MELD Protocol 2 - Instructions for FreeSurfer Cortical Segmentations Version 2

Sophie Adler,Kirstie Whitaker,Mira Semmelroch,Konrad Wagstyl

Abstract

The MELD Project is an international collaboration aiming to create open-access, robust and generalisable tools for FCD detection. To this end, we will train a neural network classifier on MRI features from FCD patients from multiple centres worldwide.

Protocol 2 provides instructions on how to create FreeSurfer cortical segmentations.

These instructions are based on the freely available protocols on the ENIGMA-epilepsy website http://enigma.ini.usc.edu

We are very grateful to Derrek Hibar, Neda Jahanshad, Roberto Toro, Jerod Rasmussen, Theo van Erp who wrote the orginal ENIGMA protocols and offered them with an unlimited license without warranty!

The main changes are the paths, which are now directed to the meld folder.

Citation: Sophie Adler,Kirstie Whitaker,Mira Semmelroch,Konrad Wagstyl MELD Protocol 2 - Instructions for FreeSurfer Cortical Segmentations. **protocols.io**
dx.doi.org/10.17504/protocols.io.n3edgje
Published: 02 May 2018

Guidelines

These instructions are based on the freely available protocols on the ENIGMA-epilepsy website http://enigma.ini.usc.edu/ongoing/enigma-epilepsy/enigma-epilepsy-protocols/

We are very grateful to Derrek Hibar, Neda Jahanshad, Roberto Toro, Jerod Rasmussen, Theo van Erp

who wrote the orginal ENIGMA protocol. The main changes are the paths, which are now directed to the meld folder.

Before start

Ensure that you have "cloned" the MELD file structure from
github. (https://github.com/MELDProject/meld)

To do this, in a terminal window cd into the location you wish to store the data.

`cd <path>`

Then "clone" the repository using the following command:

git clone https://github.com/MELDProject/meld

This will download all of the scripts, template folder structure and template control data necessary for the MELD preprocessing.

This contains the MELD_site_code_participants.csv file that is required for this protocol.

If you have any questions or run into problems, please feel free to contact the MELD project: (meld.study@gmail.com)

Protocol

Preparations

Step 1.

Double check that you have downloaded the MELD project file structure from github.

Please see guidelines of protocol 1 or 2 for instructions on how to do this.

Preparations if you have already run FreeSurfer on your subjects

Step 2.

If you have already created freesurfer reconstructions for your subjects for a previous study, you can reuse these.

Copy each whole subject folder into the meld/output folder and then rename them according to the MELD naming structure (described in Protocol 1).

ENSURE to keep a spreadsheet linking the original IDs to the new MELD IDs.

Preparations

Step 3.

Input Folder

Your input-folder should contain the scans of all your participants in a nii.gz-format.

Please create a folder for each participant e.g.

```
mkdir <path>/meld/input/MELD_H1_3T_FCD_0001
```

In each subjects folder, create a folder called T1 (and a folder called FLAIR)

```
mkdir <path>/meld/input/MELD_H1_3T_FCD_0001/T1
mkdir <path>/meld/input/MELD_H1_3T_FCD_0001/FLAIR
```

Place the T1 .nii.gz file of each participant in the T1 folder (and if available the FLAIR .nii.gz file in the FLAIR folder)

Preparations

Step 4.

Naming your participants T1 scans

Make sure that each T1 nii.gz-file is called "*MELD_[site code]_[scanner code]_[patient/control]_[number].nii.gz*".

[site code] = site identifier which will be provided to you e.g. *H1* for Great Ormond Street Hospital

[scanner code] = 15T if 1.5T scans or 3T if 3T scans

[patient/control] = FCD if patient, C if control

[number] = 0001, 0002 etc.

Examples of scan naming structure:

MELD_H1_15T_FCD_0001.nii.gz

MELD_H1_3T_C_0002.nii.gz

Preparations

Step 5.

If you have 3D FLAIR scans name them:

"*MELD_[site code]_[scanner code]_[patient/control]_[number]_FLAIR.nii.gz*".

Preparations

Step 6.

It is recommended to use the c-shell or enhanced c-shell for FreeSurfer, you can do this by simply typing:

csh or: *tcsh*

You can set this shell permanently as your default:

chsh -s /bin/csh or: *chsh -s /bin/tcsh*

Setup FreeSurfer

Step 7.

Download FreeSurfer & Register for a license

All information on how to set up and install FreeSurfer can be found on this webpage:

https://surfer.nmr.mgh.harvard.edu/fswiki/QuickInstall

Make sure that you also register to obtain a license to use FreeSurfer:

https://surfer.nmr.mgh.harvard.edu/registration.html

Remember to move the license.txt file you receive into your FreeSurfer file.

Setup FreeSurfer

Step 8.

Before you want to work with FreeSurfer, you must make sure three things have happened:

The variableFREESURFER_HOME is set (so your computer knows where FreeSurfer is installed):

```
setenv FREESURFER_HOME <freesurfer_installation_directory>/freesurfer
```

Setup FreeSurfer

Step 9.

Before you want to work with FreeSurfer, you must make sure three things have happened:

The FreeSurfer set up script must be sourced (so FreeSurfer knows the location of everything it needs):

```
source $FREESURFER_HOME/SetUpFreeSurfer.csh
```

Setup FreeSurfer

Step 10.

Before you want to work with FreeSurfer, you must make sure three things have happened:

FreeSurfer has been pointed to a directory of subjects to work on:

setenv SUBJECTS_DIR <path>/meld/output

Run Preprocessing Pipeline

Step 11.

The next step is to run “recon–all” on the subjects in your input-folder (http://surfer.nmr.mgh.harvard.edu/fswiki/recon-all). In your output-folder, make a text-file (e.g. *nano List_subjects.txt*) containing a list of all your subjects:

MELD_H1_15T_FCD_0001
MELD_H1_15T_FCD_0002
MELD_H1_15T_FCD_0003
MELD_H1_15T_FCD_0004
...

Run Preprocessing Pipeline

Step 12.

Create the following script (e.g. *nano loop_recon-all*) to run “recon-all” on multiple subjects:

```
#!/bin/bash
exec <List_subjects.txt
while read x; do
recon-all -i ../input/$x/T1/*.nii -s $x -all
done
```

It should be created in the output folder.

Run Preprocessing Pipeline

Step 13.

To run the *loop_recon-all* script, make it executable with the following command:

chmod u+x
loop_recon_all

Run Preprocessing Pipeline

Step 14.

Run the script:

```
./loop_recon_all
```

Depending on the number of your scans and the processing speed of your computer, this script will take several days to finish (24 to 36 hours/subject). When “recon-all” is done, you will see a folder for each subject in your output-folder, in which you will find 10 new folders (such as ‘mri’, ‘stats’, ‘surf’ etc).

Run Preprocessing Pipeline with FLAIR

Step 15.

If you have 3D FLAIR scans for your participants:

The next step is to run “recon–all” on the subjects with FLAIR in your input-folder (http://surfer.nmr.mgh.harvard.edu/fswiki/recon-all). In your output-folder, make a text-file (e.g. *nano List_subjects_FLAIR.txt*) containing a list of all your subjects with FLAIR:

```
MELD_H1_15T_FCD_0001
MELD_H1_15T_FCD_0002
MELD_H1_15T_FCD_0003
MELD_H1_15T_FCD_0004
…
```

Run Preprocessing Pipeline with FLAIR

Step 16.

If you have 3D FLAIR scans for your participants:

Amend the script *loop_recon_all* to co-register the FLAIR:

```
#!/bin/bash
exec <List_subjects_FLAIR.txt
while read x; do
recon-all -i ../input/$x/T1/*.nii -s $x -FLAIR
../input/$x/FLAIR/*.nii -FLAIRpial -all
done
```

Run Preprocessing Pipeline with FLAIR

Step 17.

To run the *loop_recon-all* script, make it executable with the following command:

chmod u+x
loop_recon_all

Run Preprocessing Pipeline with FLAIR

Step 18.

Run the script:

./loop_recon_all

Depending on the number of your scans and the processing speed of your computer, this script will take several days to finish (24 to 36 hours/subject). When “recon-all” is done, you will see a folder for each subject in your output-folder, in which you will find 10 new folders (such as ‘mri’, ‘stats’, ‘surf’ etc).

Warnings

PLEASE DO NOT SHARE ANY IDENTIFIABLE DATA

Data sharing only occurs at the level of anonymised demographics information and anonymised data matrices. These are in a template space that cannot be traced back to an individual.