Manual for IUSM Connectivity Pipline

Zikai Lin, Evgeny Chumin, Mario Dzemidzic, John West,
Joaquin Goni, Enrico Amico

June 14, 2019

Contents

Contributors:

- Evgeny Chumin, Indiana University School of Medicine, 2018
- John West, Indiana University School of Medicine, 2018
- Zikai Lin, Indiana University School of Medicine, 2018
- Mario Dzemidzic, Indiana University School of Medicine, 2018
- Joaquin Goni, Purdue University, 2018
- Enrico Amico, Purdue University, 2018

1 Installation

1.1 Requirements

This code has been developed to operate with the following software:

- FSL version 5.0.10/11
- AFNI
- dcm2niix (part of MRIcroGL)
- Camino
- Camino-TrackVis

1.2 Unix/Linux/Mac

Launch the terminal, go to the directory you want to put the software in, and type in:

```
git clone https://github.com/echumin/IUSM-connectivity-pipeline.git
```

1.3 Windows

If you have the Github Desktop installed, you can go to the repository, download or clone the repository via Github Desktop.

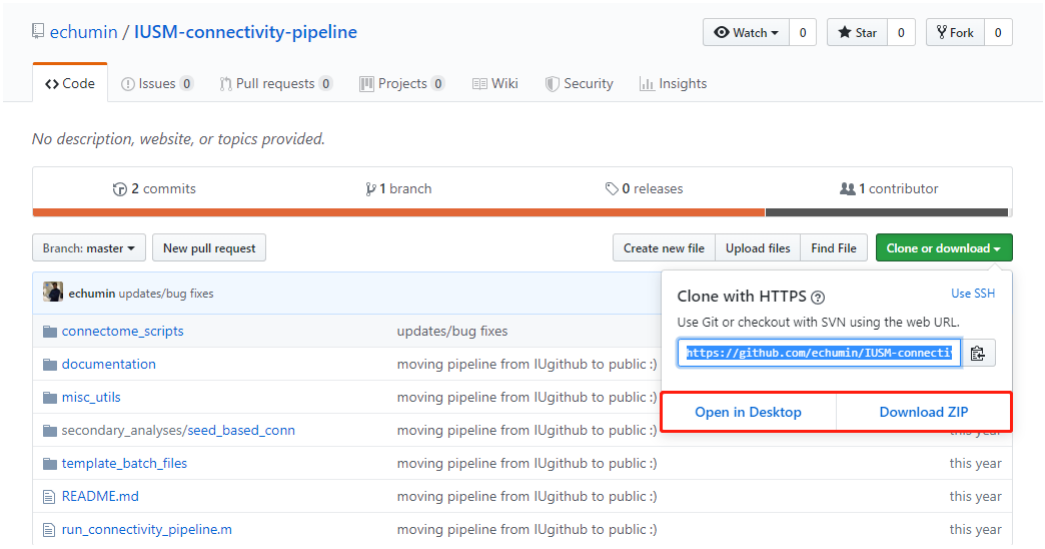

Figure 1: Download the pipeline under Windows

2 Run IUSM Connectivity Pipeline

2.1 Overview

Located within the `template_batch_files` subdirectory are two Matlab script files:

- `batch_setup.m`
- `system_and_sample_setup_local.m`

These files must be modified to contain appropriate paths for your software and data. They also contain extensive documentation on usage, directory structure set-up, software requirements, etc... We recommend that these files are copied to a separate project directory, where they can be easily associated with your data.

Once modified, the `run_connectivity_pipeline.m` function can be ran. Through the file selection user interface, select your modified files one at a time, after which processing will begin.

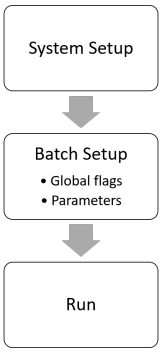

Figure 2: General workflow

2.2 System Setup

Before running IUSM Connectivity Pipeline, the first thing to do is to set up the system, including paths to the required software (FSL, AFNI, etc.)

Go to

`*/IUSM-connectivity-pipeline/template_batch_files/system_and_sample_set_up_local.m`

2.2.1 Scripts and required software

Under the first section of this script, set up the paths for connectome scripts and required software.

```
%%
            %-----------------------------------------------%
            %  SET PATH TO THE CONNECTOME SCRIPTS DIRECTORY  %
            %-----------------------------------------------%
    % Add path to connectome scripts directory
paths.scripts = '/usr/local/IUSM-connectivity-pipeline/connectome_scripts';
addpath(paths.scripts);
    % path to use MRIread MRIwrite
addpath(fullfile(paths.scripts,'toolbox_matlab_nifti/'));
    % path to templates in MNI
paths.MNIparcs =fullfile(paths.scripts,'templates/MNIparcs');
    % path to T1 denoiser
addpath(genpath(fullfile(paths.scripts,'/MRIDenoisingPackage')));

%%  (This may/should already be set in your .bashrc)
    % path to FSL bin directory
paths.FSL = '/usr/local/fsl/bin';
    % FSL setup
FSLsetup = 'FSLDIR=/usr/local/fsl; . ${FSLDIR}/etc/fslconf/fsl.sh; PATH=${FSLDIR}/bin:${PATH}; export FSLDIR PAT
%FSLsetup = 'FSLDIR=/data04/Zikai/IUSM-connectivity-pipeline/fsl/; . ${FSLDIR}/etc/fslconf/fsl.sh; PATH=${FSLDIF

    % Path to feat
paths.feat = '/usr/local/fsl/bin/feat';
    % Path to AFNI
paths.AFNI = '/usr/local/afni';
    % Path to MRIcroGL
paths.MRIcroGL = '/usr/local/mricrogl';
    % Camino setup
paths.CaminoSetup=sprintf('PATH=%s:${PATH}',fullfile('/usr/local/camino','bin'));
    % CaminoTrackVis setup
paths.CamTrackSetup=sprintf('PATH=%s:${PATH}',fullfile('/usr/local/camino-trackvis','bin'));
    % DTItk setup
paths.DTItkSetup=sprintf('PATH=%s:${PATH}',fullfile('/usr/local/dtitk','bin'));
```

Figure 3: System paths setup

Common configuration options need to be changed prior to running the pipeline:

- `paths.scripts`: Path to connectome scripts directory (The sub-folder with scripts is included under the IUSM-connectivity-pipeline).
- `paths.FSL`: Path to FSL bin directory.
- `paths.feat`: Path to `feat`.
- `paths.AFNI`: Path to `AFNI`.
- `paths.MRIcroGL`: Path to `MRIcroGL`.
- `paths.CaminoSetup`: Comino setup, specify the PATH of Camino.
- `paths.CamTrackSetup`: CaminoTrackVis setup, specify the PATH of CaminoTrackVis.
- `paths.DTItkSetup`: DTItk setup, specify the PATH of DTItk.

2.2.2 ICA-AROMA System Setup (Optional)

ICA-AROMA (i.e. 'ICA-based Automatic Removal Of Motion Artifacts') is a recently developed tool for removing motion artifacts from fMRI data[2, 1].

```
%% ICA-AROMA paths set up
    % path to ICA-AROMA
paths.aroma = '/usr/local/fsl/ICA-AROMA/ICA_AROMA.py'; % the program of ica-aroma has to be a python files
    % path to standard images
paths.stdImg = '/usr/local/fsl/data/standard/MNI152_T1_2mm_brain'; % paths to standard image
    % ICA-AROMA directory name (optional, if ICA-AROMA has been processed
    % prior to running to pipeline
configs.name.ica_aroma_folder = 'ICA_AROMA';
```

Figure 4: ICA-AROMA setup

Common configuration options need to be set if ICA-AROMA is run:

- `paths.aroma` (Required): Path to ICA_AROMA.py, see details in ICA-AROMA Manual.
- `paths.stdImg` (Optional if feat is done): Path to standard images.
- `paths.melodicDir`: Path to feat input, ICA-AROMA requires fMRI image preprocessing, includes motion correction, 4D mean intensity normalization, spatial smoothing.
- `paths.SingleMelodic_list`: Directories of single single-session Melodic output, this will be a list of paths, which should be consistent with the **order** of EPI scans.
- `configs.name.ica_aroma_folder`: ICA-AROMA will create a sub-folder and put the output inside. User, however, can specify the name of this folder.

2.2.3 Subjects Setup

Under this section, user will needs to configure the setting of the following:

1. `paths.data`: Paths of subjects' data
2. `subjectList`: Either generate a list of subjects from directories in path or process specific subject or set of subjects.

```
%%
                    %-------------------------------%
                    %  SELECT SUBJECT DIRECTORIES   %
                    %-------------------------------%
    % Set the path to the directory containing you subjects.
paths.data = '/XXXX/CONNECTIVITY/datadir/'; Specify parent data directory
    % generate a list of subjects from directories in path
subjectList =dir(paths.data); subjectList(1:2)=[]; %#ok<*NASGU> %remove '.' and '..'

    % If you wish to exclude subjects from the above generated list, use
    % the below line, replacing SUBJECT1 with the subject you want to
    % exclude. Copy and paste the line several times to exclude multiple
    % subjects.

% idx=find(strcmp({subjectList.name},'SUBJECT1')==1); subjectList(idx)=[];

    % If you only wish to process a specific subject or set of subjects,
    % use the following three lines as example. If processing more that 2
    % subjects copy and paste the second line as necessary.

clear subjectList %remove the above generated list
subjects = ["NNNN0001"; "NNNN0002"]; Put subjects' names here, separated by ";"

% A more convenient  way for user to define subjectList
for i = 1:length(subjects)
    subjectList(i).name = char(subjects(i));
end
```

Figure 5: Subjects setup

2.2.4 Directory Structure Setup

The last section of the system setup is to setup the directory structure, i.e. structure of `paths.data`. In general, as shown in the comments.

Among these, T1 is required.

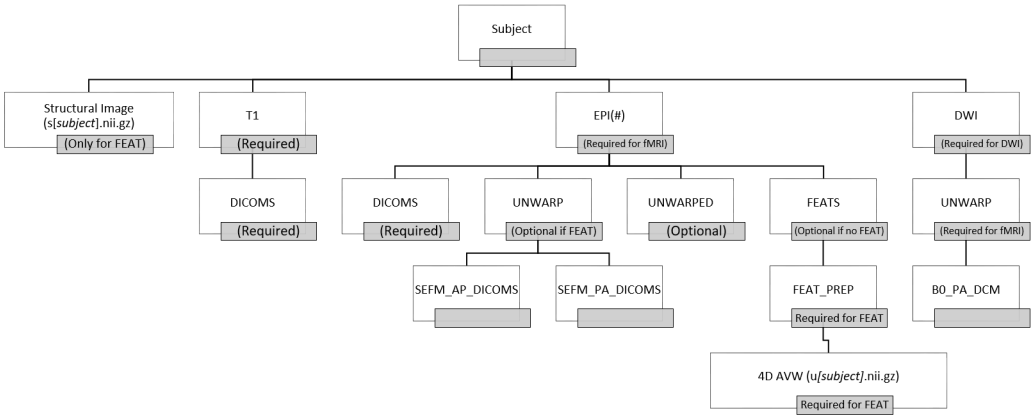

Figure 6: Directory structure

```
%%
                    %------------------------------%
                    %  SET UP DIRECTORY STRUCTURE  %
                    %------------------------------%
% The following diagrapm is a sample directory tree for a single subject.
% Following that are configs you can use to set your own names if different
% from sample structure.

% SUBJECT1 -- T1 -- DICOMS
%          |
%          -- EPI(#) -- DICOMS (May have multiple EPI scans)
%          |          |
%          |          |                (SPIN-ECHO)        (GRADIENT ECHO)
%          |          -- UNWARP -- SEFM_AP_DICOMS (OR) GREFM_MAG_DICOMS
%          |          |          |
%          |          |          -- SEFM_PA_DICOMS (OR) GREFM_PHASE_DICOMS
%          |          |
%          |          -- UNWARPED uf*.nii.gz (MELODIC UNWARPED IMAGES)
%          |          |
%          |          |
%          |          -- FEAT -- FEAT_PREP
%          |
%          -- DWI -- DICOMS
%                 |
%                 -- UNWARP -- B0_PA_DCM

configs.name.T1 = 'T1';
configs.name.epiFolder = 'EPI';
    configs.name.sefmFolder = 'UNWARP'; % Reserved for Field Mapping series
        configs.name.APdcm = 'SEFM_AP_DICOMS'; % Spin Echo A-P
        configs.name.PAdcm = 'SEFM_PA_DICOMS'; % Spin Echo P-A
        configs.name.GREmagdcm = 'GREFM_MAG_DICOMS'; % Gradient echo FM magnitude series
        configs.name.GREphasedcm = 'GREFM_PHASE_DICOMS'; % Gradient echo FM phase map series
    configs.name.melodicUnwarpedFolder = 'UNWARPED'; % Unwarped images (from Melodic)
configs.name.DWI = 'DWI';
    configs.name.unwarpFolder = 'UNWARP';
        configs.name.dcmPA = 'B0_PA_DCM'; % b0 opposite phase encoding

configs.name.dcmFolder = 'DICOMS';
configs.name.dcmFiles = 'dcm'; % Dicom file extension
configs.name.niiFiles = 'nii'; % Nifti-1 file extension
```

Figure 7: Directory structure setup

2.3 Configuration Batch Setup

The `batch_set_up.m` contains all flags and configurations required by the pipeline to process the data. You may edit this as necessary depending on what portions of the pipeline you wish to run.

2.3.1 Parcellation

If you want to introduce a new parcellation into the pipeline, follow these steps and refer to existing parcellations as examples.

1. In connectome_scripts/templates/MNIparcs:
   - Create a directory with the same name as the parcellation (just without the .nii.gz)
   - In that directory place the .nii.gz parcellation image that is in MNI152 space (e.g. registered to the MNI152_T1_1mm.nii.gz image, which can be found in FSL data/standard directory or in MNI_templaces within MNIparcs.
2. In this batch you will been three variables to describe the parcellation:
   - plabel - a short nickname for the parcellation that will be used in the file naming convention within the pipeline.
   - pdir - the name of the directory you were asked to create in step 1 (name of the parcellation volume without the .nii.gz).
   - pcort - YES=1 ; NO=0; is this a cortex only parcellation? This means no cerebellum and no subcortical. Setting this to YES, will result in an attempt to clean the bleeding of the parcellation into subcortical and cerebellar regions, due to transformations and dilations.
3. For visualization in fMRI_B a sorting .mat file can be provided. It must contain the following variable:
   - ROIs - A vector where each row represents a node and the value corresponds to the grouping/network label.

   The grouping .mat file must be in the nodal parcellation directory for those nodes to be ordered according to that parcellation during the visualization of the matrices.

If you want to specify the parcellation, then you will need to modify the following scripts. You will need to specify the label name, the directory name, whether it is cortex only parcellation, and whether it is nodal parcellation:

```
%% SET WHICH PARCELLATIONS YOU WANT TO USE
% shen 1_5 286 region parcellation with modified subcortical
parcs.plabel(1).name='shen_278';
parcs.pdir(1).name='shen_MNI152_org';
parcs.pcort(1).true=0;
parcs.pnodal(1).true=1;

%yeo7 resting state network parcellation
parcs.plabel(2).name='yeo7';
parcs.pdir(2).name='yeo7_MNI152';
parcs.pcort(2).true=1;
parcs.pnodal(2).true=0;

% yeo 17 resting state network parcellation
parcs.plabel(3).name='yeo17';
parcs.pdir(3).name='yeo17_MNI152';
parcs.pcort(3).true=1;
parcs.pnodal(3).true=0;
```

Figure 8: Parcellation Setup

2.3.2 Global Batch Flags

In this section, user will be able to globally setup the specific pre-processing steps. To turn on the flag, set to 1, otherwise 0.

For each pre-processing steps:

1. T1 Preprocessing:

- **T1 Prepare A**:
  - dicom to nifti conversion
  - denoising
  - brain extraction
  - FSL_anat
- **T1 Prepare B**:
  - Registration to MNI
  - segmentation
  - parcellation

2. fMRI Preprocessing:

- **fMRI A**:
  - dicom to nifti conversion
  - headers read-in
  - ICA-AROMA
  - top-up, motion correction
  - slice timing correction
  - register T1 images
  - global signal regression (GSR)
  - motion regressors
  - band-pass filter
  - tissue regressors
  - spatial smoothing
  - ROI parcellations
- **fMRI B** (Figures and matrices):

3. DWI Preprocessing (Under construction):

```
%%
                    %-----------------------%
                    %  SELECT GLOBAL FLAGS  %
                    %-----------------------%
                    %       =1 is ON        %
                    %       =0 is OFF       %

    %  T1   %
flags.global.T1_prepare_A = 0;
flags.global.T1_prepare_B = 0;
    %  fMRI  %
flags.global.fMRI_A = 0;
flags.global.fMRI_B = 1;
    %  DWI   %
flags.global.DWI_A = 0;
flags.global.DWI_B = 0;
flags.global.DWI_C = 0;

    % Parallel %
configs.parallel = 1;
configs.UsageCPU = 0.67;
```

Figure 9: Global Flags Setup

2.4 Start the pipeline

Once you finish setting up system paths and configuration. Go to `run_connectivity_pipeline.m`, press **F5** or click **Run**.

1. Select `system_and_sample_setup_local.m`

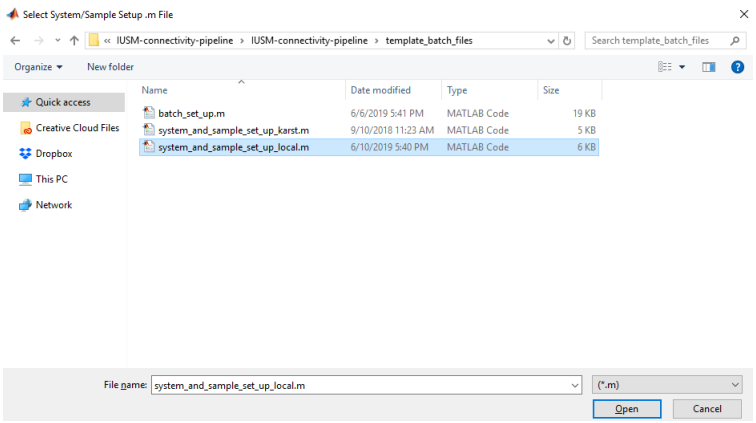

Figure 10: Select system setup scripts

2. Select `batch_set_up.m`

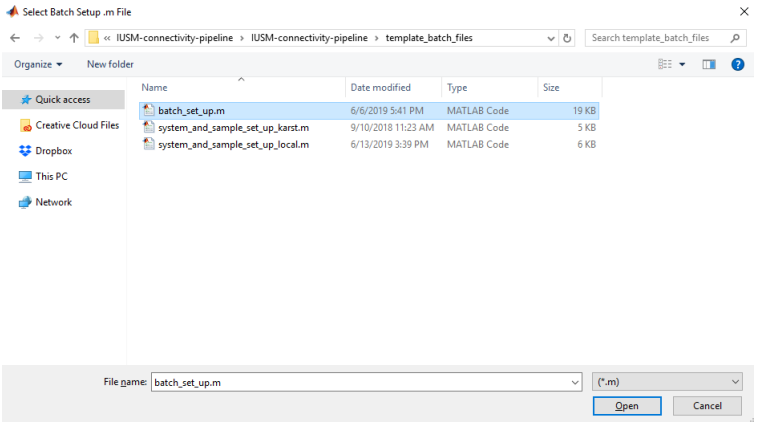

Figure 11: Select batch setup scripts

3. Connectivity pipeline will then start to run

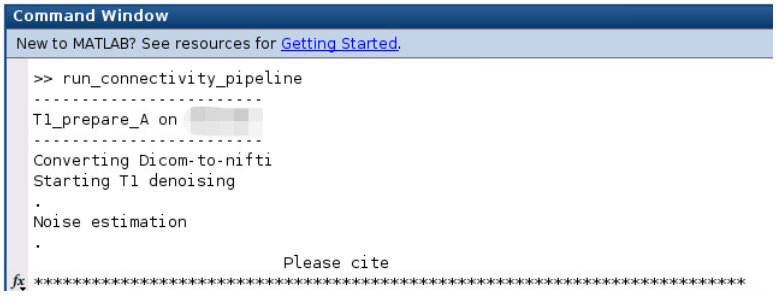

Figure 12: Select batch setup scripts

3 Notes

3.1 ICA-AROMA OR Topup + Motion Correction

ICA-AROMA (ICA-based Automatic Removalof Motion Artifacts) is an ICA-based strategy for motion artifact removal. As mentioned in Raimon et. al 2015, within the typical fMRI participant-level preprocessing stream ICA-AROMA is applied after spatial smoothing but prior to high-passfiltering and further nuisance regression (Figure 10)[2].

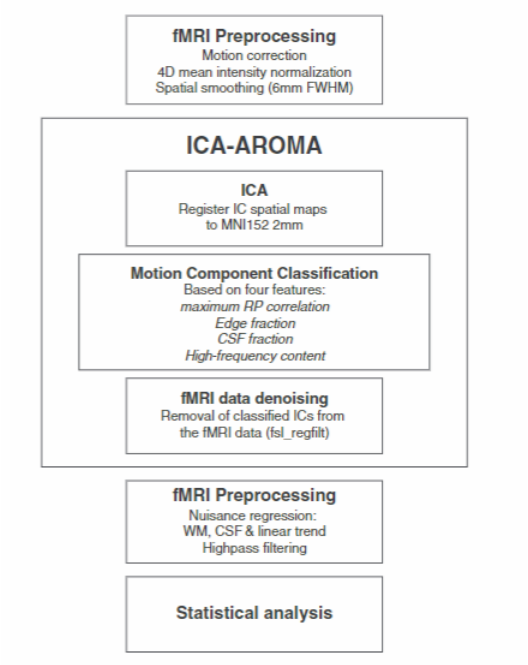

Figure 13: General workflow of ICA-AROMA [2]

Therefore, user will need to choose the motion correction method between ICA-AROMA **or** topup+correction. The related configuration can be changed in `batch_set_up.m`. If user turn on the flag of ICA-AROMA (i.e. `flags.EPI.ICA_AROMA = 1`), the pipeline will **automatically turn off** the following pre-processing steps: spin echo unwarp, top-up, slice-timing correction and motion correction.
Also, please be aware that since spatial smoothing should be done before ICA-AROMA, user should turn off the spatial smoothing after ICA-AROMA, i.e.

```
flags.EPI.SpatialSmooth = 1
     configs.EPI.fwhm = 0;
```

3.2 Running Melodic outside the connectivity pipeline

In some cases, since the extensive computation time for running FEAT/Melodic sequentially, users may want to skip running single-session Melodic inside the pipeline. Instead, user can run parallel FEAT/MELODIC outside the pipeline.

The IUSM connectivity pipeline provides option for user to skip the MELODIC.

Then go to `batch_set_up.m`, under fMRI_A setting, **turn off** `flags.EPI.ICA_AROMA` (also need to turn off `flags.EPI.useExistAROMA`). As shown below:
The pipeline will then skip the single session FEAT and use the output directory specified in `system_and_sample_set_up_local.m` to run ICA-AROMA.

```
flags.EPI.ICA_AROMA = 1; % ICA AROMA for motion correction, required single session melodic processed on the subjects
    flags.EPI.useExistAROMA = 1; % (optional) if ICA-AROMA has done
    flags.EPI.feat = 0; % single session melodic, required for ICA-AROMA, if done, this step can be skipped
        configs.EPI.featVersion = '3.15'; % version of feat, don't change unless the version of feat is changed
        configs.EPI.watcher = 1; % whether the featWatcher should be turn on
        configs.EPI.pre_fwhm = 6; % Melodic pre-processing: Full Width at Half Maximum of the Gaussian kernel
        configs.EPI.brainThres = 5; % Brain/background threshold, in percentage(%).
        configs.EPI.B0Unwarp = 0; % B0 field map unwarping
        configs.EPI.melodic_st = 0; % Slice timing correction
        configs.EPI.bgimage = 1; % Background image for higher-level stats overlays, don't change unless neccessary
                                 %1: Mean highres
                                 %2: First highres
                                 %3 : Mean functional
                                 %4 : First functional
                                 %5 : Standard space template
        configs.EPI.reghighres_search = 90; % Search space for registration to main structural, don't change unless neccessary
                                          %0: No search
                                          %90: Normal search
                                          %180: Full search
        configs.EPI.regstandard_search = 90; % Search space for registration to standard space, don't change unless neccessary
                                          %0: No search
                                          %90: Normal search
                                          %180: Full search
        configs.EPI.regstandard_dof = 12; % Degrees of Freedom for registration to standard space
        configs.EPI.regstandard_nonlinear_yn = 1; % Do nonlinear registration from structural to standard space?
        configs.EPI.regstandard_nonlinear_warpres = 10; % (mm) Control nonlinear warp field resolution
        configs.EPI.paradigm_hp = 100; % High pass filter cutoff (s)
        configs.EPI.regstandard_res = 4; % Resampling resolution
        configs.EPI.mmthresh = 0.5; % Mixture model threshold
```

Figure 14: Skip FEAT and run ICA-AROMA only. Turn off single session melodic.

3.3 Potential overflow issue when running ICA-AROMA

ICA-AROMA[2] automatically identifies which of the components are related to head motion, by using four robust and standardized features. The source code [Github Link] implicates that ICA-AROMA use `fsl_regfilt` to denoise the functional data after classification. However, when the number of motion-component reaches to a limit, it will cause a "fatal error: segmentation fault", which might due to the data structure that FSL is using.

Per our testing and experiment, the maximum number of motion component that `fsl_regfilt` can handle is roughly about **270-278**.

References

[1] Raimon HR Pruim, Maarten Mennes, Jan K Buitelaar, and Christian F Beckmann. Evaluation of ica-aroma and alternative strategies for motion artifact removal in resting state fmri. *Neuroimage*, 112:278–287, 2015.

[2] Raimon HR Pruim, Maarten Mennes, Daan van Rooij, Alberto Llera, Jan K Buitelaar, and Christian F Beckmann. Ica-aroma: a robust ica-based strategy for removing motion artifacts from fmri data. *Neuroimage*, 112:267–277, 2015.