

User Manual

Yong Li

September 6, 2013

1 Introduction

The GUI consists primarily of 3 parts, which is *Diffusion Pre-processing*, *T2 Pre-processing* and *Cortical Analysis*.

Feel free to email yong.loh12@imperial.ac.uk if you found any bugs!

2 Setting Up The GUI

Copy and paste the whole *GUITesting* folder in any directory. Open up a terminal, and hover to the directory containing the folder by using the **cd** command (Using the **pwd** command you should see something like: `/staff/yl13/GUITesting`) and run the command:

```
chmod 777 -R GUITest/
```

to make every file runnable. This step is only required to be done exactly **once** when first using the GUI.

3 Running The GUI

Run the GUI by double clicking the *RUNME* file and click Run.

3.1 Creating a shortcut for the GUI

Alternatively, drag the *RUNME* file to the top panel of any GNOME-classic interface(using the mouse) and this will create a shortcut at the top of the panel. Run the GUI by clicking the icon you just created and this will run the GUI.

If this doesn't work, you can manually create a desktop shortcut by using the following steps:

1. In the Desktop create an "Untitled Document". Just right click in the Desktop and select Create New Document.
2. Edit the file with Gedit and add the following lines:
[Desktop Entry]
Type=Application

3. Save the file and THEN rename it to whatever you want but at the end of the name add `.desktop`. For example, the name could be `gui.desktop`.
4. Now you should be able to right click the file and click **Properties**. In the command tab, type in the full path to *RUNME*, for example `/staff/yl13/GUITesting/GUITest/RUNME`, and remember to set **Permissions** as executable!

3.2 If GUI Doesn't Run

If *RUNME* doesn't run at all, i.e. if nothing pops up after clicking Run, then please open terminal and type in the following command:

```
java -version
```

If the output is:

```
java version "1.6.0_27"
```

or anything less than 1.7, the java version must be updated to version 1.7. An easy way is to open **Ubuntu Software Centre** and search for java in the search tab, then install OpenJDK 1.7. After installing, try typing the command "java -version" again and if the output is still the same version, then type in these two commands(starting with \$):

```
$update-java-alternatives -l
java-1.6.0-openjdk 1061 /usr/lib/jvm/java-1.6.0-openjdk
java-1.7.0-openjdk-amd64 1051 /usr/lib/jvm/java-1.7.0-openjdk-amd64
$sudo update-java-alternatives -s java-1.7.0-openjdk-amd64
```

Ignore the errors and typing "java-version" should now be the updated one.

4 Before Using the GUI

4.1 Suggested Input Format

Since all 3 pipelines output files in the same directory as the input file, it is highly recommended(almost a must to prevent error) to use a seperate folder for each image file to process. An example would be instead of having all 4 images files in same directory, such as:

1. `/staff/yl13/Testing/Data1.nii.gz`
2. `/staff/yl13/Testing/Data2.nii.gz`
3. `/staff/yl13/Testing/Data3.nii.gz`
4. `/staff/yl13/Testing/Data4.nii.gz`

It is highly recommended to change the above format to something like:

1. `/staff/yl13/Testing/Data1/Data1.nii.gz`

2. /staff/yl13/Testing/Data2/Data2.nii.gz
3. /staff/yl13/Testing/Data3/Data3.nii.gz
4. /staff/yl13/Testing/Data4/Data4.nii.gz

in which all image files are in their individual unique folder, yet in the same parent folder, which is /staff/yl13/Testing. This will ease the process to locate all the image files and outputting the files to their correct folder. Extra input prerequisites is mentioned in the beginning of the following sections.

****Important Note:** All inputs must be in .nii or .nii.gz extensions, other extensions will cause the process not being able to run.

4.2 Expected Output

Upon completion, the expected outcome is the terminals that are running the process will be closed automatically. If an error occurred, the terminal window will stay open so that you can see where the error occurred and (possibly) find a fix for the error.

Also note that the files will be overridden automatically if you rerun the same process.

5 The Main Frame

If *RUNME* successfully runs, you should be able to see a small frame appearing at top left corner specifying all 3 parts of pipeline mentioned earlier, which is ***Diffusion Pre-processing, T2 Pre-processing, Cortical Analysis*** and a ***Exit*** button. All 3 process uses a very similar interface with very minor changes, which all of them have both a *Auto* and *Manual* view.

5.1 Diffusion Pre-processing

As mentioned, there are two views for this process, which will be explained below.

****Important Note:** Every input must have at least one ****bvecs**** or ****bval**** present in the folder containing the image file, in which ******* means matches everything. For example:

Input data to process is: /staff/yl13/Testing/NNU123/unreg-data.nii.gz.
Please check that inside the folder /staff/yl13/Testing/NNU123/, it must contains both bvecs and bval.

****Please check input data and ensure that volumes is equal to the number of bvec or bval entries, i.e. no trace image.**

5.1.1 Diffusion Pre-processing(Auto)

By clicking Diffusion Pre-processing from the Main Frame, another frame which are titled as “Diffusion Pre-processing(Auto)” would open up. In *Auto* view, you just have to specify the parent folder. For example, if images to process are

1. /staff/yl13/Testing/NNU123/unreg-data.nii.gz
2. /staff/yl13/Testing/NNU456/unreg-data.nii.gz
3. /staff/yl13/Testing/NNU789/unreg-data.nii.gz
4. /staff/yl13/Testing/NNU1000/unreg-data.nii.gz

Steps required are as follows:

First specify /staff/yl13/Testing in input directory by using the **Open** Button. Next type unreg-data.nii.gz(The default is unreg-data.*) in the text field besides **File name to search(1)** and hit next. This will proceed to the Diffusion Pre-processing(Manual) frame with all inputs populated(The next step will be continued at **Diffusion Pre-processing Manual** section marked \$\$).

****Note:** You can add more files name to search by using the **Add more files to search** button, and note that * matches everything, e.g. unreg-data.* will work for the example above as well.

Alternatively, you can also use the **Switch View to Manual** button to manually select the image files.

5.1.2 Diffusion Pre-processing(Manual)

In this view, you’ll have to manually specify all image files by using the **Open** button. For example, if images to process are the same as above, steps required are as follows:

First increase the number of datasets by dragging the **slider** besides the label **Number of datasets** to 4 and hit the **Go** button besides it. This will increase the available inputs to 4(instead of 1). Next specify the full path to the image files using the **Open** button. The final result should look like:

Number of datasets : 4

Input data 1: /staff/yl13/Testing/NNU123/unreg-data.nii.gz

Input data 2: /staff/yl13/Testing/NNu456/unreg-data.nii.gz

Input data 3: /staff/yl13/Testing/NNU789/unreg-data.nii.gz

Input data 4: /staff/yl13/Testing/NNU1000/unreg-data.nii.gz

\$\$ If you used the **Diffusion Pre-processing(Auto)** and followed the steps in the previous section, you should reach the same stage as using the manual up until this point. The next steps is to determine whether to run **bedpostx** and **eddy_correct** by using the check box. Finally, to run the process, simply hit

the **Go** button and the process should start. Additional buttons such as **Switch View to Auto** allows you to go back to the *Auto* view, while the **Clear** button clears all entries typed in the inputs. Additionally, you can specify how many process to run in parallel by typing the number in the textfield besides the **Number of Concurrent Process** label.

5.2 T2 Pre-processing

There are also two views for this process, which will be explained below.

5.2.1 T2 Pre-processing(Auto)

The steps required here is very similar to the *Diffusion Pre-processing(Auto)*, please refer to the section above for more details. Some noticeable changes are the additional **Age file**. Using any text editor(such as gedit), create a text file that contains the age at scan corresponding to the input files. For example, if the images to process are:

1. /staff/yl13/Testing/NNU123/NNU123_T2.nii.gz
2. /staff/yl13/Testing/NNU456/NNU456_T2.nii.gz
3. /staff/yl13/Testing/NNU789/NNU789_T2.nii.gz
4. /staff/yl13/Testing/NNU1000/NNU1000_T2.nii.gz

simply create a text file that looks like this:

```
28
33
40
44
```

in which the subject NNU123 is scanned at age 28, etc. The format is type in 28 and hit *Enter* and continue with 33, and hit *Enter*, etc. The order of the age should match the same way as how the system sort the files. After saving the file, back in the GUI, specify the full path to the age file using the **Open** button besides it.

****Important Note:** Age text file can be left blank and it will populate all inputs to age of 36(by default), in which you can still manually change it in the *Manual* view after hitting the **Next** button.

5.2.2 T2 Pre-processing(Manual)

This view will be of a very familiar sight if you used *Diffusion Pre-processing(Manual)*, with an exception of an extra field for each input, which is the **Age at scan(weeks)**. There will also be a check box for running **parcellation and segmentation**. There's also a button **Run Segmentation Only** which is a convenient way if you've completed the T2 Pre-processing(without checking the **parcellation and segmentation** box) and decides to run segmentations only. This will avoid repeating the same process. For more details about the other parts, simply refer to the *Diffusion Pre-processing(Manual)* section.

5.3 Cortical Analysis

This pipeline uses exactly the same interface as *T2 Pre-processing* but is a continuation of the previous process. In the *Manual* view, you will need to specify a **Template_age** as well.

For this pipeline, there is a few important input prerequisites that must be followed:

1. *T2 Pre-processing* must be done beforehand with all the outputs in the same folder as the input.
2. If **Diffusion Space** is checked, *Diffusion Pre-processing* is expected to be finished and the output must be inside the same folder as the input as well.

An example would be:

Input data 1: /staff/yl13/Testing/NNU123/filename.nii.gz.

So in the folder /staff/yl13/Testing/NNU123, it should contains (assuming **Diffusion Space** is checked):

1. /staff/yl13/Testing/NNU123/segmentations, which contains all 3 expected output after running segmentation, and other files outputted from *T2 Pre-Processing*.
2. /staff/yl13/Testing/NNU123/dti_FA.nii.gz, other dti files, /staff/yl13/Testing/NNU123/nodif_brain.nii.gz, and other files outputted from *Diffusion Pre-processing*