The Generalized Psychophysiological Interaction (gPPI) Toolbox

The gPPI toolbox is a MATLAB-based toolbox which uses SPM software to specify and estimate psychophysiological interaction (PPI) models at the first-level. The gPPI toolbox is written by Drs. Donald McLaren and Aaron Schultz. For additional information, see: http://www.brainmap.wisc.edu/PPI

BEFORE USING THE GPPI TOOLBOX…

1. **Estimate level 1 models.** You will need to know the names you gave your conditions when you specified your level 1 model, since the toolbox picks out the psychological timecourses based on these names. If you don't know the names, you can find out in the Matlab command window by loading in the SPM.mat ('load SPM.mat') for a subject, and typing in 'SPM.Sess(1).U.name' where the number '1' refers to the session (i.e., run).
2. **Compute effects of interest F-contrast**. This is an F-contrast that specifies the columns (i.e., predictors) in your design matrix that are expected to relate to neural activity. This is not the same as "covariates of interest", because effects you do not care about but that are expected to relate to neural activity (e.g., motor responses, instruction screens) should also be included in this contrast.
3. **Save binary image files corresponding to the VOIs (seed regions).** These files can be in either .nii or .img/.hdr format. I do this either in xjview or wfu_pickatlas. Note that if you are defining your region based on a single coordinate, you should expand a shape around this coordinate (e.g., sphere or box) in wfu_pickatlas and save the resulting VOI.

USING THE GPPI WRAPPER (gppi_wrapper.m)

A wrapper is a function or script whose main purpose is to call a second function. gppi_wrapper.m is a wrapper for the main function of the gPPI toolbox, PPPI.m. Within the wrapper, I have tried to preserve all of options available in the gPPI toolbox, however, I highly recommend taking a look at the help documentation for PPPI.m (just type 'help PPPI'). Briefly, PPPI works by accepting a structure variable called "P" which stores all of the information needed to extract the VOI and psychological timecourses, compute the PPIs, specify and estimate the first-level models, and compute contrasts. What gppi_wrapper.m does is loop through your subjects and through your VOIs (if you have multiple seed regions), changes values in P corresponding to subjects and VOIs, and then gives P to the function PPPI.m. In order to use gppi_wrapper.m for your own study, you need to modify a number of variables. Below, I provide definitions for some of the variables that are not completely self-explanatory:

subTAG. If subTAG = 'all', then it will run PPPI for all subjects it finds in your study path. Alternatively, you can give subTAG position indices (e.g. subTAG = 1 or subTAG = [1 3 5 10]) to do only a subset.

estimateTAG. This may be useful if you want to use the gPPI toolbox to only extract VOI timecourses and/or compute PPI regressors. By default, the gPPI toolbox software saves .mat files containing the timecourse information for each VOI, as well as the PPI regressor information. This is saved in each subject's first-level analysis folder.

ppi_folder_affix. For the PPI first-level model, gPPI automatically makes a folder in the original first-level model. By default, this folder name is of the form 'PPI_<name of VOI>' (e.g., PPI_amygdala). This variable allows you to specify an affix to that name. I find this useful because you might wind up running multiple models for the same VOI; in these cases you will want to give each model a different affix so you don't estimate a model in a folder that already contains an estimated model (e.g., 'eig_111011').

method. This specifies the method to use to define the psychological timecourse(s) of interest. To use the traditional (condition-contrast) method, method = 'trad'. To use the newer, condition-specific method, method = 'cond'. Please use the new, condition-specific method ☺

conditions. This variable has slightly different meanings depending on the method you are using. If method is 'trad', then this should contain the names of the conditions you would like to use to build the psychological contrast. If method is ‘cond’, then this should contain the names of all of the conditions you would like to estimate PPI effects for. Note that this must be a cell array of strings, where all strings are surrounded by curly brackets ({}) and all strings are surround by single quotes (‘’), e.g. {‘str1’ ‘str2’ ‘str3’}.

weights. If method is ‘cond’, IGNORE THIS. If method is ‘trad’, this variable contains the weights applied to the conditions specified in the variable ‘conditions’. For example, if your conditions are {‘A’ ‘B’}, then to make your psychological variable the contrast A – B, then your weights should be [1 -1]. Note that you build your contrast from multiple conditions, e.g., if your conditions are {‘A’ ‘B’ ‘C’ ‘D’} you could use weights [-1 .33 .33 .33] for the contrast B+C+D – A.

fcontrast. This is the number of the effects-of-interest F-contrast to use to adjust the timecourse of the VOI, e.g., if your F-image is spmF_0005.img, then fcontrast = 5.

extract. This specifies the method for averaging the timecourse of the VOI across voxels. If extract = ‘mean’, then the mean will be taken. If extract = ‘eig’, then the first eigenvariate (from a principal components analysis) will be taken.

masks. This is a cell array of strings specifying the mask(s) to use to constrain localization of your VOI prior to extracting the data. As far as I know, these can be either .nii or .img/.hdr files and they must be located in the original first-level analysis directory. By default, I recommend using at least the ‘mask.img’ file, which contains all voxels that SPM deemed valid when estimating the first-level model. Another common choice is an spmT image for a contrast-of-interest. Note that you can include multiple image files, and what the toolbox does is find the intersection of your VOI image and ALL of your mask images.

threshold. This specifies the threshold(s) to apply to the masks prior to using them to constrain your VOI. For binary images (e.g., ‘mask.img’), I typically input 0.5. For spmT images, you can specify a t-value corresponding to some statistical threshold. For example, a commonly used threshold is 1.65, which corresponds to *p*<.05 (uncorrected). Note that the vector you specify must contain as many values as there are masks, and the order of values should correspond to the order of filenames in the ‘masks’ variable. For example, if masks={‘mask.img’ ‘spmT_0001.img’}, then threshold=[0.5 1.65] would threshold ‘mask.img’ at 0.5 and ‘spmT_0001.img’ at 1.65.

SPECIFYING CONTRASTS

> **NOTE**: If you are doing traditional PPI (method = ‘trad’), there is currently a bug in the gPPI toolbox that prevents automatic contrast computation. You will have to compute contrasts manually or with a batch script. I’d provide a script, but that would severely undermine my pathological disdain for traditional PPI.

Contrasts are specified in gppi_wrapper.m around line 130. Each contrast requires three fields:

```
P.Contrasts(1).name = ‘Faces-Houses’;       % a string used to name your contrast
P.Contrasts(1).left = {‘Faces’};            % the condition(s) on the left of the minus sign
P.Contrasts(1).right = {‘Houses’};          % the condition(s) on the right of the minus sign
```

Left/right is relative to the minus sign in a contrast, e.g. in the contrast Faces – Houses, Faces is on the left and Houses is on the right. You can specify contrasts involving multiple conditions on the left and/or the right, e.g. left = {‘Hot_Faces’ ‘Ugly_Faces’} and right = {‘Hot_Houses’ ‘Ugly_Houses’}. If you want to compare to implicit baseline, put the condition of interest on the left and an empty string (‘’) on the right:

```
P.Contrasts(2).left = {‘Faces’}             % don’t forget to put the string in a cell!
P.Contrasts(2).right = {‘’}                 % this is an empty string inside a cell!
```

Finally, if you have more than one contrast, you need to initialize the contrast before defining the variables by setting the new contrast equal to the first contrast prior to filling the fields, e.g. P.Contrasts(3) = P.Contrasts(1). This is just to ensure that the new contrast has the correct structure.