Lesson 1 START Lab 2

Questions:

- What is a command shell and why would I use one?
- How can I move around on my computer?
- How can I see what files and directories I have?
- How can I specify the location of a file or directory on my computer?

Objectives:

- Describe key reasons for learning shell.
- Navigate your file system using the command line.
- Access and read help files for bash programs and use help files to identify useful command options.
- Demonstrate the use of tab completion, and explain its advantages.

One common command that you will want to remember is 'print working directory'

```
pwd
>/Users/jeffreymiller/Desktop/gen711_lab
```

- If this doesn't end in 'gen711_lab', we need to help you navigate to the right place.

```
ls
>lab1_notes.md            lab1_notes.sh            pre-lab-instructions.md  shell_data
```

- If your terminal gets messy, and you do not want to look at the stuff you just ran, use 'clear'

```
clear
```

- For the lab, we are going to work on the files that we downloaded from canvas. We want to navigate into the 'shell_data' folder (or directory). If the 'ls' from above told you that the folder is there, we can use:

```
cd shell_data
```

- If the folder downloaded and extracted correctly, and you run 'ls' again from inside the shell_data directory, you should see this

```
ls
>sra_metadata  untrimmed_fastq
```

We can make the ls output more comprehensible by using the flag -F, which tells ls to add a trailing / to the names of directories

```
ls -F
>sra_metadata/  untrimmed_fastq/
```

- Anything with a "/" after it is a directory. Things with a "*" after them are programs. If there are no decorations, it's a file.
- ls has lots of other options- they are also called flags. To find out what they are, we can type:

```
man ls
```

- I dont use man. It is much quicker to google stuff. Show biostars and stackoverflow for doing the same thing

QUESTION1

- Use the -l option for the ls command to display more information for each item in the directory.
- What is one piece of additional information this long format gives you that you don't see with the bare ls command? ### ANSWER

```
ls -l
>total 8
>drwxr-x--- 2 dcuser dcuser 4096 Jul 30  2015 sra_metadata
>drwxr-xr-x 2 dcuser dcuser 4096 Nov 15  2017 untrimmed_fastq
```

- The additional information given includes the name of the owner of the file, when the file was last modified, and whether the current user has permission to read and write to the file.

Most commands take the options, which is a dash- followed by a letter. Or many letters for multiple options:

```
ls -ltrh
>total 184
>-rw-r--r--@ 1 jeffreymiller  staff    42K Nov 15  2017 SRR098026.fastq
>-rw-r--r--@ 1 jeffreymiller  staff    46K Nov 15  2017 SRR097977.fastq
```

Lets again change directories to work on our untrimmed_fastq sequences. Once you are in the untrimmed directory, what is in there?

```
ls
> sra_metadata    untrimmed_fastq
cd untrimmed_fastq
ls -F
> SRR097977.fastq  SRR098026.fastq
```

The next command you'll memorize by using it a bunch is head. Head gives you a peek at the first few lines of a file.

```
head SRR097977.fastq

>@SRR098026.1 HWUSI-EAS1599_1:2:1:0:968 length=35
NNNNNNNNNNNNNNNNCNNNNNNNNNNNNNNNNNN
+SRR098026.1 HWUSI-EAS1599_1:2:1:0:968 length=35
!!!!!!!!!!!!!!!!#!!!!!!!!!!!!!!!!!!
@SRR098026.2 HWUSI-EAS1599_1:2:1:0:312 length=35
NNNNNNNNNNNNNNNNANNNNNNNNNNNNNNNNNN
+SRR098026.2 HWUSI-EAS1599_1:2:1:0:312 length=35
!!!!!!!!!!!!!!!!#!!!!!!!!!!!!!!!!!!
@SRR098026.3 HWUSI-EAS1599_1:2:1:0:570 length=35
NNNNNNNNNNNNNNNNANNNNNNNNNNNNNNNNNN
```

Shortcut: Tab Completion

It was time consuming to type that exactly.

- There is a handy feature called 'automatic tab completion'.
- To use it, we hit tab to bring up the options on the file that terminal thinks you are going to use. So, if we hit tab once, we get

```
head S<tab>
>head SRR09
```

- then if we hit it once more, it is almost like running 'ls'

```
head SRR09
>SRR097977.fastq  SRR098026.fastq
```

- this works for files- and thats what most use it for- but it also works for the commands
- if we start typing 'pw' we get stuff that starts with that

The key points here are:

- The shell gives you the ability to work more efficiently by using keyboard commands rather than a GUI.
- Useful commands for navigating your file system include: ls, pwd, and cd.
- Most commands take options (flags) which begin with a -.
- Tab completion can reduce errors from mistyping and make work more efficient in the shell.

Navigating Files and Directories

- How can I perform operations on files outside of my working directory?

- What are some navigational shortcuts I can use to make my work more efficient?

Objectives:

- Use a single command to navigate multiple steps in your directory structure, including moving backwards (one level up).
- Perform operations on files in directories outside your working directory.
- Work with hidden directories and hidden files.
- Interconvert between absolute and relative paths.
- Employ navigational shortcuts to move around your file system.

More navigation

- We got an idea for moving around using cd and the name of the folder to move into.
- But how to we go back out? We dont see the folder we are in.
- We have a special command to tell the computer to move us back or up one directory level.

We have a special command to tell the computer to move us back or up one directory level.

- we use 'cd ..'

```
cd ../
ls
>sra_metadata    untrimmed_fastq
```

- That seems pretty slow, if you have to type cd every time if your data is like ten folders deep.
- lets go back into 'untrimmed_fastq'

```
cd untrimmed_fastq
```

- To navigate out two folders from here, we can double it up with

```
cd ../../
ls
> lab1_notes.md            lab1_notes.sh            pre-lab-instructions.md  shell_data
```

Hidden files

- Sometimes, its better to hide files from the command line. Like system files that shouldn't be messed with by the command line user.
- Somewhere in 'shell_data' is a hidden file.
- Mention ls -aF and ls -Fa are the same, but case matters alot

Navigate by multiple directories

- if you are in gen711_lab, navigate one folder out. 'cd ../'

```
pwd
>/Users/jeffreymiller/Desktop/gen711_lab
```

- if you are in 'shell_data', navigate one folder out. 'cd ../../'
- if you are in 'untrimmed_fastq' or '.hidden', navigate two out with 'cd ../../'

The tilda shortcut takes you home

- green yellow esc (~)

```
cd ~
cd Desktop/...
```

- The /, ~, and .. characters represent important navigational shortcuts.
- Hidden files and directories start with . and can be viewed using ls -a.
- Relative paths specify a location starting from the current location, while absolute paths specify a location from the root of the file system.

The / by itself, takes you all the way back to root.

- This is an absolute path:

```
cd /Users/jeffreymiller/Desktop/gen711_lab/shell_data
```

- We have been navigating using relative paths
- You can use absolute paths anywhere, and they will get you there
- relative paths only work if you can see the folder that you want to be in

Relative path

```
untrimmed_fastq/
```

Absolute

```
cd /Users/jeffreymiller/Desktop/gen711_lab/shell_data
```

Working with Files and Directories

cd ~/shell_data/untrimmed_fastq

ls *.fastq