DTI Analysis Pipeline (DAP)

User Manual

Hagai Tzafrir

Introduction

DTI Analysis Pipeline (DAP) is a pipeline for processing diffusion tensor images (DTI). The pipeline was developed to analyze statistical properties of individual subjects, as well as compare between different groups. DAP provides automated end-to-end procedure to setup, pre-process, analyze and post-process large groups of data. The pipeline provides characteristics of major tracts for each individual and group-wise, as well as their statistical properties. The results are given both quantitatively and graphically, and can be easily used for further analysis using either DAP or other softwares.

This pipeline was developed for the Neuro-Imaging Center.

Contents

1

General Information

Pipeline Overview

DAP is a pipeline for analyzing and comparing fiber tracts between individuals and different groups.

- A software pipeline based on Ubuntu linux platform.
- Pipeline for diffusion tensor acquired images.
- Analysis of major fiber tracts.
- Comparison between individuals and groups.
- Visual and numerical outputs.

Software Reference

This software utilizes multiple available modules to operate.

- MATLAB.
- spm.
- Vistasoft.

- AFQ.

Other relevant references for this project

- DAP.
- Neuro-Imaging Center.

Authorized Use Permission

In order to use this pipeline an authorization must be granted by *Dr.Tzipi Horowitz-Kraus* (`Tzipi.Kraus@technion.ac.il`). In addition, usage of DAP must be done with accordance of the terms in each of the modules mentioned above.

Contact Information

The list of contacts for further information and/or troubleshooting:

Hagai Tzafrir - *Author* - `HagayT@hotmail.com`

Dr.Rola Farah - *Contributer* - `rola.farah2@gmail.com`

Dr.Tzipi Horowitz-Kraus - *Lab Prof.* - `Tzipi.Kraus@technion.ac.il`

2

Software Specifications

DAP is a pipeline that processes each subject individually and then calculate the groups statistics, finishing with a comparison between different groups is done. DAP utilizes different available modules (described in previous chapter) to create a pipeline that consists of four core stages (Figure 2.1):

1. Data setup.
2. DTI process.
3. Fiber analysis.
4. Groups statistics & comparison.

The pipeline outputs two main folders: '*_Processed' and '*_logs', where '*' is the name of the original folder, both arranged in the same manner as the original folder. The '*_logs' folder contains all the log files for the different pipeline stages. The '*_processed' folder keeps the results from all of the stages. While the folder arranged in the same as the original, an 'Analysis' folder is made for each group, as well as for the entire dataset. The group 'Analysis' folder summarize the characteristics of each major fiber tract. For 'between' groups another 'Analysis' folder is made with the statistics and comparison between the groups. The results are both graphic i.e. figure/images, and numeric - tables containing the properties and statistics of each tract.

*** DAP does not changes the original data.**

Software Details

DAP was developed using both Unix shell and MATLAB scripts, each file has the described purpose:

Unix Shell Scripts

runPipe.sh - This is the main script that executes the whole pipeline. It calls other modules to correctly process all the given data. The script requires exactly one input - folder to be processed.

procFold.sh - This script setups the data so it is ready to be processed, it executes as a part of the pipeline. The script arranges the data in '*_Processed', while renaming and converting the files as needed.

dtiInit.sh - This is a batch script, it runs through all subjects and executes a MATLAB function for each one. This script is executed as a part of the pipeline.

afqProc.sh - This is a batch script that calls the AFQ pipeline for each subject. This script is executed as a part of the pipeline.

tractAnalysis.sh - This script executes the post-process stage, where a MATLAB script is called with all the processed data. This script can be executed either as a part of the pipeline, or independently. If the data was already processed but needed to be analyzed in a different configuration this script can be executed with exactly one input - the processed folder i.e. '*_Processed'.

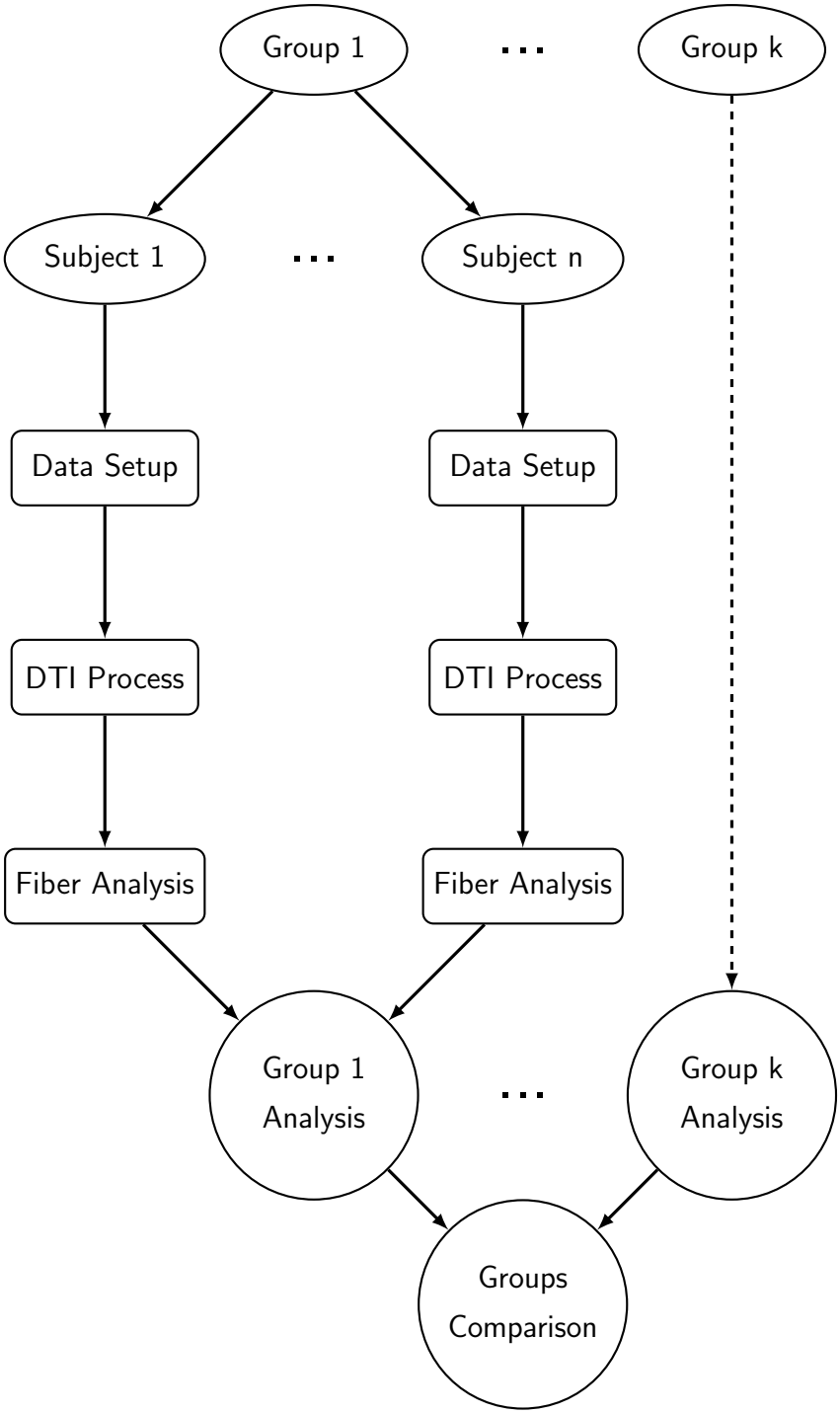

Figure 2.1: Schematic chart of the pipeline work flow.

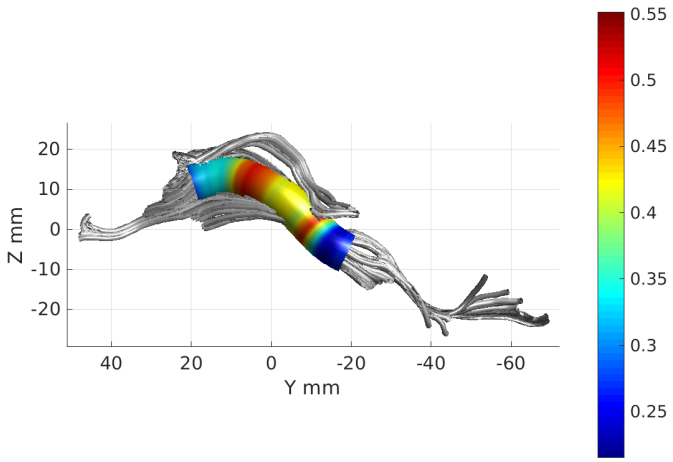

Figure 2.2: Example of a property drawn upon tract, in this case fractional anisotropy(FA) is drawn upon the right Thalamic radiation.

MATLAB Scripts and Functions

subjDti.m - This MATLAB function executes the dtiInit pipeline for a given subject subject, which includes diffusion tensor estimation and MNI registration. This function may need parameter tuning, Although the default setup should work, sometimes they need to be changed. In order to change the parameter refer to the comments inside the file and follow the instructions, one should also refer to Vistasoft instructions.

subjAfq.m - This function runs the AFQ pipeline. It relays on subjDti outputs, and computes for each subject tract characteristics.

analyzeGroups.m - This function calculates 'within' and 'between' groups statistics. It creates an 'Analysis' folder for each group, and if more than one group exists another 'Analysis' folder will be created for group comparison.

plotTractParams.m - This is an independent function for plotting fibers characteristics on-top the tract itself, see example in Figure 2.2. The

function inputs are a tract in AFQ representation, and a certain property to draw upon it, the property should be a vector with length of 100 entries.

In general the user interacts with three of the of the mentioned scripts/-functions in the following way:

```
#Shell Script
runPipe.sh [Input folder]
tractAnalysis.sh [Processed folder]
```

```
%MATLAB Function
plotTractParams(fg,vals)
```

3

Getting Started

Getting Started chapter explains how to get DAP and install it. The chapter presents briefly how to use the pipeline. These instructions will walk you through on setting up the environment and installing this pipeline on your local machine.

This pipeline was developed for and tested on Ububtu based linux - linux Mint.

Prerequisites

You will need to install the following softwares before using the pipeline:

- MATLAB.
- spm.
- Vistasoft.
- AFQ.

Refer to the softwares homepage for installation guides. You will also need to add them to your MATLAB path, this can be done by editing MATLAB

startup file called *startup.m*. The file can found at */PATH/TO/MATLAB/-toolbox/local/*. For each mentioned software you should add a path by adding a line in *startup.m* file

```
addpath(genpath(fullfile(PATH,TO,SOFTWARE)));
```

example - if spm folder is /opt/spm then add the following line

```
addpath(genpath(fullfile('opt','spm')));
```

Installing

After the environment is set you can install the pipeline simply by cloning. First go to the folder where the pipeline will be cloned to, then download the pipeline -

```
cd /PATH/TO/PIPELINE
git clone https://github.com/HagaiTz/DAP.git
```

You should also add the pipeline to MATLAB path. Also, for convenience you can add an alias so that the pipeline can be executed with ease. This is done by going to your home folder and editing *.basrc* environment file. Open *.bashrc* with your favorite editor and add the following lines at the bottom -

```
alias runPipe='/PATH/TO/PIPELINE/DAP/runPipe.sh'
alias tractAnalysis='/PATH/TO/PIPELINE/DAP/tractAnalysis.sh'
```

Quick Start

In order to run the pipeline use the command

```
runPipe [Input folder]
```

If the environment was set-up as described above, the processing should start. However, if you did not define alias then to run the pipeline you will need to type the full path for the main script

```
/PATH/TO/PIPELINE/DAP/runPipe.sh [Input folder]
```

For more details see next chapter.

4

Using the Pipeline

In order to use the pipeline and its features the data must be set in a certain structure. The following chapter will guide you in details how to setup the data in the desired structure, and running the pipeline.

Data Setup

DAP is expecting to be given data in a certain organization as an input. Any input in different structure will result in error. The pipeline main script *runPipe.sh* expects a full path to the folder containing the data, for example if the data resides in a folder called *'MRIdata'* in the home folder, the expected input will be

```
  ~/MRIdata/
# --- or ---
  /home/<user>/MRIdata/
```

The data in the folder can be arranged in one of two ways:

- A single group, where each subject is a sub-folder in the main data folder, as in Figure 4.1.
- Multiple groups, where each group is a sub-folder in the main data folder. And inside each group are the folders of the subjects, as in Figure 4.2.

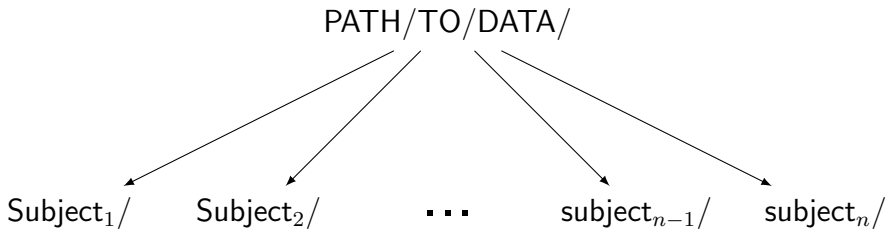

Figure 4.1: Configuration of the data folder for a single group.

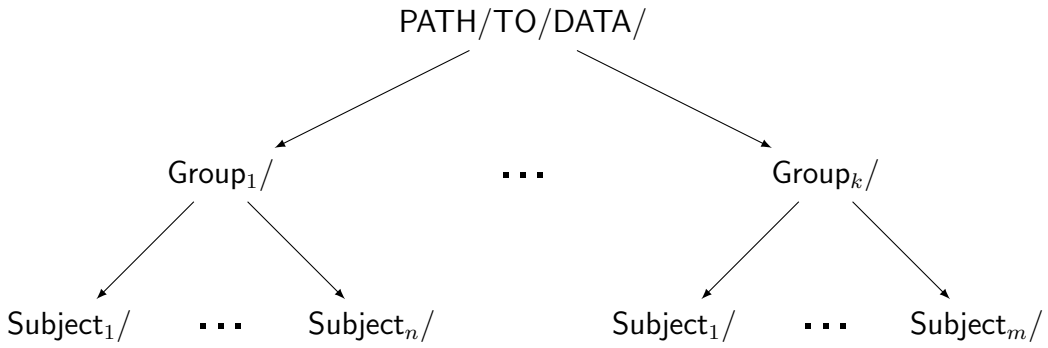

Figure 4.2: Configuration of the data folder for multiple groups.

The names of the groups and subject as well as the main data folder have no influence on the pipeline, thus you can choose them as you see fit. Furthermore, the groups size can vary as there is no restriction that the groups must be the in the same size.

Each subject folder should contain all the required MRI images in a specific structure. It can be either in **NIFTI** of **DICOM** format.

For **NIFTI** format, each subject folder should contain a T1-weighted image named 't1.nii.gz' and another sub-folder named 'raw'. Inside the 'raw' sub-folder three more files must be present representing the DTI scan b-values, b-vectors & DTI image, named 'dti.bval', 'dti.bvec' & 'dti.nii.gz' respectively, see Figure 4.3.

For **DICOM** format, each subject folder should contain sub-folder named 'dicom', that consists of two more sub-folder called 'T1' and 'DTI'. The 'T1' sub-folder should contain all the dicom files of a T1-weighted scan, while

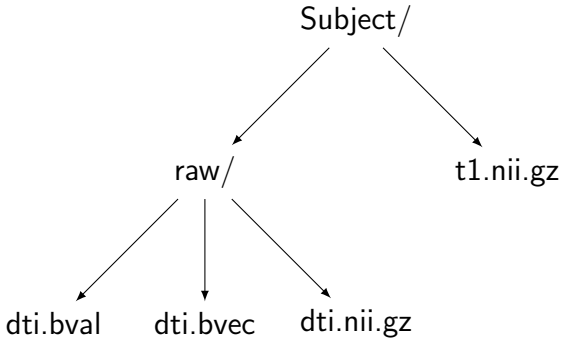

Figure 4.3: Structure of subject folder for **NIFTI** format.

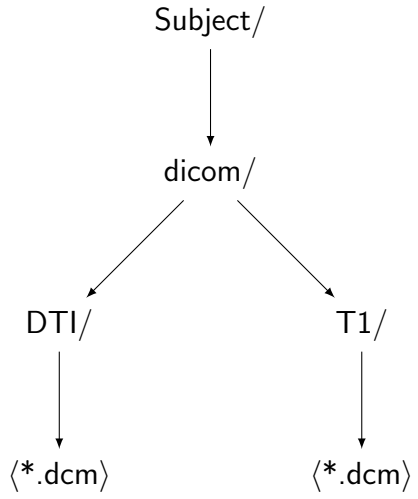

Figure 4.4: Structure of subject folder for **DICOM** format.

the 'DTI' sub-folder should contain all of the DTI scan dicom files, as in Figure 4.4.

Note, that in the subject folder the names are very important and must be exactly as described here (case sensitive), any other name will result in error. In addition, before running the pipeline you will be prompted and asked to make sure that the subjects folder are in the correct structure.

Pipeline Execution

After the data is all setup in a compatible structure, DAP is ready to be executed with the following command:

```
runPipe [Input folder]
```

Where *[Input folder]* is the main data folder.

Example - if the main data folder is *'MRIdata'* in the home directory, DAP should be provoked by entering

```
runPipe ~/MRIdata
# --- or ---
runPipe /home/<user>/MRIdata
```

in the terminal. The user will be asked to make sure that the data is in the correct structure before continuing. Then the pipeline should start automatically and process all the data. The output are given in two folders: *'*_processed'* & *'*_logs'*. Using the same example above, if the original folder was named *'MRIdata'*, then *'MRIdata_Processed'* will contain all the processed data, and *'MRIdata_logs'* will save all the logs files.

Pipeline Output

The original folder will stay untouched, all the processing is done in the *'MRIdata_Processed'* directory. Furthermore, all of the folders structure will remain the same as the original one, meaning that the groups division will remain,

The results will be saved in a sub-folder named *'Analysis'*. This sub-folder will be created for each group independently that can be found in the groups sub-folder, Figure 4.5. If only one group is found this will be the configuration of the main data folder output. If more than one group exist, the main data folder will contain another *'Analysis'* directory, Figure 4.6. The *'Analysis'* will be divided into different tracts sub-folders, each summarize the tracts characteristics across the group and between the group respectively.

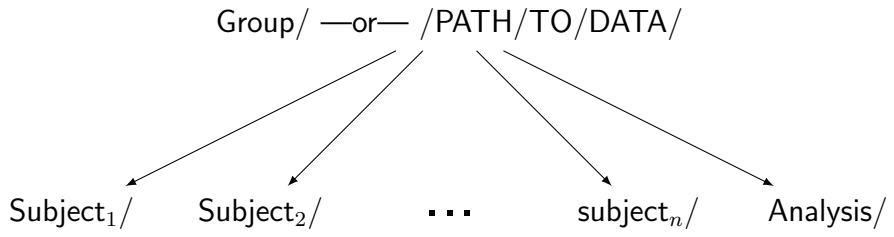

Figure 4.5: Output configuration of a single group. This output exists for each group separately, moreover, if only one group exists this will be the output structure of the main data folder.

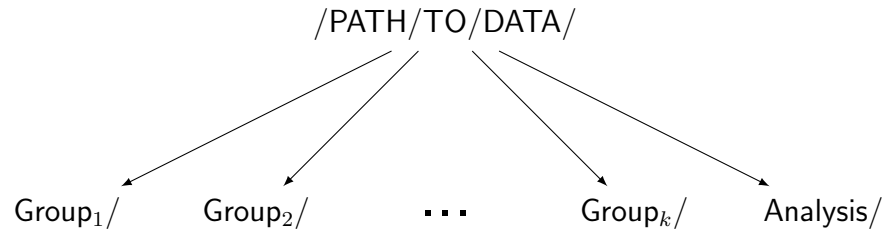

Figure 4.6: Output configuration of the main folder for multiple groups.

Additional Features

This section will cover two more features of this software, that the user can take advantage of.

Tract Analysis

In addition to the main pipeline script, the user can provoke another script for recalculating the 'within' and 'between' group statistics. The script can be executed using the command

```
tractAnalysis [Input folder]
```

where the *[Input folder]* is a processed folder. The processed folder should be data that underwent the pipeline. This script is useful when the user want to permute the groups, and recalculate the statistics without running the whole pipeline again. This is possible since each individual is processed

independently and the results are saved in the subject sub-folder, thus by moving the subject folder from one group to another will not change the characteristics of that subject, only the groups statistics. This script will overwrite the existing *'Analysis'* folder, however, it is highly recommended to erase all current *'Analysis'* folder to be on the safe side.

Example - if the original folder was named *'MRIdata'*, and the output named *'MRIdata_Processed'*. The user then changed the groups in *'MRIdata_Processed'*, in order recalculate the statistics the user simply executes the command

```
  tractAnalysis ~/MRIdata_Processed
# --- or ---
  tractAnalysis /home/<user>/MRIdata_Processed
```

and the groups statistics will update accordingly.

Tract Plot

Another useful feature is the ability to plot tracts with some characteristics on-top as in Figure 2.2. This can be done through MATLAB by using the function

```
  plotTractParams(fg,vals)
```

while the inputs are

fg - A tract in AFQ structure.

vals - Some property in the shape of a vector at length 100.

Example - The user wants to plot the right Thalamic radiation with the fractional anisotropy(FA) property, as in Figure 2.2. First the user should refer to Table 4.1 (or Table 4.2 for tracts in the Corpus Collosum), and find which file contain the desired tract and its index. Then the file should be loaded, and the the tract index should be chosen. In addition, the value to be plotted should be loaded, or hard coded before calling the function. This is a possible implementation:

```
% Index of the right Thalamic radiation.
idx = 2;
% Path to subject1 in group1 in the processed folder.
subjectFolder = fullfile('home','<user>',...
  'MRIdata_Processed','group1','subject1');
% Path to fibers folder
fiberFolder = fullfile(subjectFolder,'dti','fibers');
% Load the subject fibers
fg = dtiReadFibers(fullfile(fiberFolder,...
  'MoriGroups_clean_D5_L4.mat'));
% Get the desired tract from the list
fg = fg(idx);
% Load the subjects data
load(fullfile(subjectFolder,'dti','AFQ.mat'));
% Load FA for the right Thalamic radiation
fa = afq.vals.fa{idx};
% Plot the tract with property.
plotTractParams(fg,fa);
```

The value can doesn't have to be loaded from the subject itself, any vector with 100 values will suffice. Also, the values can be defined directly as in

```
% Define vector with elements 1,2,...,100
val = 1:100
% Plot the tract with property.
plotTractParams(fg,val);
```

Table 4.1: Main tracts file names and indexes.

Tract Name	File Name	Index
Left Thalamic Radiation	MoriGroups_clean_D5_L4.mat	1
Right Thalamic Radiation	MoriGroups_clean_D5_L4.mat	2
Left Corticospinal	MoriGroups_clean_D5_L4.mat	3
Right Corticospinal	MoriGroups_clean_D5_L4.mat	4
Left Cingulum Cingulate	MoriGroups_clean_D5_L4.mat	5
Right Cingulum Cingulate	MoriGroups_clean_D5_L4.mat	6
Left Cingulum Hippocampus	MoriGroups_clean_D5_L4.mat	7
Right Cingulum Hippocampus	MoriGroups_clean_D5_L4.mat	8
Callosum Forceps Major	MoriGroups_clean_D5_L4.mat	9
Callosum Forceps Minor	MoriGroups_clean_D5_L4.mat	10
Left IFOF	MoriGroups_clean_D5_L4.mat	11
Right IFOF	MoriGroups_clean_D5_L4.mat	12
Left ILF	MoriGroups_clean_D5_L4.mat	13
Right ILF	MoriGroups_clean_D5_L4.mat	14
Left SLF	MoriGroups_clean_D5_L4.mat	15
Right SLF	MoriGroups_clean_D5_L4.mat	16
Left Uncinate	MoriGroups_clean_D5_L4.mat	17
Right Uncinate	MoriGroups_clean_D5_L4.mat	18
Left Arcuate	MoriGroups_clean_D5_L4.mat	19
Right Arcuate	MoriGroups_clean_D5_L4.mat	20

Table 4.2: Corpus Collosum tracts file names and indexes.

Tract Name	File Name	Index
Occipital	CC_Occipital_clean_D5_L4.mat	1
Post Parietal	CC_Post_Parietal_clean_D5_L4.mat	1
Sup Parietal	CC_Sup_Parietal_clean_D5_L4.mat	1
Motor	CC_Motor_clean_D5_L4.mat	1
Sup Frontal	CC_Sup_Frontal_clean_D5_L4.mat	1
Ant Frontal	CC_Ant_Frontal_clean_D5_L4.mat	1
Orb Frontal	CC_Orb_Frontal_clean_D5_L4.mat	1
Temporal	CC_Temporal_clean_D5_L4.mat	1