**MCB 525, Fall 2016**

**2.0: Manipulating files on the command-line**

Hello again command-line world! In this section, we will copy the necessary read files and scripts into our working directory and become more comfortable manipulating text files on the command-line.

**Objectives:**

* Learn to make and navigate directories
* Edit, copy, and view files on the command line

**Protocol:**

1. Navigate to your home directory and make 3 new folders:
   * one for your assigned unknown sample
   * one for CC7 (a clone of the sea anemone that was used to make the *Aiptasia* reference genome, a sort of *in silico* positive control.
   * one folder to contain a slew of scripts we will use to process the read data later on.

$ mkdir <new\_folder\_name>

$ cd my\_new\_CC7\_folder/

1. Navigate to the directory for CC7:
2. What is the full path to your current directory?

$ pwd

# The pwd command, for ‘print working directory’, tells you the complete path or “address” to your current directory.

You should see something like this:

/raid1/home/zoo/weissem/2bRAD/CC7

1. From your current directory, take a peek at the files in a different, far-off directory, the MCB 525 class folder:

$ ls /nfs1/Teaching/MCB525/525f\_16/

1. We don’t want to change any of these files in the class directory! Instead, **make a copy** of the reads file for CC7, putting the copy in your own folder for CC7:

$ cp <path/to/file/CC7\_reads.fastq.gz> <where\_to\_put\_it\_from\_step\_2.0c>

1. We can also use a single dot as shorthand in place of the path to our current directory. Try this method to copy read files from your assigned unknown sample to the directory for your unknown sample:

$ cd ../unknown\_sample\_folder/

$ cp <path/to/your/unknown/reads.fastq.gz> .

#Two dots is shorthand for the path to the parent directory (the folder that contains the current directory).

1. Now, lets copy all the scripts from the MCB class folder to our ‘scripts’ directory:

$ cd ../scripts/

$ cp /Teaching/MCB525/scripts/\*.pl .

$ cp /Teaching/MCB525/scripts/\*.sh .

#The \*.pl will match any file that ends in .pl

#The \*.sh will match any file that ends in .sh

1. The MCB course folder also contains a file named SampleInfo.txt which contains information about your assigned sample. Using the commands you learned above, make a copy of the sample info file, putting it into a new directory or a directory of your choosing.
2. We can change the name of this file using the mv command. Try changing the name of the SampleInfo.txt file:

$ mv SampleInfo.txt <new\_file\_name.txt>

#careful not to overwrite an existing file! (this will happen if a file with the new name already exists.)

1. Text files can be edited using a command line text editor. Any will do (but I find nano to be very user-friendly!):

$ nano my\_sample\_info\_file.txt

This will open the file for editing. We can move around using the arrow keys, enter and delete text, etc. To exit and return to the command line, type CTRL+X. There are several other keyboard short cuts you can explore at the bottom of the nano window, such as saving without closing (CTRL+O) or cutting/uncutting lines of text (CTRL+K/CTRL+U). You can also create brand new file using the command above, if a file of that name doesn’t already exist.

Edit your sample info file to contain only information about your unknown sample.

Though we didn’t specifically go over them, many other useful commands exist for navigating and manipulating files on the command line e.g.: <https://kb.iu.edu/d/afsk> . One we did not discuss is rm, which will remove a file. Be very careful with rm, as sadly there are no commands that will un-delete files once they are gone…