ACAP Preprocessing Work Flow

1) **Run batchBIDSdcm.m**
    a. Dependencies: BIDSdcm2nii.m, dcm2nii package
    b. Update batchBIDSdcm.m SOURCE_DIR, TAR_DIR and PAR_??? as required
    c. ???_LIST is an *n* x 4 cell array with column number:
        i. Regex expression to find dicom files
        ii. BIDS subfolder where the modality is stores (*e.g.*' func', 'anat')
        iii. BIDS name of nifti file in printf format
        iv. BIDS name of nifti file in printf format if multiple runs exist

    and the *n* rows correspond to *n* separate modalities *e.g.:*

(.*)ep2d(.*)Resting_State(.*)	func	sub-%s_ses-%s_task-resting_bold	sub-%s_ses-%s_task-resting_run-%02d_bold
(.*)_mprage_(.*)iso0.8$	anat	sub-%s_ses-%s_T1w	sub-%s_ses-%s_run-%02d_T1w

    d. Switch to dcm2niix?

2) **Run BOLDpreprocess.sh**
    a. Dependencies: BOLDpreprocess.py, optiBET.sh
    b. Usage:

```
alexbarton$ bash BOLDpreprocess.sh [BIDS-ROOT-DIRECTORY] [#-OF-PARALLEL-PROCESSES] [optiBET.sh-LOCATION]
```

3) **Manually Check T1's and make .csv list**
    a. RepeatList.csv is an *n* x 3 .csv with column number:
        i. Subject Name
        ii. Session Name
        iii. Run # to use

    and the *n* rows being the number of subjects with repeat scans e.*g.:*

ACAP1215	B0	1
ACAP1217	B0	1
ACAP2136	M6	2
ACAP2137	M6	2
ACAP2140	M6	2
ACAP2142	M6	2
ACAP2143	M6	4

4) **Run icaPrep.py**
    a. Makes the .fsf files for each participant based on a template
    b. All fields that are replaced are strings in the form of `XXXFIELDNAMEXXX` (*e.g.* XXXNVOLSXXX)
    c. Usage:

```
alexbarton$ python icaPrep.py [BIDS-ROOT-DIRECTORY] -t [T1-REPEATS.csv] -f [FSF-TEMPLATE-FILE.fsf]
```

5) **Make list of .fsf files**
    a. I use:

```
alexbarton$ ls -1d [/PATH/TO/FILES/.../sub*/ses*/*.fsf] > [FSF_LIST.txt]
```

6) **Run:**

```
alexbarton$ parallel -j[#-OF-PARALLEL-PROCESSES] feat :::: [FSF_LIST.txt]
```

Python packages:

- os
- sys
- argparse
- shutil
- csv
- nibabel
- re
- numpy
- glob
- nipype (?unsure if needed?)