_mask.nii.gz
3) Check out the voxel and sphere masks overlayed on the functional file in fslview to make sure they're in the right spot:
fslview snff_convert.nii.gz 21_11_14_point.nii.gz 21_11_14_sphere_mask.nii.gz &

If you wanted to use an anatomical mask, you could refer back to Wednesday's lab for how to obtain an anatomical region from an atlas and then register it to a subject's BOLD space. Once you have that, you can use fslmeants in the same fashion to obtain the regional timeseries.

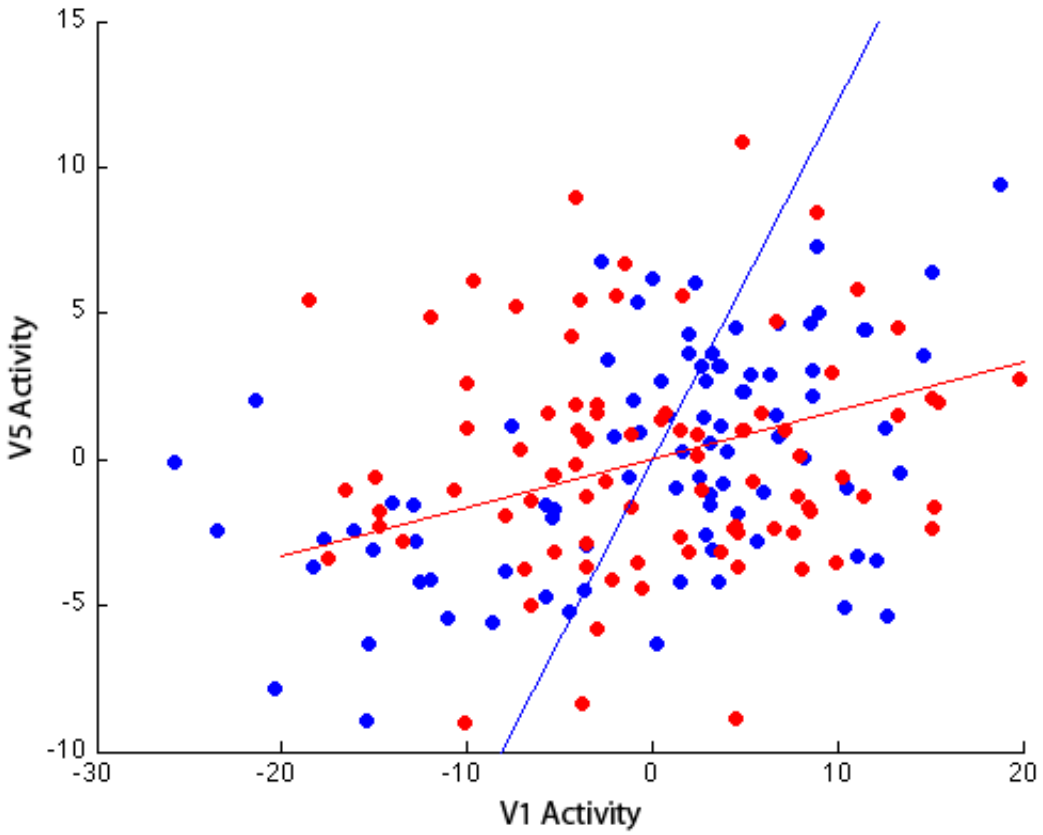
Now we're going to back to the FEAT window and use the seed timeseries file we just created. Ignore prestats for now and go straight to the Stats tab.

PPI model
1) Now, set up the PPI model:
*Use prewhitening
*Click Full model setup. Set up 3 EVs:
*#Psychological Regressor: your task regressor. Call it 'psych'. Custom entry with 1 column, file is outputs/ev_a_vs_na.txt, convolution Double Gamma (or whatever model you prefer), Add temporal derivative and Apply temporal filtering.
*#Physiological Regressor: your seed timeseries. Call it 'phys'. Custom entry with 1 column, file is outputs/21_11_14_seed_ts.txt. Not convolved - can you explain why?
*#Interaction: the intereaction between Psych and Phys, call it 'PPI', basic shape is interaction, make zero Center for Psych (because it already has a mean of 0) and Mean for Phys (because you want to demean the timeseries before creating the interaction term).
*Set up contrasts as:
1 0 0
0 1 0
0 0 1
0 0 -1
Which will give copes for psych mean, phys mean, interaction positive mean, and interaction negative mean. Take note of your model and the shape of each EV.
In the Post-stats tab, set thresholding to none. Because this is a single subject analysis, we won't worry about p-values. Those would normally be controlled for at the group level.

PPI results
1) Don't bother running this model, it's already been run. The results are in: ppi_a_vs_na.feat
Look at the results in fslview.
fslview ppi_a_vs_na.feat/example_func.nii.gz ppi_a_vs_na.feat/stats/zstat3.nii.gz &

(if the filtered_func.nii.gz file opens, you can close it)
Change the threshold range for the zstat3 file to something reasonable like 2.5 – 4. Remember the purpose of a PPI: to find regions whose activity is significantly more functionally coupled with our seed region during a specific condition. Another way to put that is to find regions that are significantly related to our seed region, above and beyond the main effect of the task.
Find a voxel in or near V5/MT, the motion-specific visual processing region. Need help? Try 16,12,17.

One common way to display the results of a PPI is to plot the activity of your seed region and a region that is functionally connected in the PPI. Specifically, the activity is shown for the different conditions of the task, such as attention (in blue below) and no attention (in red below), to show an increased coupling during a specific condition of the task.



We won't cover this in the lab but if you wanted to create this type of plot in Matlab, you'd need three things: 1) a text file for your task EV, 2) a text file with the timeseries for your seed region (V1), 3) a text file with the timeseries for your target region (V5). You could then use the Matlab below to plot this figure:

```
%ppi_region_regressplot.m

%Inputs: task regressor, timeseries for two regional timeseries' from PPI
%Takes timeseries for different conditions from task regressor, runs
%separate regression for each condition based on (demeaned) timeseries
from
%each region
%Plots regression lines for each condition separately
```

```
import ev_a_vs_na.txt v1_point_ts.txt v5_point_ts.txt

v1_a=v1_point_ts(find(ev_a_vs_na==1));
v1_a=v1_a-mean(v1_a);

v1_na=v1_point_ts(find(ev_a_vs_na==-1));
v1_na=v1_na-mean(v1_na);

v5_a=v5_point_ts(find(ev_a_vs_na==1));
v5_a=v5_a-mean(v5_a);

v5_na=v5_point_ts(find(ev_a_vs_na==-1));
v5_na=v5_na-mean(v5_na);

[b,bint,r,rint,stats]=regress(v1_a,[ones(88,1),v5_a]);
[b2,bint2,r2,rint2,stats2]=regress(v1_na,[ones(88,1),v5_na]);

scatter(v1_a,v5_a,'filled','blue')
hold on
scatter(v1_na,v5_na,'filled','red')
x=[-20,20];
y1=b(1)+b(2).*x;
y2=b2(1)+b2(2).*x;
plot(x,y1,'blue')
plot(x,y2,'red')
axis([-30,20,-10,15])
xlabel('V1 activity')
ylabel('V5 activity')
```