**Bash 101**

Note: This is only relevant if you are working on a Mac or Linux system.

**What is bash and why should I use it?**

When you open your Terminal on a Mac, what you see is a bash shell. This bash shell is an interface to your operating system. From here, you can input commands to access files or programs on your computer. Doing this will accomplish the same task as double clicking to open a program and then using the graphical user interface (GUI) of the program to interface with it. The main difference is that by using the Mac Terminal, you will interface with programs and files via the command line (CLI) instead of a GUI.

Being able to use bash is essential for most image analysis techniques discussed within this collective of documentation. For example, programs like Freesurfer and FSL become significantly more versatile by interfacing with them via bash. Plus, bash allows you to quickly move, delete, rename, copy, or otherwise reorganize long lists of files very quickly (along with many other convenient perks).

[Here is a quick list of common, useful bash commands](https://www.educative.io/blog/bash-shell-command-cheat-sheet).

**It will be helpful to get familiar with bash quickly by opening the Terminal and trying out some simple commands.**

For example, if you want to copy a file, try using cp [/path/to/file.fileextention] [/new/file/location/file.fileextension] – you will replace the first path with the current file location and the second path with location you’d like your file to be.

Other useful commands that work with similar logic are rm (which will delete a file or folder you specify) and mv (which will move a file or folder you specify). Be careful with these as they will permanently change the file you give them.

I also recommend trying to accomplish simple tasks via bash by web searching how to do the tasks. For example “how to open text files with bash” or “how to delete all files with a specific file extension in bash”, etc.

**Logical loops** may also be useful to you. These will allow you to carry out commands across numerous files or folders depending on specific logic.

For example, say you have a dataset of diffusion data across numerous subjects and you want to preprocess all of the subjects with a single command. You could take your base command and put it inside of a for loop. Recall the PyDesigner command example from 01\_Diffusion\_Image\_Analysis\_101. If we wanted to execute that command across numerous subjects, we could set up a for loop that looks like this:

for ID in Subj1 Subj2 Subj3 ; do

pydesigner --denoise --degibbs --mask -w --force /desktop/user/PyDesigner-

Example/${ID}/nifti/DKI.nii, /desktop/user/PyDesigner-

Example/${ID}/nifti/B0.nii -o /desktop/user/PyDesigner-

Example/${ID}/pydesigner

done

Let’s walk through this command:

* The first line contains the following elements:
  + “for” – this indicates that the following commands are going to be executed for every unit in a list
  + “ID” – this is a chosen variable that will stand in for each unit in the following list; this can be whatever you want. For this example, I chose “ID” to indicate that it will stand in for a list of subject IDs
  + “in” – this indicates that the list of items that will be affected by the commands is about to begin
  + “Subj1 Subj2 Subj3” – this is an example list of my subject names; this list could theoretically be a list of any series; no commas are necessary to distinguish one unit from another, spaces will do that.
  + “do” – this indicates that what comes next is the actual command
* The indented set of lines is the actual command that will be executed based on the for loop you’ve set up above.
  + The important takeaway from this is the “${ID}” element. The dollar sign and brackets around our “ID” variable name tell the command that it will be iteratively going through a list we specified (Subj1 Subj2 Subj3)
    - For example, the command is going to take the path we’ve given it /desktop/user/PyDesigner-Example/${ID}/nifti/DKI.nii and it will sub in each unit in our list for the “${ID}” element by looking for the following paths:
      * /desktop/user/PyDesigner-Example/Subj1/nifti/DKI.nii
      * /desktop/user/PyDesigner-Example/Subj2/nifti/DKI.nii
      * /desktop/user/PyDesigner-Example/Subj3/nifti/DKI.nii
    - This is extremely same process will be carried out for the entire command so that each subject is processed through PyDesigner using only the single command within the for loop
* “done” tells the for loop to close and run the commands wrapped inside of it.

**How to add elements to your bash profile**:

In order to access software such as Freesurfer or FSL via your Terminal, it is first helpful to add these programs to your bash profile. This tells bash where the source codes for these programs live on your computer so that all you need to do is type specific commands into your terminal rather than entire paths to the code for each command.

A screenshot of a cell phone

Description automatically generated

As you can see in the example above, some pieces of software are set up to automatically add elements to your bash profile upon installation (such as MRtrix3). However, some elements need to be added manually.

You can do this by doing the following:

1. Open a terminal window
2. Type vi .bash\_profile
3. When you see your bash profile open (which may be empty at first, type “i” (this stands for “insert” and will allow you to type things into the window that you will later save.
4. Start by adding a header to indicate which program you’re adding to your bash profile by using #. # at the beginning of a line indicates that a line is a comment and will not by read by any program as usable script. As you can see, I have headings for all programs beginning with # in my profile above.
5. Add whatever you need to add\* dependent on the program. Typically, this will involve inputting export PATH=$PATH:/path-to-application/
6. Hit “Esc”, then “;” (the semi-colon), then “wq” (this stands for “write/save and quit”, then “enter”.

\*If you are uncertain of what you need to add to your profile to get bash to recognize the program you want it to recognize, you can copy the lines in the example above based on the program in question. Or you can Google “add [program] to bash profile”

After you successfully add a program to your bash profile, you should be able to access all of the call-able commands the software offers. Such as “bet” or “flirt” from FSL or “recon-all” from Freesurfer.

**Programs to consider adding to your bash profile**: PyDesigner, FSL, MRIcroGL, Freesurfer, MRtrix3, or anything you will regularly call from your Terminal.