Overview: All of the scripts in the toolbox follow a similar framework. This framework consists of a *master* level script with *global* variables that calls upon a *shell* script for all aspects of processing. The *shell* script often calls upon additional scripts for further processing as well. Key components of the *shell* scripts are referenced in this document. However, the vast majority annotation is at the level of the *master* scripts. Ideally, you shouldn’t have to dig any deeper than this top-level script to complete your analyses. Updates will almost always be made to the *shell* scripts or lower, as, in general, the *master* level scripts do little more than define pathways and processing parameters.

**Main Script: Master\_RSA.m**

**Description v2: Updates**

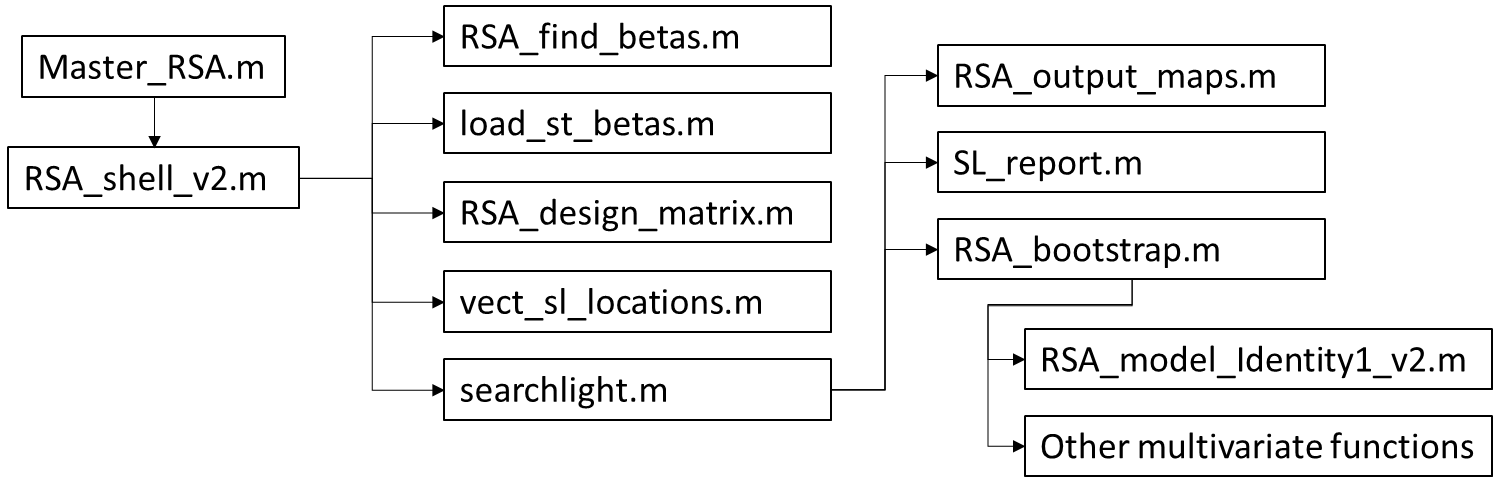
**Description:**

The RSA script is setup to run representational similarity analyses. It has been formatted to look mostly at encoding/retrieval similarity, but can also function with ‘custom’ similarity matrices. The script was also further updated to run multivariate analyses for any ROI. Finally, it was integrated into the graph theory scripts and serves as the fundamental starting point for extracting ROI based beta-series.

One of the most important components of the script is SL.design.cond\_str. This variable defines the trials that should be included in the models. It serves as a critical starting point for all analyses including subsequent graph theory analyses.

**Organization:**

* Master\_RSA.m = define directories, settings, etc.
  + RSA\_shell\_v2.m = wrapper for any subsequent processing
    - RSA\_find\_betas.m = examines the file structure and/or SPM.mat file in order to identify beta volumes that should be included in the analysis
    - load\_st\_betas.m = loads in the single trial beta volumes (SL.files) for subsequent analysis
    - RSA\_design\_matrix.m = creates the majority of the ‘design matrices’. These matrices serve as contrast masks in order to selectively analyze similarity matrices
    - vect\_sl\_locations.m = creates a list of searchlight volumes that are subsequently looped over
    - searchlight.m = the main searchlight script that loops over the entire brain
    - RSA\_output\_maps.m = generates brain volumes to be saved. These brain volumes are filled in as the searchlight volumes are analyzed
    - SL\_report.m = a short quality assurance script that reviews the inputs and provides some output such that the user can verify their suggested inputs match the wanted output
    - RSA\_bootstrap = wrapper script for multivariate analyses. In principle, any multivariate function could be placed within this wrapper.
      * RSA\_model\_Identity1\_v2.m = runs a basic on diagonal / off diagonal analysis of identical vs. non-identical trials.



**Figure 1:** Organizational structure of the RSA script.

**Required Updates & Known Errors:**

* Voxel-level regression analyses need finalized and generalized
* Parallel processing needs added
* Setup for identity1 models needs moved from RSA\_output\_maps.m to RSA\_design\_matrix.m
* 4D volumes are x-flipped, these are very rarely used, but in general, especially for ROI analyses, it is important to ensure that x-dimensions don’t end up flipped. Strongly recommend running some quality assurance checks to verify nothing was flipped.

**Main Script: Master\_Network\_Maker.m (calls ROI\_shell\_v2)**

**Description:** Creates adjacency matrices, this is likely the simplest of all the scripts included. One component that is potentially difficult is ROI selection. Lines 11-49 of the script ROI\_shell\_v2 are dedicated to ROI selection. The variable specified is SL.region.mask (cell string). This is the same variable created in Master\_RSA.m. It is merely a directory list of ROIs. Moreover, its primary purpose is to ensure that the correct files are grabbed for each subject. Simply copying and pasting the contents from Master\_RSA.m should be sufficient to bypass this step.

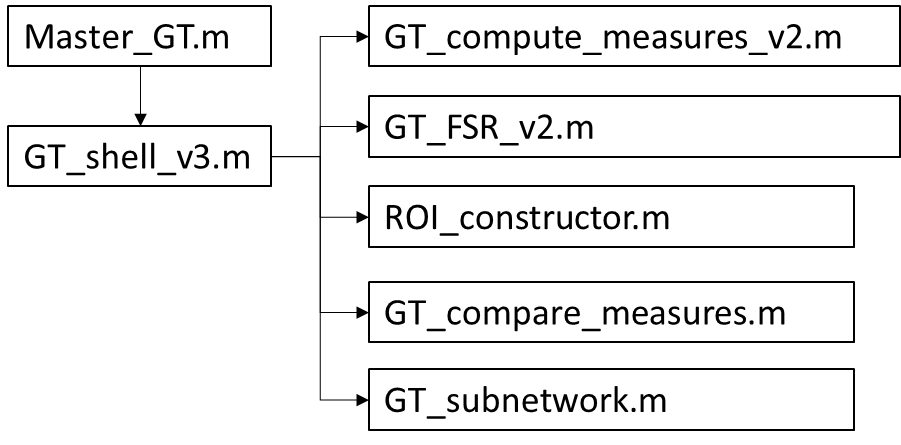
**Main Script: Master\_GT**

**Description:** Computes graph theory measures. Depends heavily upon the Brain Connectivity Toolbox. In principle this script serves as a wrapper in order to run multiple subjects, compute stats, etc. It is also the script most in flux at the moment, thus it is likely ‘buggiest’ of the all those included.

Similar to the Master\_Network\_Maker.m script, the ROI selection needs manual input. In the GT\_shell\_v3.m script, the setting of SL.region.mask occurs from line 31-60.

**Organization:**

* Master\_GT.m = define directories, settings, etc
  + GT\_shell\_v3.m = wrapper for all GT functions
  + GT\_compute\_measures\_v2.m = wrapper for the brain connectivity toolbox. For the most part, this is just a list of switch statements calling the BCT. The switch statements start at line 28.
  + GT\_FSR\_v2.m = computes FSR
  + ROI\_constructor.m = maps computed measures back onto brain-space and saves data out to .csv files
  + GT\_compare\_measures.m = runs all of the stats, sets of stats include…
    - T-values
    - Ranksum
    - Permutation: lines 68-76. This is sometimes nice to turnoff in case you’re running a lot of test or just curious on ‘how’ the data looks. This step can potentially take a very time. It also produces some ugly output, but functions fine.
  + GT\_subnetwork.m = creates subnetworks that can be visualized in gephi



**Figure 2:** Organizational structure of the GT script.

**Required Updates & Known Errors:**

* Z-score manipulations prior to GT\_compare\_measures.m in order to examine ‘shifts’ as opposed to absolute valeus