# PSC Pipeline Manual

Zhengwu Zhang @ Rochester
Tengfei Li @ MDAnderson
Hongtu Zhu @ MD Anderson
April 22, 2017

## Preparations:

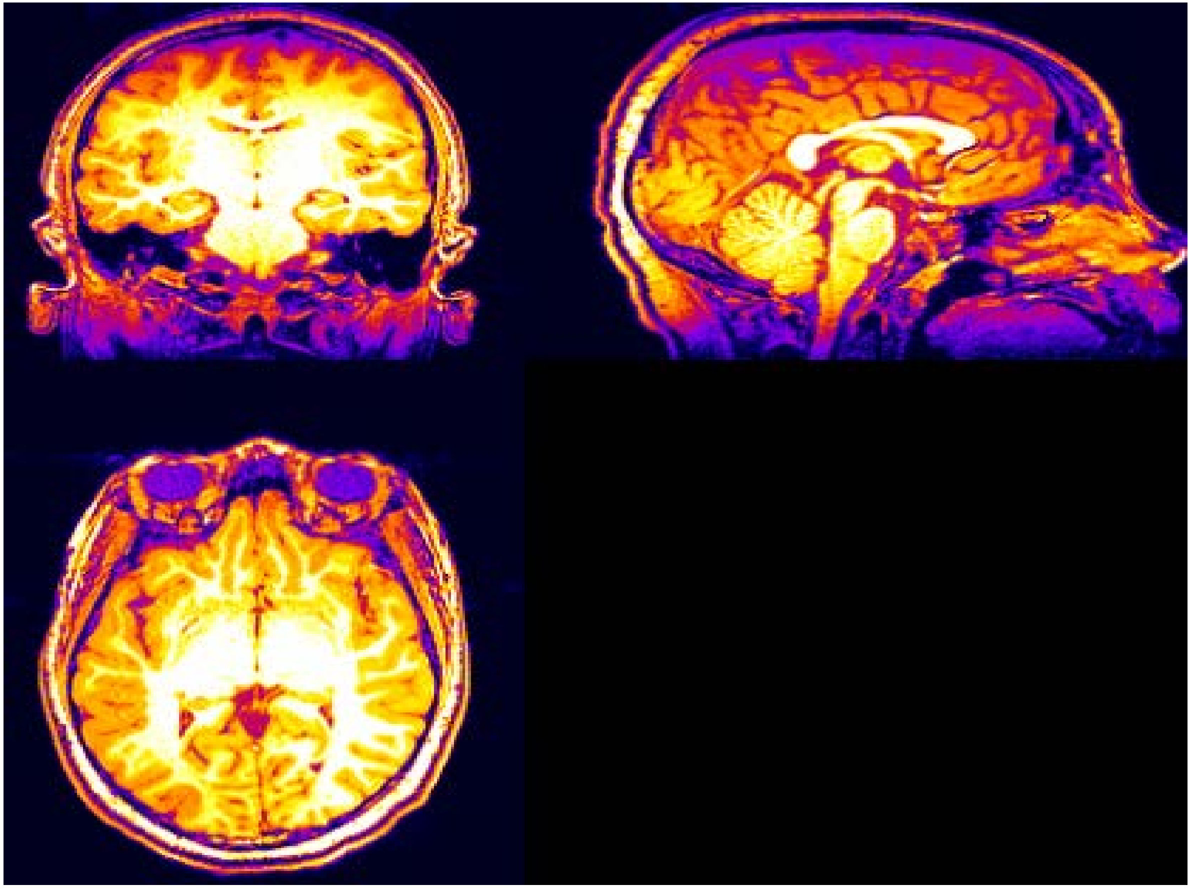
Before running this pipeline, we need some prerequisite softwares on the cluster.

1. Ask the department IT group to help install the software Mrtrix version 0.2.10, GSL, and the latest version of Freesurfer, FSL, and Ants on the cluster.
2. Login to the server Yourserver using x-win32/Putty or "ssh  Yourserver " on Mac. Install Anaconda by the following commands.
   ```
   cd /SoftwarePath
   wget https://repo.continuum.io/archive/Anaconda2-4.0.0-Linux-x86_64.sh
   sh Anaconda2-4.0.0-Linux-x86_64.sh
   ```
   Then select the installation location at /SoftwarePath
3. Download and unzip the Scilpy python package to /YourPath/. Install some prerequisite python packages by the following commands.
   ```
   setenv PATH /SoftwarePath/Anaconda2/bin:${PATH}
   cd /YourPath/Scilpy
   chmod 775 /YourPath/Scilpy/
   chmod 775 /YourPath/Scilpy/script/*.py
   pip install --user -r requirements.txt
   pip install --user -r requirements-git.txt
   module load gsl
   python setup.py build_all
   ```
4. Upload parallel scripts step1.m, step2.m, step3.m, step4.m and step5.m to /YourPath/.

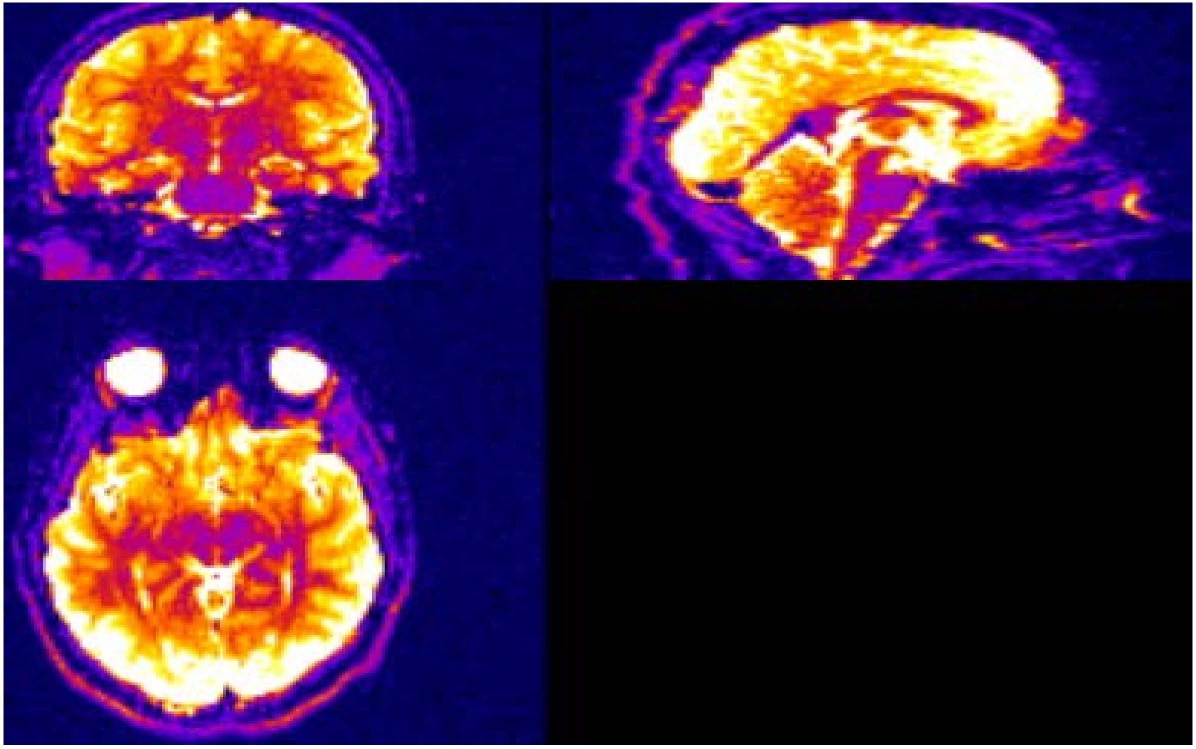
## Start the pipeline:

1. Login on the cluster using x-win32/Putty or "ssh Yourserver" on Mac. Input your username and password. Cd to your working directory, e.g.,
   ```
   cd  /YourPath/
   ```
   Copy your Imaging dataset here and organized in this way.
   YourPath/data/ID1/T1.nii.gz, YourPath/data/ID1/dwi.nii.gz
   YourPath/data/ID1/ bvecs, YourPath/data/ID1/bvals
   YourPath/data/ID2/T1.nii.gz, YourPath/data/ID2/dwi.nii.gz
   YourPath/data/ID2/ bvecs, YourPath/data/ID2/bvals
   … …
   … …
   YourPath/data/ID2/T1.nii.gz, YourPath/data/IDN/dwi.nii.gz
   YourPath/data/ID2/ bvecs, YourPath/data/IDN/bvals

T1 image



dwi0 image



Modify YourPath, your PBS headerlines and your version of softwares in step1.m. Run step1.m by

```
matlab –nojvm
step1
exit
```

in matlab and then a new folder code will be generated. It includes All_step1.sh, jobsubmission_stg1_1.pbs, jobsubmission_stg1_2.pbs, …, jobsubmission_stg1_N.pbs. Type the following commands to sumbit N parallel jobs.

```
cd YourPath/code
chmod 775 All_step1.sh
./All_step1.sh
```

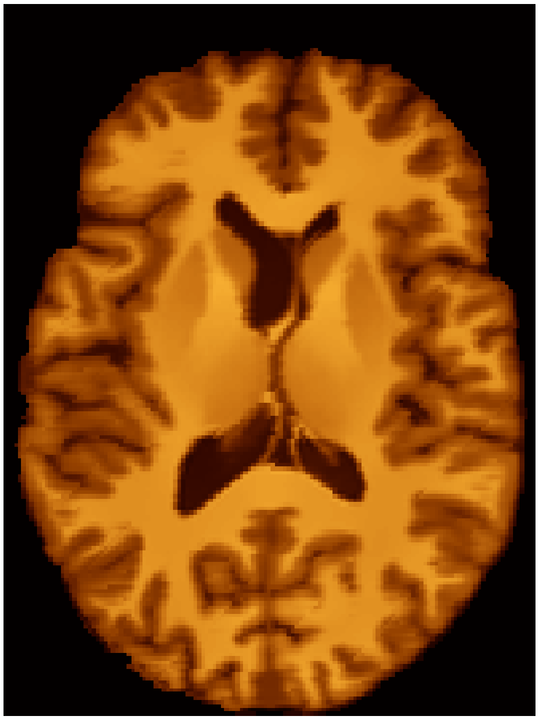
Summary of this step:

Input: dwi.nii.gz, bvals, bvecs, T1.nii.gz

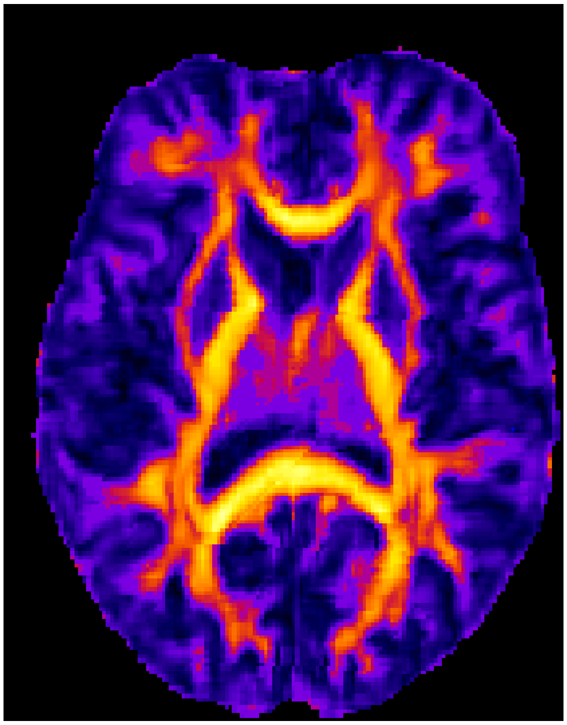
Output: diffusion/bvals_x, diffusion/bvecs_x (translated bval bvec)
diffusion/data.nii.gz (dti image after eddy correction)
diffusion/data_1x1x1.nii.gz (resampled data.nii.gz to uniform resolution)
diffusion/fa.nii.gz, diffusion/tensor.nii.gz (extracted fa and tensor images using FSL)
structural/T1_1x1x1.nii.gz (resampled T1.nii.gz to uniform resolution)
structural/T1_1x1x1_denoised.nii.gz (denoised T1_1x1x1.nii.gz)
registration/T1_dti_1.nii.gz (linearly registered T1_1x1x1_denoised.nii.gz to dti space)
T1_dti_final.nii.gz (nonlinearly registered T1 image to dti space)
t1_brain.nii.gz (extracted T1 brain of T1_dti_final.nii.gz)
nodif_brain_mask.nii.gz (brain mask of t1_brain.nii.gz)
…

Make sure the output t1_brain.nii.gz, fa.nii.gz, tensor.nii.gz and data.nii.gz are in the same space.

t1_brain



fa



```
# See the following for details of this step.
```

The file All_step1.sh is used to submit N job
jobsubmission_stg1_1.pbs, jobsubmission_stg1_2.pbs, …, jobsubmission_stg1_N.pbs
in a parallel way. jobsubmission_stg1_1.pbs reads like

```
#PBS -l nodes=1:rhel7:ppn=1 -l walltime=23:59:59,mem=16gb
#PBS -N Zhengwu_1
#PBS -o Zhengwu_1.out
#PBS -j oe
# sub job for subject 2545591
module load gsl/1.16
module load mrtrix
module load FSL
module load ANTs/2.1.0
setenv PATH /SoftwarePath/Anaconda2/bin:$PATH
setenv PATH /YourPath/Scilpy//scripts/:${PATH}
cd /YourPath/data/000001
chmod 775 /YourPath/Scilpy/PSC_Pipeline/UK_Biobank//PREPROCESS_Step1_Registration.sh
mv dwi.nii.gz data.nii.gz
mv nodif.nii.gz data.nii.gz
mv bvecs0 bvecs
mv bvals0 bvals
sh /YourPath/Scilpy/PSC_Pipeline/UK_Biobank//PREPROCESS_Step1_Registration.sh #
# -------------------------   #
```

Running this pbs file is equivalent to run
/Scilpy/PSC_Pipeline/UK_Biobank/PREPROCESS_Step1_Registration.sh
for subject 0000001. Stage1.sh do eddy correction, resampling and extraction the brain mask on the DWI image, and denoising and resampling on the T1 image; it then registraters first linearly then nonlinearly the T1 image to the dti space. It reads:

```
rm -r diffusion
 rm -r structural
 rm -r streamlines
 rm -r diffution
 rm -r connectome
 rm -r registration
mkdir registration/
mkdir structural/
mkdir diffusion/
eddy_correct AP.nii.gz data.nii.gz 0
mv AP.bval bvals
mv AP.bvec bvecs
mrconvert data.nii.gz -stride 1,2,3,4  diffusion/data.nii.gz -force
```

```
#translate bval bvec to the right space [+1 +2 +3 +4]
scil_convert_gradient_fsl_to_mrtrix.py bvals bvecs diffusion/encoding.b -f
scil_flip_grad.py --mrtrix diffusion/encoding.b diffusion/encoding_x.b x
scil_convert_gradient_mrtrix_to_fsl.py diffusion/encoding_x.b diffusion/bvals_x diffusion/bvecs_x -f
#resample t1 image into 1x1x1
mv T1.nii.gz structural/T1_1x1x1.nii.gz
#denoise the t1 image
scil_run_nlmeans.py structural/T1_1x1x1.nii.gz structural/T1_1x1x1_denoised.nii.gz 1 --noise_est basic -f
#resample the dti image into 1x1x1
scil_resample_volume.py diffusion/data.nii.gz diffusion/data_1x1x1.nii.gz --resolution 1 -f
####### initial registration using FSL
flirt -in structural/T1_1x1x1_denoised.nii.gz -ref diffusion/data_1x1x1.nii.gz -out registration/T1_dti_1.nii.gz -omat registration/T1_dti_1.mat -bins 256 -cost mutualinfo -searchrx -90 90 -searchry -90 90 -searchrz -90 90 -dof 6  -interp sinc -sincwidth 7 -sincwindow hanning
## improved the registration
# from DTI data, extract b0
bet diffusion/data_1x1x1.nii.gz mask.nii.gz -R -m -f 0.25
mv mask.nii.gz b0_brain.nii.gz
# using the same mask, extract t1 image
fslmaths registration/T1_dti_1.nii.gz -mul mask_mask.nii.gz registration/mask_dt1_t1.nii
#calculate the fa map for dti data
cd diffusion
scil_compute_dti_metrics.py data_1x1x1.nii.gz bvals_x bvecs_x --mask ../mask_mask.nii.gz -f \
--not_all --fa fa.nii.gz --tensor tensor.nii.gz
#now we have fa, do registration using fa, b0 data together
cd ..
# Use ANTS to improve the registration. This does not work on a non-masked version.
antsRegistration -d 3 -m MI[b0_brain.nii.gz,registration/mask_dt1_t1.nii.gz,1,32,Regular,0.25] \
-m MI[diffusion/fa.nii.gz,registration/mask_dt1_t1.nii.gz,1,4] \
-c [1000x500x250x0,1e-7,5] -t affine[0.1] -f 8x4x2x1 -s 4x2x1x0 -u 1 -o Antsaffine
antsApplyTransforms -d 3 -i registration/T1_dti_1.nii.gz -o T1_dti_final.nii.gz \
 -r b0_brain.nii.gz -t Antsaffine0GenericAffine.mat -n Linear
# rename the mask;
bet T1_dti_final.nii.gz t1_brain.nii.gz -m -R -B -f 0.5
mv t1_brain_mask.nii.gz nodif_brain_mask.nii.gz
```

2. Upload step2.m to /YourPath and modified YourPath, your PBS headerlines and your version of softwares in step2.m. Run step2.m by

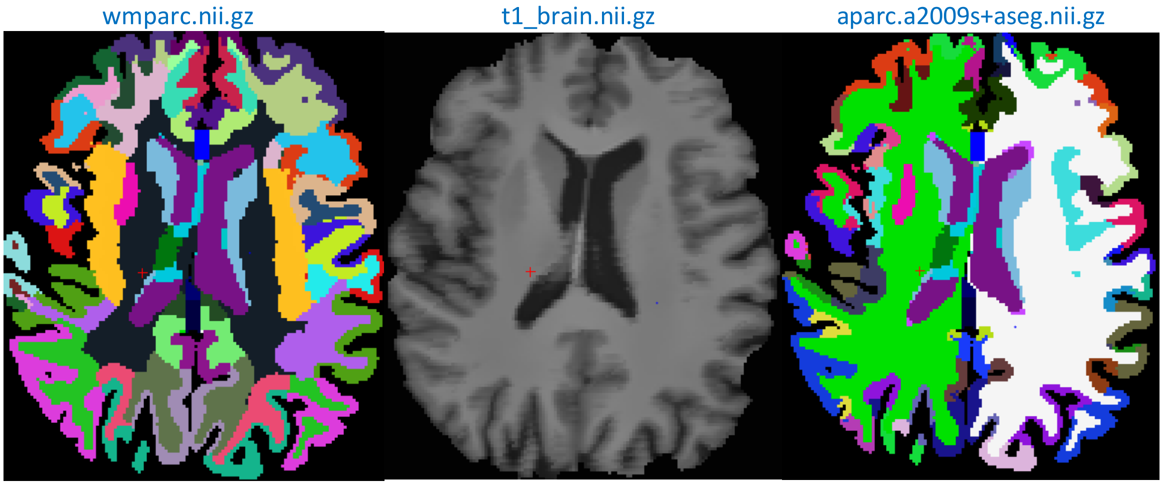
   ```
   matlab –nojvm
   step2
   exit
   ```

   to run step2.m in matlab and then a new folder code will be generated. It includes FS_batAll.sh, FS_bat1.pbs, FS_bat2.pbs, …, FS_batN.pbs. Type following commands to sumbit N parallel jobs.

   ```
   chmod 775 All_step2.sh
   ```

```
./All_step2.sh
```

Summary of this step:
Input: t1_brain_crop.nii.gz
Output: aparc.a2009s+aseg.nii.gz (freesurfer cortical segmantation and subcortical parcellation)
wmparc.nii.gz (freesurfer white matter parcellation)
/freesurfer (freesurfer recon-all pipeline output folder)



# See the following for details of this step.

The file All_step2.sh is used to submit N job FS_bat1.pbs, FS_bat2.pbs, …, FS_batN.pbs in a parallel way. FS_bat1.pbs reads like:

```
#PBS -l nodes=1:ppn=8,walltime=23:59:59,mem=40gb
#PBS -N Zhengwu_1
#PBS –o Zhengwu_1.out
#PBS –j oe
module load freesurfer
#!/bin/bash
setenv FREESURFER_HOME /SoftwarePath/freesurfer/5.3.0/freesurfer
source $FREESURFER_HOME/SetUpFreeSurfer.sh
setenv SUBJECTS_DIR /YourPath/data/000001
cd /YourPath/data/000001
rm -r -f pnc000001
recon-all -openmp 8 -subjid pnc000001 -i T1_dti_final.nii.gz -all
rm -r -f pnc000001 /mri/transforms/
rm -r -f pnc000001 /surf/
```

This step processes the parcellation of the registered T1 brain t1_brain.nii.gz.
Summary of this step:
Input: T1_dti_final.nii.gz
Output: pnc0000001/mri/rawavg.nii.gz (created mgz file in the native space of T1_dti_final.nii.gz)
pnc0000001/mri/brainmask.mgz (freesurfer brain mask output)

pnc0000001/mri/aparc.a2009s+aseg.mgz (freesurfer parcellation by desikan atlas)
pnc0000001/mri/wmparc.nii.gz (freesurfer parcellation by destrieux atlas)

3. Upload step3.m to /YourPath and modified YourPath, your PBS headerlines and your version of softwares in step3.m. Run step3.m by

```
matlab –nojvm
step3
exit
```

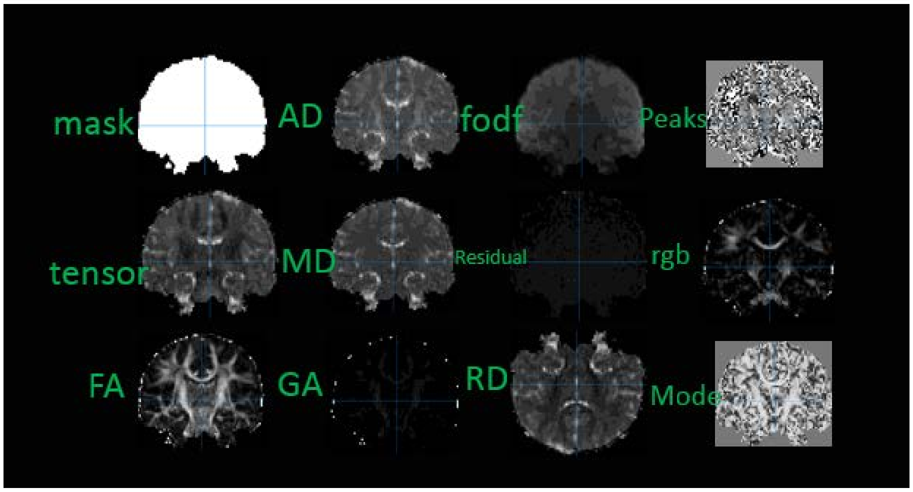
to run step3.m in matlab and then a new folder code will be generated. It includes All_step3.sh, jobsubmission_stg3_1.pbs, jobsubmission_stg3_2.pbs, …, jobsubmission_stg3_N.pbs. Type following commands to sumbit N parallel jobs.

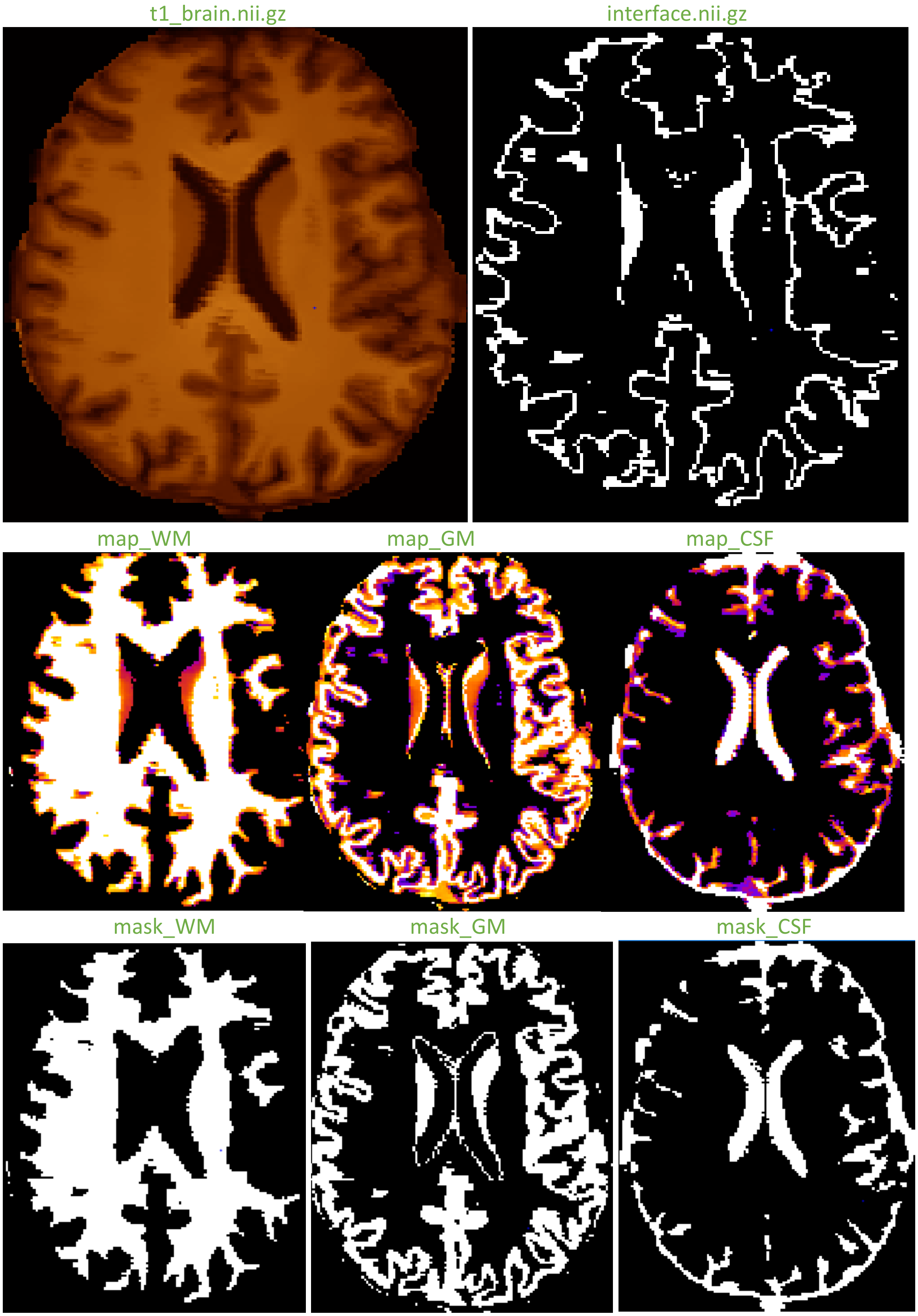
```
chmod 775 All_step3.sh
./All_step3.sh
```

Summary of this step:

Input: pnc0000001/mri/rawavg.nii.gz, pnc0000001/mri/brainmask.mgz, nodif_brain_mask.nii.gz pnc0000001/mri/aparc.a2009s+aseg.mgz, pnc0000001/mri/wmparc.nii.gz, T1_dti_final.nii.gz data_1x1x1.nii.gz, bvals_x, bvecs_x

Output: ad.nii.gz, dti_residual.nii.gz, fa.nii.gz, fodf.nii.gz,ga.nii.gz, md.nii.gz, mode.nii.gz,rd.nii.gz peaks.nii.gz, rgb.nii.gz, tensor.nii.gz, tensor_norm.nii.gz, tensor_evals_e1.nii.gz, tensor_evals_e2.nii.gz, tensor_evals_e3.nii.gz, tensor_evals.nii.gz, tensor_evecs.nii.gz, tensor_evecs_v1.nii.gz, tensor_evecs_v2.nii.gz, tensor_evecs_v3.nii.gz, physically_implausible_signals_mask.nii.gz, pulsation_and_misalignment_std_b0.nii.gz pulsation_and_misalignment_std_dwi.nii.gz, fodf_frf.txt (all dti and odf metrics)
t1_brain.nii.gz (extracted T1 brain)
t1_brain_mixeltype.nii.gz (represents the classification of voxel's tissue mixture)
t1_brain_seg.nii.gz (t1_brain.nii.gz brain segmentation)
t1_brain_pveseg.nii.gz (t1_brain.nii.gz pve <partial volume effect> brain segmentation)
mask_gm.nii.gz, mask_wm.nii.gz, mask_csf.nii.gz (segmentation mask of t1_brain.nii.gz)
map_gm.nii.gz, map_wm.nii.gz, map_csf.nii.gz (pve segmentation mask of t1_brain.nii.gz)
interface.nii.gz, interface_count.txt (segmentation interface file of t1_brain.nii.gz)
map_include.nii.gz, map_exclude.nii.gz (other output by FSL FAST segmentation pipeline)




t1_brain.nii.gz
interface.nii.gz
map_WM
map_GM
map_CSF
mask_WM
mask_GM
mask_CSF

```
    # See the following for details of this step.
```

The file All_step3.sh is used to submit N job jobsubmission_stg3_1.pbs, jobsubmission_stg3_2.pbs, …, jobsubmission_stg3_N.pbs in a parallel way. jobsubmission_stg3_1.pbs reads like

```
#PBS -l nodes=1:rhel7:ppn=4 -l walltime=23:59:59,mem=32gb
#PBS -N Zhengwu_0000001
#PBS -o Zhengwu_0000001.out
#PBS -j oe
# sub job for subject 0000001
module load gsl/1.16
module load mrtrix
module load FSL
module load ANTs/2.1.0
module load freesurfer
setenv PATH /SoftwarePath/Anaconda2/bin:$PATH
setenv PATH /YourPath /Scilpy//scripts/:${PATH}
setenv PYTHONPATH /YourPath/Scilpy/
cd /YourPath /data/0000001
cp pnc0000001/mri/aparc.a2009s+aseg.mgz ./
cp pnc0000001/mri/wmparc.mgz ./
cp pnc0000001/mri/rawavg.mgz ./
cp pnc0000001/mri/brainmask.mgz ./
chmod 775 /YourPath/Scilpy/PSC_Pipeline/UK_Biobank/TRACTOGRAPHY_Step1.sh
sh /YourPath/Scilpy/PSC_Pipeline/UK_Biobank//TRACTOGRAPHY_Step1.sh
```

This step denoises the data_1x1x1.nii.gz, computes the dti metrics and odf metrics, and runs the FSL FAST pipeline to generate WM, GM, CSF segmentation-related results. Running this pbs file is equivalent to run /YourPath/Scilpy/PSC_Pipeline/UK_Biobank/TRACTOGRAPHY_Step1.sh for subject 0000001. This step will generate the fibertracking result. /YourPath/Scilpy/PSC_Pipeline/UK_Biobank/TRACTOGRAPHY_Step1.sh reads like:

```
    # See the following for details of this step.
    mri_convert -rl rawavg.mgz -rt nearest wmparc.mgz wmparc_in_rawavg.mgz
    mri_convert -rl rawavg.mgz -rt nearest aparc.a2009s+aseg.mgz
aparc.a2009s+aseg_in_rawavg.mgz
    # improve the brain segmentation (from the freesufer result)
    mrconvert brainmask.mgz brainmask.nii.gz -force
    mrconvert brainmask.nii.gz -stride 1,2,3 brainmask.nii.gz -force
    mrtransform -template T1_dti_final.nii.gz -interp linear -datatype int32 brainmask.nii.gz
mr_brainmask.nii.gz -force
    fslmaths mr_brainmask.nii.gz -bin t1_brain_mask.nii.gz
    fslmaths t1_brain_mask.nii.gz -add nodif_brain_mask.nii.gz merged_brain_mask.nii.gz
    fslmaths merged_brain_mask.nii.gz -bin merged_brain_mask.nii.gz
    mv merged_brain_mask.nii.gz nodif_brain_mask.nii.gz
    mv T1_dti_final.nii.gz structural/T1_dti_final.nii.gz
    fslmaths structural/T1_dti_final.nii.gz -mul nodif_brain_mask.nii.gz structural/t1_brain.nii.gz
    cd diffusion
```

```
mrconvert ../nodif_brain_mask.nii.gz -stride 1,2,3 mask.nii.gz -force
######## compute dti metrics
#scil_compute_dti_metrics.py data_1x1x1.nii.gz bvals_x bvecs_x --mask mask_crop.nii.gz -f
scil_compute_dti_metrics.py data_1x1x1.nii.gz bvals_x bvecs_x --mask mask.nii.gz -f
########## computer odf metric, denoising the data
# this step takes about 1~1.5 hours
scil_run_nlmeans.py data_1x1x1.nii.gz dwi_rnlm.nii.gz 1 \
  --mask mask.nii.gz --noise_est basic --processes 4 -f
cd ..

#segment
cd structural
fast -t 1 -n 3 -H 0.1 -I 6 -l 20.0 -g -o t1_brain.nii.gz t1_brain.nii.gz
mv t1_brain_seg_2.nii.gz mask_wm.nii.gz
mv t1_brain_seg_1.nii.gz mask_gm.nii.gz
mv t1_brain_seg_0.nii.gz mask_csf.nii.gz
mv t1_brain_pve_2.nii.gz map_wm.nii.gz
mv t1_brain_pve_1.nii.gz map_gm.nii.gz
mv t1_brain_pve_0.nii.gz map_csf.nii.gz

#compute_pft_maps.py map_wm.nii.gz map_gm.nii.gz map_csf.nii.gz -f
scil_compute_maps_for_particle_filter_tracking.py map_wm.nii.gz map_gm.nii.gz map_csf.nii.gz -f
scil_count_non_zero_voxels.py interface.nii.gz -o interface_count.txt
############### processe the label data
mrconvert ../wmparc_in_rawavg.mgz ../wmparc_in_rawavg.nii.gz -force
mrconvert ../wmparc_in_rawavg.nii.gz -stride 1,2,3 wmparc.nii.gz -force
mrconvert ../aparc.a2009s+aseg_in_rawavg.mgz ../aparc.a2009s+aseg_in_rawavg.nii.gz -force
mrconvert ../aparc.a2009s+aseg_in_rawavg.nii.gz -stride 1,2,3 aparc.a2009s+aseg.nii.gz -force
cd ..
############### calculate fodf
cd diffusion
scil_compute_fodf.py dwi_rnlm.nii.gz bvals_x bvecs_x --sh_order 6 --mask mask.nii.gz --mask_wm ../structural/mask_wm.nii.gz --processes 4 -f --frf 15,3,3 --not_all --fodf fodf.nii.gz --peaks peaks.nii.gz
# clean large files that we don't need
rm data.nii.gz # keep dwi_all.nii.gz
rm data_1x1x1.nii.gz
rm dwi_rnlm.nii.gz
cd ..
```

4. Upload step4.m to /YourPath and modified YourPath, your PBS headerlines and your version of softwares in step4.m. Run step4.m by

   ```
   matlab –nojvm
   step4
   exit
   ```

   to run step4.m in matlab and then a new folder code will be generated. It includes All_step4.sh, jobsubmission_stg4_1.pbs, jobsubmission_stg4_2.pbs, …, jobsubmission_stg4_N.pbs. Type following commands to sumbit N parallel jobs.

```
chmod 775 All_step4.sh
./All_step4.sh
```

Summary of this step:

Input: all output in step 3.

Output: streamlines/full_interface_prob_pft_invcoord.trk (fibertracking results)

# See the following for details of this step.

The file All_step4.sh is used to submit N jobs in a parallel way:

jobsubmission_stg4_1.pbs, jobsubmission_stg4_2.pbs, …, jobsubmission_stg4_N.pbs

jobsubmission_stg4_1.pbs reads like:

```
#PBS -l nodes=1:rhel7:ppn=10 -l walltime=71:59:59,mem=80gb
#PBS -N Zhengwu_0000001
#PBS -o Zhengwu_0000001.out
#PBS -j oe
# sub job for subject 0000001
module load gsl/1.16
module load mrtrix
module load FSL
module load ANTs/2.1.0
module load freesurfer
setenv PATH /SoftwarePath/Anaconda2/bin:$PATH
setenv PATH /YourPath /Scilpy//scripts/:${PATH}
setenv PYTHONPATH /YourPath/Scilpy/
cd /YourPath /data/0000001
chmod 775  /YourPath/Scilpy/PSC_Pipeline/UK_Biobank /TRACTOGRAPHY_Step2.sh
sh /YourPath/Scilpy/PSC_Pipeline/UK_Biobank /TRACTOGRAPHY_Step2.sh
```

Running this pbs file is equivalent to run /YourPath/Scilpy/PSC_Pipeline/UK_Biobank /TRACTOGRAPHY_Step2.sh for subject 0000001. This step will generate the fibertracking result.

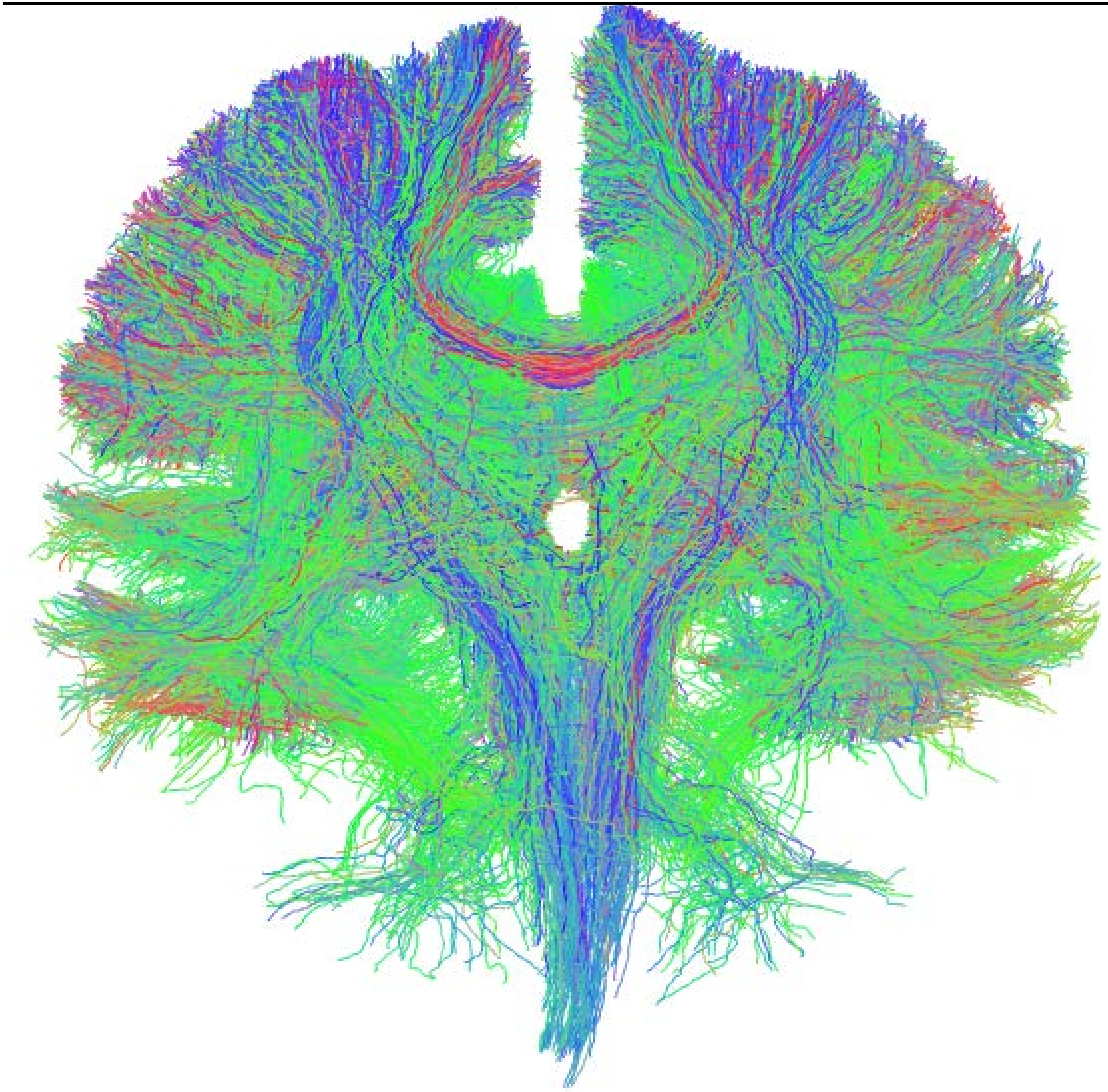
/YourPath/Scilpy/PSC_Pipeline/UK_Biobank /TRACTOGRAPHY_Step2.sh reads like:

```
#fiber tracking
  mkdir streamlines
  scil_compute_particle_filter_tracking.py --algo 'prob' --npv 10 diffusion/fodf.nii.gz \
    structural/interface.nii.gz structural/map_include.nii.gz structural/map_exclude.nii.gz \
    streamlines/full_interface_prob_pft.trk --processes 10 -f

  # remove invalid streamlines
  scil_remove_invalid_coordinates_from_streamlines.py --gnc --fnc \
  streamlines/full_interface_prob_pft.trk structural/t1_brain.nii.gz
  streamlines/full_interface_prob_pft_invcoord.trk -f

  # remove unnecessary files
  cd streamlines
  rm full_interface_prob_pft.trk
```

```
cd ..
mkdir connectome
```



5. Upload step5.m to /YourPath and modified YourPath in step5.m. Run step5.m by

   ```
   matlab –nojvm
   step5
   exit
   ```

   to run step5.m in matlab and then a new folder code will be generated. It includes All_step5.sh, stage5_connectomeanalysis_pipeline.sh, jobsubmission_stg5_1.pbs, jobsubmission_stg5_2.pbs, …, jobsubmission_stg5_N.pbs. Type following commands to sumbit N parallel jobs.

   ```
   Chmod 775 All_step4.sh
   ./All_step5.sh
   ```

   Summary of this step:

   Input: full_interface_prob_pft_invcoord.trk, output of Step 3.

   Output: UKBB_desikan_cm_streamlines.mat, UKBB_destrieux_cm_streamlines.mat

UKBB_desikan_partbrain_subcort_cm_streamlines.mat,
UKBB_destrieux_partbrain_subcort_cm_streamlines.mat
(fibertracking matrix)
desikan_cm_processed_famax_100.mat
desikan_cm_processed_famean_100.mat
desikan_cm_processed_mdmax_100.mat
desikan_cm_processed_mdmean_100.mat
desikan_cm_processed_sfa_100.mat
desikan_cm_processed_smd_100.mat
desikan_cm_processed_volumn_100.mat
desikan_cm_processed_volumn_ratio_100.mat
desikan_partbrain_cm_processed_famax_100.mat
desikan_partbrain_cm_processed_famean_100.mat
desikan_partbrain_cm_processed_mdmax_100.mat
desikan_partbrain_cm_processed_mdmean_100.mat
desikan_partbrain_cm_processed_sfa_100.mat
desikan_partbrain_cm_processed_smd_100.mat
desikan_partbrain_cm_processed_volumn_100.mat
desikan_partbrain_cm_processed_volumn_ratio_100.mat
desikan_partbrain_cm_processed_fiberlen100.mat
destrieux_cm_processed_famax_100.mat
destrieux_cm_processed_famean_100.mat
destrieux_cm_processed_mdmax_100.mat
destrieux_cm_processed_mdmean_100.mat
destrieux_cm_processed_sfa_100.mat
destrieux_cm_processed_smd_100.mat
destrieux_cm_processed_volumn_100.mat
destrieux_cm_processed_volumn_ratio_100.mat
destrieux_partbrain_cm_processed_famax_100.mat
destrieux_partbrain_cm_processed_famean_100.mat
destrieux_partbrain_cm_processed_mdmax_100.mat
destrieux_partbrain_cm_processed_mdmean_100.mat
destrieux_partbrain_cm_processed_sfa_100.mat
destrieux_partbrain_cm_processed_smd_100.mat
destrieux_partbrain_cm_processed_volumn_100.mat
destrieux_partbrain_cm_processed_volumn_ratio_100.mat
destrieux_partbrain_cm_processed_fiberlen100.mat
(features generated)

# See the following for details of this step.

The file All_step5.sh is used to submit N jobs in a parallel way:
jobsubmission_stg5_1.pbs, jobsubmission_stg5_2.pbs, …, jobsubmission_stg5_N.pbs.
And jobsubmission_stg4_1.pbs reads like:

```
#PBS -l nodes=1:rhel7:ppn=10 -l walltime=23:59:59,mem=80gb
#PBS -N Zhengwu_0000001
#PBS -o Zhengwu_0000001.out
#PBS -j oe
# sub job for subject 0000001
```

```
module load gsl/1.16
module load mrtrix
module load FSL
module load ANTs/2.1.0
module load freesurfer
setenv PATH /SoftwarePath/Anaconda2/bin:$PATH
setenv PATH /YourPath /Scilpy//scripts/:${PATH}
setenv PYTHONPATH /YourPath/Scilpy/
cd /YourPath /data/0000001
chmod 775  /YourPath/Scilpy/PSC_Pipeline/UK_Biobank/Connectome_Step1.sh
sh  /YourPath/Scilpy/PSC_Pipeline/UK_Biobank/Connectome_Step1.sh
# -------------------------   #
```

Running this pbs file is equivalent to run /YourPath/Scilpy/PSC_Pipeline/UK_Biobank/Connectome_Step1.sh for subject 0000001. This step will use matlab codes to generate features and summary statistics for subject 0000001.  Connectome_Step1.sh reads like:

```
cd connectome
#desikan atlas
# extract connectivity matrices, get the dilation of images for Desikan
extraction_sccm_withfeatures_cortical.py ../streamlines/full_interface_prob_pft_invcoord.trk ../diff
usion/fa.nii.gz ../diffusion/md.nii.gz ../structural/wmparc.nii.gz
$SCIP_PATH/scilpy/connectome/Desikan_ROI.txt
$SCIP_PATH/scilpy/connectome/FreeSurferColorLUT.txt UKBB 20 240 1 2 4 desikan
extraction_sccm_withfeatures_subcortical.py ../streamlines/full_interface_prob_pft_invcoord.trk ../
diffusion/fa.nii.gz ../diffusion/md.nii.gz ../structural/wmparc.nii.gz
UKBB_desikan_dilated_labels.nii.gz  $SCIP_PATH/scilpy/connectome/Subcortical_ROI.txt
$SCIP_PATH/scilpy/connectome/FreeSurferColorLUT.txt UKBB 20 240 4 0 desikan


# Destreoux
extraction_sccm_withfeatures_cortical.py ../streamlines/full_interface_prob_pft_invcoord.trk ../diff
usion/fa.nii.gz ../diffusion/md.nii.gz ../structural/aparc.a2009s+aseg.nii.gz
$SCIP_PATH/scilpy/connectome/Destrieux_ROI.txt
$SCIP_PATH/scilpy/connectome/FreeSurferColorLUT.txt UKBB 20 240 1 2 4 destrieux
extraction_sccm_withfeatures_subcortical.py ../streamlines/full_interface_prob_pft_invcoord.trk ../
diffusion/fa.nii.gz ../diffusion/md.nii.gz ../structural/aparc.a2009s+aseg.nii.gz
UKBB_destrieux_dilated_labels.nii.gz  $SCIP_PATH/scilpy/connectome/Subcortical_ROI.txt
$SCIP_PATH/scilpy/connectome/FreeSurferColorLUT.txt UKBB 20 240 4 0 destrieux
```