Instruction of batch files

Examples are HCP data.

P reprocessed structural data of HCP could be download from <https://db.humanconnectome.org/app/template/Login.vm> (eg., 100307\_3T\_Structural\_preproc\_extended). So we could directly follows pipeline from Step2. Step1 only use ‘recon-all’ command to processed raw T1 data to get same files as preprocessed-HCP data. Actually, preprocessing steps are not only contain ‘recon-all’ but also other complex commands (contains PreFreesurfer pipeline and Freesurfer pipeline, details see ref. https://www.humanconnectome.org/storage/app/media/documentation/s1200/HCP\_S1200\_Release\_Reference\_Manual.pdf ).

And all script of HCP processing pipeline could be download from: <https://github.com/Washington-University/HCPpipelines>.

For simplicity, we use ‘recon-all’ directly when the original T1 data is of high-quality.

**Files introduction:**

List of batch:

freesurfer\_batch.sh; dsiBatch.sh; matlabBatch.sh;

Subjects list: file\_list\_HCP\_all\_subset.txt;

Tutorial: tutorial\_batch;

List of all subcortical and cortical regions: RegionInfo.txt;

**Detailed description:**

***Step1: processing raw structural data***

Install freesurfer (<https://surfer.nmr.mgh.harvard.edu/fswiki/DownloadAndInstall>)

settings:

export FREESURFER\_HOME=/usr/local/freesurfer

source $FREESURFER\_HOME/SetUpFreeSurfer.sh

export SUBJECTS\_DIR=…/../N0001 % setting individual structural data

for one subject:

recon-all -i $SUBJECTS\_DIR/T1.nii.gz -s bert -all

batch script naming **freesurfer\_batch.sh** shows multi-subject pipeline for recon-all.

file\_list\_BSD\_all\_subset.txt lists all subject name.

eg. N0001, N0002, N0003…

SUBJECTS\_DIR is set as the path to individual subject. In our example, raw data store as:

.../BSDdata/T1/N0001/T1.nii.gz

.../BSDdata/T1/N0002/T1.nii.gz

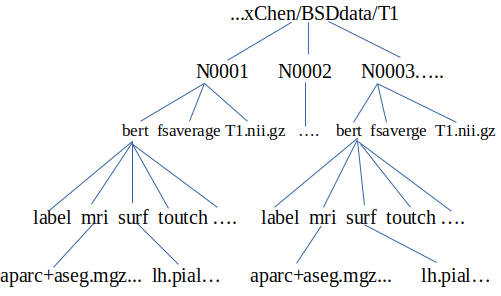
.../BSDdata/T1/N0003/T1.nii.gz

…………

No other raw data except T1.nii.gz and file\_list\_BSD\_all\_subset.txt.

Note: After recon-all，two folders: ‘bert’ and ‘fsaverage’ will be produced under subject data；important subfolders, such as label, mri, surf are under ‘bert’；All subfolders of ‘bert’ will be used later, eg., get aparc+aparc.nii and lh(rh).pial files.

Root trees is like:



***Step2: get aparc+aseg.nii files.***

convert .../mri/aparc+aseg.mgz to .../mri/aparc+aseg.nii which will used as T1 reference when perform tracking steps. The detailed commands are also shown in freesurfer\_batch.sh.

Note: Preprocessed HCP data also doesn’t contain nifti file. So using the same command (i.e., mri\_convert) to get individual aparc+aseg.nii file and copy that to individual folder naming such as ’100307/mri’, ’103414/mri’, ‘105115/mri’,…

***Step3: extract 68-ROI labels from aparc.annote.ctab based on surface file and save them under like ‘../100307/ROI’ folder.***

Two types of label:

type1: based on lh(rh).pial.surf.gii file to get label coordinates;

type2: based on lh(rh).pial file to get label coordinates;

Commands are also shown in **freesurfer\_batch.sh**. Copy all surface files to individual folder naming such as ‘.../100307/surf’ folder.

Note: for HCP data, both of two types of surface files are exist. However, the structural data processed only by ‘recon-all’ doesn’t include pial.surf.gii files. So in the following matlab scripts, we define two types of program according to surface format. Although both of files are surface, their 3D coordinates are quite different. lh(rh).pial is using surface RAS (that is tksurfer) coordinate system. While lh(rh).pial.surf.gii is based on structure files, its surface coordinates is more or less same as the scanner RAS coordinates of aparc+aseg.nii.

***Step4: get tracking results using dsi-studio.***

using command line of DSI studio to perform tracking. Reference see: [http://dsi-studio.labsolver.org/Manual/command-line-for-dsi-studio#TOC-Fiber-tracking](http://dsi-studio.labsolver.org/Manual/command-line-for-dsi-studio" \l "TOC-Fiber-tracking).

Our path to data is set as: /home/campus.ncl.ac.uk/b7071887/Xchen/Demo\_high\_resolution

Our dsi\_studio software is installed: /home/campus.ncl.ac.uk/b7071887/Xchen/dsistudio

Change paths to your own data and dsi\_studio, run ***dsiBatch.sh***.

Note: command line method could not save 10000,000-streamline files because of out of memory, however, GUI could.

So using command line method, we divide 10000,000 streamlines into 10 parts (in script: --fiber\_count=1000,000). And turn random seeding on so that every repeated round is different (in script: --random\_seed=1).

***Step4: get high-resolution matrix and map to MNI space.***

All toolbox packages used can be download online:

gifti: <https://github.com/nno/matlab_GIfTI>

along-tract-stats: <https://github.com/johncolby/along-tract-stats>

iso2mesh: <https://github.com/fangq/iso2mesh>

fieldtrip: <https://github.com/fieldtrip/fieldtrip>

surfstat: <https://github.com/blachniet/SurfStat>

Change paths to data and parameter settings in **matlabBatch.sh**, you can get final high-resolution data, including binary high-resolution matrix (‘adj\_local’ (summarized with local connectivity), ‘adj\_remote\_bin’ (only consider existing fiber) ), weighted high-resolution matrix (weigh is streamline counts, ‘adj\_remote\_wei’), high-resolution length matrix (length is Euclidean distance of average fibers between node pairs, ‘adj\_remote\_len’), low-resolution matrix (‘lo\_adj\_wei’, ’lo\_adj\_len ’), all triangle coordinates of cortical regions (‘nvl’,’nvr’) and their attached DK labels (‘faceROIidL’, ‘faceROIidR’, write in ‘filenames’) , all node coordinates of subcortical regions (‘subCoor’) and their labels(‘subROIid’, write in ‘subfilenames’) .

Note: Our matlab function could choose two surface format (pial or pial.surf.gii), and choose if downsample surface triangles, subcortical nodes and its corresponding sample rate. All regions and their order are shown in RegionInfo.txt (number is freesurfer color index).

function batch\_process(***pathToFile, subjects, type, downsample, rate***)

description:

***pathToFile***: path to file that list all subjects and matlab scripts;

***subjects***: subject name;

***type***: two format of surface data.

1-----surface is lh(rh).pial.gii,

2 ----surface is lh(rh).pial.

***downsample***:

‘yes’----- downsample surface triangles and subcortical coordinates

‘no’ ------ don’t perform downsample (**default**)

***rate***: corresponding downsample rate, **default value is 0.1.**

The method of mapping coordinate to MNI space is **linear** and its transformation matrix is stored in subject/bert/mri/transforms/talairach.xfm (got by recon-all). In order to processing by batch files, we copy this file to the path: eg. HCP\_100307/mri/talairach.xfm.

**Summarize:**

dsiBatch.sh and matlabBatch.sh could be combine together if dti data (dti, bvecs, bvals), aparc+aseg.nii, talairach.xfm, all DK labels, and surface files in native space are provided in advance.

**List of functions in batch\_process and resulting variables:**

functions:

1. [] = **conversion**(); % convert streamline endpoints into surface space and record all fibers whose two extreme points ends within cortical regions or subcortical structures;

2. [] = **loadLabels**(); % load downsample or no-downsample surface nodes and subcortical nodes.

3. [] = **makeEdgeList**(); % make edges between node pairs according to remote fiber endpoints. Besides, create a local connection with neighborhood triangles for surface nodes. While, subcortical nodes not.

4. [] = **getmatrices**(); % create high-resolution and low-resolution matrices.

5. [] = **getMNIcoor**(); % transform all nodes coordinates into MNI305 or MNI152 space.

Datasets and variables:

1. **trsfmTrk**.mat

three variables are in trkmTrk.mat:

**trkEP**: coordinates in surface space. The first three columns are start point coordinates of streamlines and the last three are endpont coordinates of streamlines;

**trk\_len**: the length of all streamlines

**trk\_type**: the first two columns are label color index (freesurfer color table), for example, 2007 represents lh.fusiform. the third is a flag, if 1, there is connection between two regions.

2. **labelSRF.mat**

14 variables are in this mat:

**filenames**: 68 cortical region names;

**subfilenames**: 17 subcortical region names including brain-stem;

**glpvertex** and **grpvertex**: 3D coordinates of all triangles for left and right hemisphere ;

**glpfaces** and **grpfaces**: the index in glpvertex(or grpvertex) for each angle of triangles;

**faceROIidL** and **faceROIidR**: the corresponding cortical region label of each triangle;

**nfl** and **nfr**: similar with glffaces and grpfaces, but are downsampled data;

**nvl** and **nvr:** similar with glpvertex and grpvertex, but are downsampled 3D coordinate.

**SubCoor**: 3D subcortical coordinates (or downsampled coordinates) in surface space;

**subROIid:** the corresponding subcortical region label of SubCoor.

3. **edgeList.mat**

five variables include:

**edgeListLocal** records two column node indices that have local connection at each row (only for cortical triangles, subcortical nodes have no local connections).

**edgeListRemote** is a five column matrix, first two columns records node indices that have fibers connected at each row; the third column is distance bias when denote fiber start point to a cortical /subcortical node; the fourth column is distance bias when denote fiber endpoint to a cortical /subcortical node; the five column is the fiber index, there are 1000000 streamlines.

**lpcentroids/rpcentroids**: is center coordinates of surface triangles.

**SubCoor:** is subcortical coodinates.

4. **matrices.mat**

nine variables:

Note: matrix is arranged as: left cortical nodes, right cortical nodes and subcortical structures. Within subcortical part, regions are also list as left and right; the final node is brain-stem.

Subcortical: lh/rh.thalamus, lh/rh.caudate, lh/rh.putamen, lh/rh.pallidum, lh/rh.amgdala, lh/rh.hippocampus, lh/rh.accumbens, lh/rh.cerebellum, brain-stem

**adj\_remote\_bin**: binary matrix. If there is a fiber connected, edge is set to 1.

**adj\_remote\_wei**: weighted matrix. weight is the number of streamlines

**adj\_remote\_len**: weighted matrix. weight is the average length of streamlines

**adj\_local**:local connectivity matrix, only show local connection near triangles. Binary.

**adj\_matrix**: sum of adj\_local and adj\_remote\_bin, also a binary matrix

**faceROI\_all**: region label of each node.

**faceROI\_cortical**: region label of each cortical node.

**lo\_adj\_wei**: low\_resolution matrix. weight is the number of streamlines. There are 85 regions.

**lo\_adj\_cortical\_wei**: low\_resolution matrix only consider cortical regions. Weight is the number of streamlines. 68 regions.