Background info:

Since CLPipe uses python, you must use the following command to enable python before anything else can be run:

module load python/3.6.6

Creating a config file:

Config files are json files that can store your preferences for each process. They can be especially useful when needing to remember what parameters you have set for a specific command. Creating a new config file with default values is easy, simply use the command:

grab\_config\_file -outputFile [path & name].json

Example: grab\_config\_file -outputFile ./newConfigFile.json

Converting Dicom to Nifti Files:

Converting from dicom to nifti uses a command with a few parameters but we’ll break it down.

(More info on the differences between dicom and nifti & why we convert)

dicom\_to\_nifti\_to\_bids\_converter -config\_file [path & name].json -heuristic\_file [path & name].py -dicom\_directory [string] -output\_directory [path] -submit -debug

Parameters:

-config\_file: the location and name of your config file

Example: ./newConfigFile.json

-heuristic\_file: location and name of the heuristic file(should be given to you)

Example: ucla\_schizkidsconv.py

-dicom\_directory: the path of the dicom files

Example: ./raw\_dicoms/{subject}\_{session}/\*/\*/\*/\*/\*

-output\_directory: path for where you want the bids output to be generated

Example: ./outputBids/

Validating BIDS:

Brain Imaging Data Structure(BIDS) is a way of organizing and standardizing neuroimaging data. CLPipe uses this data format for Pre- and Post-processing. BIDS validation is used to ensure that dicom to nifti conversion was successful before running the rest of the processing commands.

bids\_validate -config\_file [path & name].json -submit

Example: bids\_validate -config\_file ./newConfigFile.json -submit

Preprocessing with fMRIprep:

Now that we have converted our files to bids/nifti format and verified the correct BIDS structure, we can use fmriprep for preprocessing with the following command:

(more info on preprocessing)

fmriprep\_process -config\_file [path & name].json -bids\_dir [path] -working\_dir [path] -output\_dir [path] -log\_output\_dir [path] -submit

Parameters:

-config\_file: the location and name of your config file

Example: ./newConfigFile.json

-bids\_dir: the path of the bids output to be preprocessed

Example: ./outputBids/

-working\_dir: a folder fmriprep can create and use to process files

Example: ./workingDirectory/

-output\_dir: path for where you want the preprocessed output to be generated

Example: ./preprocessedOutput/

-log\_output\_dir: path for where you want the log files to be generated

Example: ./preprocessing\_logs/

Postprocessing with fMRIprep:

Postprocessing is very similar to preprocessing as far as commands go with a few minor changes:

(more info on postprocessing)

fmri\_postprocess -config\_file [path & name].json -target\_dir [path] -output\_dir [path] -log\_output\_dir [path] -submit

Parameters:

-config\_file: the location and name of your config file

Example: ./newConfigFile.json

-target\_dir: the path of the preprocessed files to be postprocessed

Example: ./preprocessedOutput/

-output\_dir: path for where you want the preprocessed output to be generated

Example: ./postprocessedOutput/

-log\_output\_dir: path for where you want the log files to be generated

Example: ./postprocessing\_logs/

ROI Extraction:

Finally, this command can be used to perform roi extraction on your postprocessed files:

(more info on roi extraction)

Fmri\_roi\_extraction -config\_file [path & name].json -target\_dir [path] -target\_suffix [path] -output\_dir [path] -log\_output\_dir [path] -submit

Parameters:

-config\_file: the location and name of your config file

Example: ./newConfigFile.json

-target\_dir: the path of the postprocessed files to undergo roi extraction

Example: ./postprocessedOutput/

-target\_suffix: the ‘ending’ of all postprocessed files to use

Example: preproc\_bold\_.nii

-output\_dir: path for where you want the roi extraction output to be generated

Example: ./RoiExOutput/

-log\_output\_dir: path for where you want the log files to be generated

Example: ./RoiEx\_logs/

NOTE(suggestion for Teague): Maybe streamline the commands a bit to make them easier

Instead of bids\_dir in preprocessing use target dir like postprocessing

Instead of dicom\_directory and output\_dir, make them all dir or directory