Correct flips on bvec

Sébastien Dam

March 2022

1 Files

- *download_data_on_Shanoir_and_BIDS_reorganisation.py* : Script to download and BIDS-like organize data on Shanoir using "shanoir_downloader.py" developed by Arthur Masson. Codes provided by Malo Gaubert. Downloads archive with DICOM files as zip files in the folder `dicom` ;
- *download_data_on_Shanoir_and_BIDS_reorganisation_diffusion.py* : Script adapted from the previous one to download diffusion data. This file is needed to cope with datasets whose name includes a '/'. Otherwise, as the script sequentially downloads files, each of them is named the same way and it would result in a single file downloaded at the end. As a result, all of the subjects were added in a list (copied pasted from an Excel file) as well as a `For` loop to rename .zip files ;
- *shanoir_downloader.py* : Needed to download data from Shanoir, see GitHub.
- *extract_nifti_from_dicom.py* : Script to extract DICOM .zip files and convert them to NIFTI using dcm2niix. The important files are the ones with the extensions .dcm, .bval, .nii.gz ;
- *modifiedAnimaDiffusionPreprocessing_Sebastien_flip.py* (to rename later) : Script adapted from animaDiffusionImagePreprocessing.py from Anima-Scripts. The changes added consist in taking the 31 first rows of `bvecs_corrected` and change some extensions from .nii.gz format to .nrrd.
- *MAIN_DTIpreprocess_LONGIDEP_Sebastien.sh* (to rename later) : Script to apply the correct conversion from DICOM to NIFTI for all the subjects. You might want to change the directories for the input files and folders.

2 Process

1. Download DICOM files on Shanoir :
`python ./download_data_on_Shanoir_and_BIDS_reorganisation.py` and type the password
2. (optional) If diffusion DICOM files are downloaded :
`python ./download_data_on_Shanoir_and_BIDS_reorganisation_diffusion.py` and type your password
3. Run `python ./extract_nifti_from_dicom.py diffusion` (may need to replace *diffusion* with another name of the subdirectory containing the sequences)
4. Upload these files on Igrida : `scp -r` *`path_to_dicom_folder user`*`@igrida-frontend:`*`some_path`*

On Igrida

5. Run `bash MAIN_DTIpreprocess_LONGIDEP_Sebastien.sh`

3 Tips

As it is not practical to debug on Igrida, it is best to try this process on your local machine. For that, you would need to install Anima. Here's the installation process for compiling from source with errors that I have encountered, inspired from anima.readthedocs :

1. Create an Anima-Public folder
2. Inside it, clone the repository from github (use the first line by default, the second if you have set up your SSH keys): keys):

```
git clone https://github.com/Inria-Empenn/Anima-Public.git src
git clone git@github.com:Inria-Empenn/Anima-Public.git src
```

3. then, run CMake in a new build folder, change any options if you wish to change the default compilation (which downloads and compiles all dependencies and tools)

```
mkdir build
cd build
ccmake ../src
```

4. Install OpenGL developer version : (on fedora)

```
sudo dnf install mesa-libGL-devel
```

5. Build using your environment (a make or ninja will be enough on Linux and OSX, open Visual Studio on Windows)
6. At root, change the *config.txt* in the folder *.anima* so that the path corresponds to your folder where Anima-Public is located.