

# Acronym Interface Package:

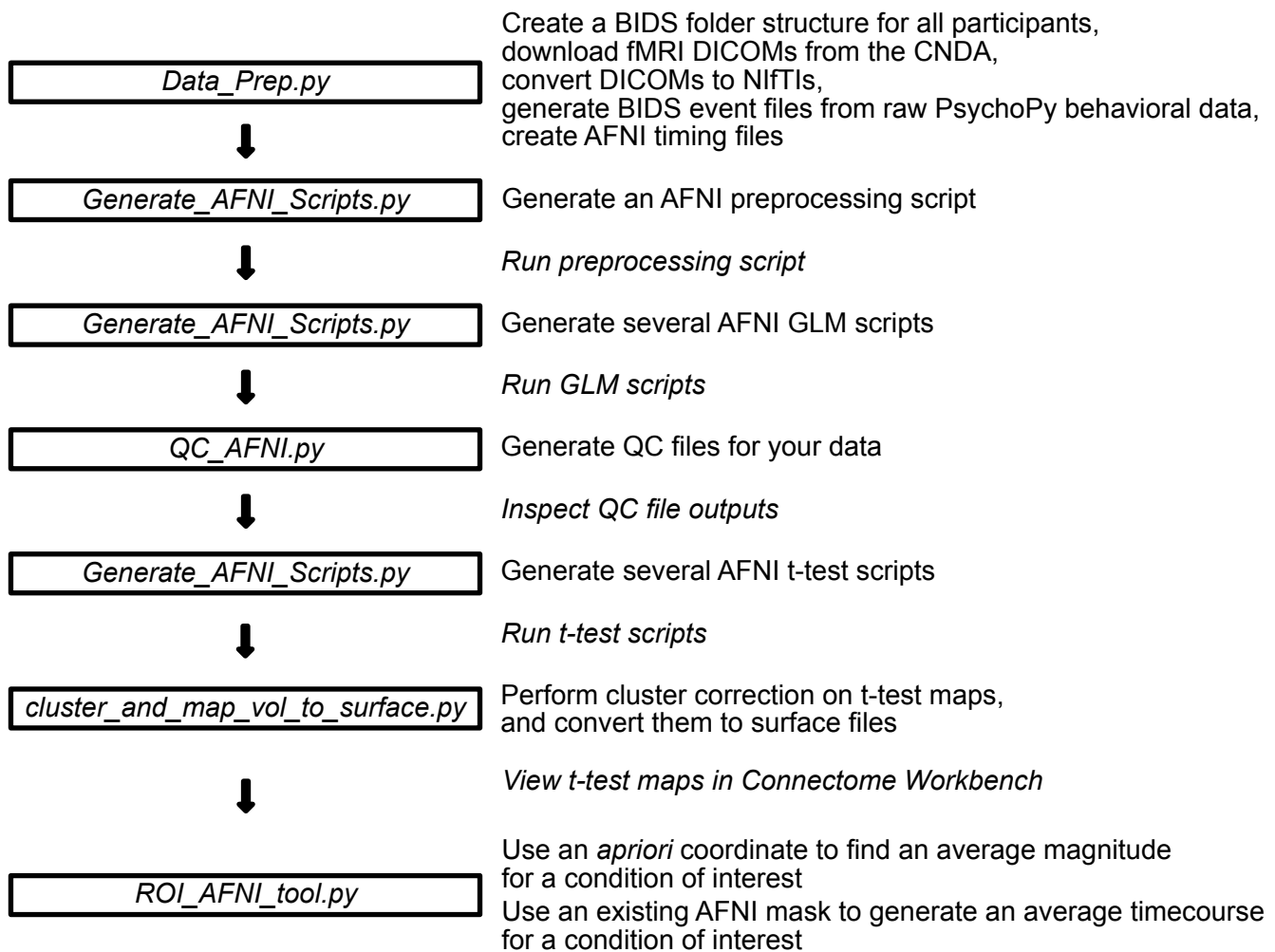
*MCL tool for BIDS, AFNI, DICOMs, NiftIs, CNDA, GLMs, and more!*

The Acronym Interface Package is intended to simplify, streamline, and standardize fMRI analysis in the MCL. It includes tools that work with raw fMRI data, raw behavioral data, AFNI, and Connectome Workbench. Additionally, this package follows the Brain Imaging Data Structure (BIDS) standard; however, none of these tools require the input data to be in BIDS format.

Currently, the Acronym Interface Package contains the following tools:

- Data\_Prep.py:** Downloads, organizes, and prepares raw fMRI and behavioral data for analysis
- Generate\_AFNI\_Scripts.py:** Creates preprocessing, GLM, and t-test scripts for AFNI
- ROI\_AFNI\_tool.py:** Extracts average magnitudes and timecourses from ROIs
- map\_vol\_to\_surface.py:** Used to convert AFNI or NiftI volume files into surface (GIFTI) files for use in Connectome Workbench
- cluster\_and\_map\_vol\_to\_surface.py:** Applies cluster correction to AFNI t-test files, and then converts them to GIFTI surface files
- QC\_AFNI.py:** Automatically outputs a set of useful quality control files after AFNI analysis has been completed
- LME\_ROI\_magnitudes.py:** A specialized tool for extracting trial-level beta estimates for use in linear mixed effect modeling

An example workflow is shown below to demonstrate how these tools can be used:



*Graph magnitudes and timecourses in program of your choice [e.g. R, GraphPad Prism, Excel (ha!)]*

# Acronym Interface Package:

*MCL tool for BIDS, AFNI, DICOMs, NiftIs, CNDA, GLMs, and more!*

## Data\_Prep\_Bundle

GUI script  
(run this!)

Functional  
script

### Data\_Prep.py

GUI to prepare raw MRI and behavioral data

Takes as input:

DICOMs (downloaded by script)  
CSV files containing behavioral data (from PsychoPy)  
LOG files containing timing data (from PsychoPy)

### Setup\_BIDS\_Folder\_Structure.py

-Create a folder structure for the participant(s) data,  
following the BIDS specification

### DICOM\_Files.py

-Download participant fMRI data (DICOMs) from  
remote CNDA repository  
-Back up DICOMs to the remote Icarus server

### NiftI\_Files.py

-Convert DICOMs to NIFTI format using *dcm2niix*  
-Reorganize NIFTI files into BIDS folder structure

### Generate\_BIDS\_Event\_Files.py

-Code behavioral data (from .csv and .log files)  
and generate BIDS-style event files

### Create\_AFNI\_Timing\_Files.py

-Use BIDS event files to generate  
AFNI timing (onset) files

These scripts  
**MUST**  
be edited for your  
individual experiment

**To use these scripts,**  
**you must have the following**  
**installed:**

-*pydicom*: used to read DICOM  
header info  
-*dcm2niix*: used to convert  
DICOMs to NiftIs  
-*sshpas*: used to upload  
files to Icarus

# Acronym Interface Package:

*MCL tool for BIDS, AFNI, DICOMs, NIfTIs, CNDA, GLMs, and more!*

## AFNI\_Script\_Bundle

GUI script  
(run this!)

### Generate\_AFNI\_Scripts.py

GUI to create AFNI processing scripts

Creates:  
-Preprocessing scripts  
-GLM scripts  
-t-test scripts

Preprocessing and GLM scripts use a template (.txt file) which can be modified to alter AFNI script outputs

For scripts with "Auto" input options, a spreadsheet can be used to enter data rather than the GUI provided

#### Parameter input options:

	Manual	Auto
Preprocessing scripts	✓	
GLM scripts	✓	✓
t-test scripts	✓	✓

Additionally, all scripts can be backed up to the remote Icarus server

**To use this script,**  
**you must have the following**  
**installed:**

-sshpass: used to upload  
files to Icarus

# Acronym Interface Package:

*MCL tool for BIDS, AFNI, DICOMs, NiftIs, CNDA, GLMs, and more!*

GUI script  
(run this!)

## **ROI\_AFNI\_tool.py**

GUI to extract group average  
magnitudes and timecourses

ROIs can be specified either with an AFNI mask file or an XYZ coordinate in a text file.  
If a coordinate is provided, a sphere will be drawn around the coordinate to create a mask.

Additionally, conditions of interest can be chosen (e.g. hits, high-confidence false alarms, etc) from  
the conditions provided in your GLM.

The tool will then extract the average signal within the mask area, for the user-specified condition,  
from each participant's data file. This is then averaged across all subjects.

The signal extracted will depend on the data used:

Input	Output
Stats file from GLM with Gamma basis function	Activation magnitude
Iresp file from GLM with Tent basis function	Timecourse

The outputs will then be aggregated into a master CSV file (organized by ROI/condition),  
and the results can be plotted in the program of your choice.

# Acronym Interface Package:

*MCL tool for BIDS, AFNI, DICOMs, NIfTIs, CNDA, GLMs, and more!*

## Clustering\_and\_Vol\_Surf\_Convert

Commandline  
script  
(run this!)

### map\_vol\_to\_surface.py

Converts AFNI or NIfTI volumetric files  
to GIFTI surface files

This is a Python version of Evan Gordon's Matlab script *map\_vol\_to\_surface\_march2016.m*. It will convert a specified volumetric neuroimaging file (AFNI .HEAD/.BRIK or NIfTI .nii) into a surface file (GIFTI) which can be viewed in programs like Caret or Connectome Workbench.

### cluster\_and\_map\_vol\_to\_surface.py

Applies cluster correction to AFNI files,  
then converts them to GIFTI surface files

This script is primarily intended for the output of AFNI t-tests which include the '-Clustsim' argument. It will extract the cluster simulation information from the file header and use it to perform the cluster correction on the data (this is the equivalent of using the "Clusterize" panel in the AFNI GUI, outputting the mask, and then creating a new dataset only containing those voxels).

Once the cluster correction is complete, the *map\_vol\_to\_surface.py* script is run to convert the result to a surface file.

Additionally, this command can be used with a "loop" argument, to perform these operations on all AFNI files within the current directory.

For either script, simply executing the script without any arguments will provide a detailed help output.

# Acronym Interface Package:

*MCL tool for BIDS, AFNI, DICOMs, NifTIs, CNDA, GLMs, and more!*

GUI script  
(run this!)

## QC\_AFNI.py

Creates useful quality control output files from a processed AFNI dataset

This script should be run on a completed AFNI dataset (preprocessing and at least one GLM generated).

All files will be created in a "QC\_AFNI" folder inside the "subject\_results" folder.

This script does the following:

- Copies all of the outputs from AFNI proc scripts into a "proc\_outputs" folder.
- Copies all movement files ("dfiles") in a "dfiles" folder
- Creates review tables for each GLM for each subject using AFNI's [gen\\_ss\\_review\\_scripts.py](#), [@ss\\_review\\_basic](#), and [gen\\_ss\\_review\\_table.py](#) commands. Additionally, all outputs from [gen\\_ss\\_review\\_scripts.py](#) are saved in a "review\_files" folder within each GLM folder.
- Creates "snapshots" of alignment between each subject's anatomical file and the template selected, as well as each EPI run with the anatomical file. This is done using AFNI's [@snapshot\\_volreg](#) command.
- Summarize the results of the AFNI command [@radial\\_correlate](#) (run during preprocessing) for each subject into a text file ("radial\_correlate\_summary.txt").
- Create motion and outlier plots for each EPI run based on the thresholds selected during processing.

# Acronym Interface Package:

*MCL tool for BIDS, AFNI, DICOMs, NIfTIs, CNDA, GLMs, and more!*

GUI script  
(run this!)

## **LME\_ROI\_magnitudes.py**

Extracts trial-level magnitudes from an AFNI dataset  
that can be used in linear mixed effects modeling

This script is designed to be run on an AFNI dataset which has beta coefficients estimated at each trial (rather than for general conditions).

When creating this AFNI dataset, there should only be one timing file used (with an onset for every trial in every run). Additionally, the "-stim\_times" argument for 3dDeconvolve (part of the GLM script) should be changed to "-stim\_times\_IM" in order to output an estimate for every trial.

This script will then produce magnitude estimates (similar to *ROI\_AFNI\_tool.py*), except that they will not be averaged across participants. You can provide multiple ROIs in the form of AFNI mask files, and an average magnitude will be calculated for each trial for each participant.

The results will be organized into a CSV file and placed in a folder named "LME\_results" within your "subject\_results" folder. This can then be used in a program like R to perform a linear mixed effects modeling analysis.