

Functional Connectivity Pipeline

To run this pipeline, ensure that the following programs are installed

- mricron (sudo apt install mricron)
- FSL (see FSL website)
- Nipype
- MATLAB
 - Brain connectivity toolbox
- Freesurfer
- BIAC pipeline requisites
 - https://wiki.biac.duke.edu/biac:analysis:resting_pipeline

Starting from the raw output, a few things need to be collected for each subject:

- A subject folder containing
 - The RSFC data
- A Freesufersub directory containing
 - The aparc-aseg for the subject
 - A high-quality structural image

It should be noted that the “RSFC Pipeline” folder is synonymous with the scripts folder, and that the analysis folder was subsumed into the RSFC pipeline folder as well.

Preparation

This pipeline is written to work from a “data” directory. This contains separate folders for each subject, which contain key files that are inputs to the pipeline, and are generated from various scripts. Ensure before you begin that such a data directory has been created. This folder can later be deleted, if space is scarce, after extracting desired information.

The following environmental variables need to be created:

- data=<location of your data directory (you need to make this)>
- RSFCscripts=<location of the scripts folder which I sent you>
- output=<location of output folder (you need to make this)>
- Freesurfersub=<location of freesurfer output directory>

SubjectPrep

Subjectprep is the first major file used. This generates a subject directory within your data directory which will have .nii files generated from the .dcm files in the RSFC directory withing the raw data directory. See file for usage.

MaskGen

This script takes the high-quality mgz anatomical image and generates .nii files which correspond to the anatomical image (fslbrain), and masks for the CSF, GM, and WM (refcsf, refgm, and refwm) using FSL’s fast tool through nipype. Ensure Freesurfer is installed for this one. See file for usage.

Freesurfer_ROIs

This python script takes the `aparc+aseg.mgz` file and converts it into `.nii` format, and then generates individual masks based on a supplied text file (see `aparc_labels.txt`) detailing regions of interest and their value in `aparc+aseg`. It then calls `maskcat`, which is assumed to be in the same directory as `Freesurfer_ROIs`, which creates an aggregate mask from the supplied parameters. See file for usage.

Path3

This is the wrapper which calls the resting pipeline, using established parameters. This should be the file which you will edit the most, changing it based on the kind of analysis you want to perform. The output files are detailed on duke's site https://wiki.biac.duke.edu/biac:analysis:resting_pipeline. If you have questions about the nature of a particular output file (or any fr that matter), please feel free to contact me at my cell phone (541) 905-7743 or by email.

`Eye.mat` – used by `resting_pipeline_2mm.py` (identity matrix)

Analysis

regAnalysis.m

This is the main file for performing the correlation analysis. It requires data as well as the contents of the output from the connectivity pipeline. Notably, it requires the following files:

- Parcellation label file (see `aparc_labels.txt`)
- Subject number file (see `subjectnumbers.txt`)
- A data matrix for correlation
 - The script is somewhat tailored to the walking scores, but it can be easily modified to utilize different sources of information
- The fisher-transformed correlation matrix for each subject (see any `zr_matrix.csv`)

It also calls the **viewcorrresults** file, which is much more highly tailored to the walking data, but can also be worked with. I found it to be a useful graphical tool.

ViewCorrResults.m

This is a graphical function tool made to be called by `regAnalysis` to interpret results graphically. The button in the top right toggles between viewing correlates with different supplied data types, and the slider masks which connections are shown based on a p-value of interest. Clicking on a point in the matrix will show which two regions the connection corresponds to, and will display a distribution of the normalized correlation coefficients in a histogram on the right. By using this tool, one can investigate which regions are correlated at increasingly stringent p-value thresholds.

ImportantRegions.m

You will need to edit this file to add where you want to output the correlation table before running it. This script was written to be run immediately after the `regAnalysis` script, as it uses workspace variables created in that script.