QIIME2 Workshop: Day 1
Introduction to HCC

Exercises: Wildcards and Pipes

- You can use more than one * at a time. How would you run **`wc -l`** on every file with fb in it?

```
wc -l *fb*
```

- Create a folder called **`fastq`** and move all of our fastq files there in a single mv command.

```
mkdir fastq
mv *.fastq ./fastq
```

- How many files are there in the **`fastq`** directory? Hint: use a shell command to count them!

```
ls -l ./fastq | wc -l
```

- Save the output of your previous command to a file named counts.txt

```
ls -l ./fastq | wc -l > counts.txt
```

Schedule

9:00 – 10:30	Introduction to HPC Connecting to the Clusters Navigating in Bash
10:30 – 10:45	Break
10:45 – 12:00	File Manipulation Wildcards and Pipes
12:00 – 1:00	Lunch
1:00 – 2:15	**Writing Reusable Scripts** **Running Applications on the Clusters**
2:15 – 2:30	Break
2:30 – 4:00	Submitting Jobs Transferring Data with Globus

Exercises: Writing Reuseable Scripts

- cd to the **`fastq`** directory from earlier and write a loop to print off the name and top 4 lines of every fastq file in that directory.

  Is there a way to only run the loop on fastq files ending in **`_1.fastq`**?

- Concatenating (i.e. mashing together) variables is quite easy to do. Add whatever you want to concatenate to the beginning or end of the shell variable after enclosing it in **`{}`** characters. For example:

```
FILE=stuff.txt
echo ${FILE}.example
```

  Produces the output:

```
stuff.txt.example
```

  Can you write a script that prints off the name of every file in the **`fastq`** directory with ".processed" added to it?

Exercises: Writing Reuseable Scripts

- cd to the **`fastq`** directory from earlier and write a loop to print off the name and top 4 lines of every fastq file in that directory.

```
for FILE in *.fastq
do
    head -n 4 $FILE
done
```

Is there a way to only run the loop on fastq files ending in **`_1.fastq`**?

```
for FILE in *_1.fastq
do
    head -n 4 $FILE
done
```

Exercises: Writing Reuseable Scripts

- Concatenating (i.e. mashing together) variables is quite easy to do. Add whatever you want to concatenate to the beginning or end of the shell variable after enclosing it in **{ }** characters. For example:

```
FILE=stuff.txt
echo ${FILE}.example
```

Produces the output:

```
stuff.txt.example
```

Can you write a script that prints off the name of every file in the **fastq** directory with ".processed" added to it?

```
for FILE in $(ls)
do
    echo ${FILE}.processed
done
```

Running Applications

- All applications installed on HCC clusters are loaded in individual modules
- Modules dynamically change the user environment through shell variables
  - $PATH
  - $LD_LIBRARY_PATH
- Hierarchical structure
  - If module A is dependent on module B, then module B must be loaded first to load module A
- Typically follows the naming convention software/version
  - Example: python/2.7
- Load using the **`module`** command

Module Commands

Command	What it does
`module avail`	Lists all modules available to be loaded
`module keyword <keyword>`	Search for a module by keyword
`module spider <name>`	Information about a specific module – can also be used to search
`module load <module_name>`	Load module(s) – can load a list of space delimitated modules
`module unload <module_name>`	Unload module(s) – can unload a list of space delimitated modules
`module purge`	Unloads all currently loaded modules
`module list`	Lists all currently loaded modules

For more information

`module --help`

Available software lists for each cluster: Crane Tusker Sandhills

Exercises: Running Applications

- In order to do some comparisons, your collaborator Professor Reddy, wants to analyze his newest samples using the same pipeline used to analyze data from mid 2017. His previous pipeline used samtools version 1.4. Would he be able to run this analysis on the Crane cluster?

- You agree to work with your collaborator to develop a more recent version of his pipeline. What is the most recent version of samtools available?

Exercises: Running Applications (cont.)

- In addition to the BLAST+ alignment package (the **`blast`** module), HCC also has a parallel version of BLAST in the **`mpi-blast`** module. How many other implementations of the BLAST algorithm are available on HCC clusters?

- Research Assistant Sofia has 10,000 sequences she wants to do a BLAST alignment against the human genome. She decides she wants to run it in parallel using the **`mpiblast`**. What modules does she need to load to begin using it?