BTEP course

Alexandra L Emmons Ph.D. & Joe Wu Ph.D.
BTEP/GAU/CCR/NCI/NIH - email ncibtep@mail.nih.gov
Bioinformatics Training and Education Program

Table of Contents

Course overview

Lesson 1

Lesson 2

Lesson 3

Lesson 4

Practice questions

Course Overview

Biowulf is the Unix-based high-performance compute cluster at NIH and houses thousands of bioinformatics analyses prorgrams. While most are used to working with point-and-click operating systems such as Windows or Mac, working in a command-line driven environment such as Biowulf can be intimidating. This course series will help participant overcome fear of working on high-performance computing clusters so that they can start taking advantage of the resources available for their bioinformatics and data science needs.

Course Expectations / Learning Objectives

After this course, participants will be able to

1. Log onto the NIH High Performance Compute Cluster Biowulf
2. Navigate the folder and file (directory) structure on a Unix system
3. Work with very large Next Generation Sequencing (NGS) files on a Unix system
4. Find and load bioinformatics applications that are installed on Biowulf
5. Run interactive, swarm and batch jobs on Biowulf

Course schedule and topical outline

- Lesson 1 (May 16th, 2023):
    - Overview of Unix and Biowulf
    - Logging into Biowulf
- Lesson 2 (May 23rd, 2023):
    - Navigating around the Biowulf directory structure
- Lesson 3 (May 30th, 2023):
    - Working with files and directories
    - Interactive sessions
    - Exploring Next Generation Sequencing data
- Lesson 4 (June 6th, 2023):
    - Bioinformatics applications on Biowulf
    - Submitting batch jobs
        - Swarm
        - Shell script

student account assignment

Lesson 1: Overview and logging on to Biowulf

Learning objectives

After this lesson, participants will be able to

- Provide reasons for learning Unix command line
- Describe Biowulf and why it is useful for NIH researchers
- Know how to obtain a Biowulf account
- Log on to Biowulf
- Find Biowulf help documentation
- Explore the Biowulf user dashboard

Commands that will be discussed

- `ssh`: securely connect to remote computer
- `pwd`: to find present working directory
- `cd`: change into one directory from another

Reasons to use Biowulf

Biowulf is the Unix-based high performance computing system at the National Institutes of Health. Below are reasons for using Biowulf.

- Many bioinformatics programs/tools are written for the Unix operating system
- There 900+ programs/modules installed on Biowulf, including those used for Bioinformatics
- Current, past, and future versions of tools and databases available
- Reproducibility – written scripts/programs keep track of analyses steps
- Big data analysis – can open and work with very large data files
- Compute Power -
    - has over 100,000 processor nodes
    - large storage capacity (30+ petabytes)
    - Globus *(https://hpc.nih.gov/docs/globus/)* for transfer of large data files

Skills learned while working on Biowulf apply to other high performance computing systems.

Warning

Do not store personally identifiable information on Biowulf!

Getting a Biowulf account

Information for obtaining a Biowulf account can be found at https://hpc.nih.gov/docs/accounts.html *(https://hpc.nih.gov/docs/accounts.html)*. The following conditions have to be met for Biowulf staff to grant accounts.

- PI approval
- PI pays $35 a month
- Annual renewal, which also requires PI approval

Biowulf student accounts

Each participant will be assigned a student account.

See here *(https://nih-my.sharepoint.com/:x:/g/personal/fitzgepe_nih_gov/Eb3cc5ZoaWBOjRB_ec49clMBBvNbuYF9XCQ0A1CZjO-HNw?e=bpaHxi)* for student account assignment. Please use this account throughout this course. Enter NIH credentials to see the student account assignment sheet after clicking the link.

Connecting to Biowulf

To sign onto Biowulf, the users must be connected to the NIH network either by being on campus or through the VPN.

Windows 10 or above users will need to open the Command Prompt (type `cmd` in the Windows search box) while Mac users will need to open the Terminal application (type terminal in Spotlight search).

Once the Windows Command Prompt or Mac Terminal is opened, type the following to connect to Biowulf, where `ssh` is the command used to securely connect to a remote computer. Replace username with the assigned student account ID.

```
ssh username@biowulf.nih.gov
```

Note

For those who already have a Biowulf accounts or will obtain one in the future, use NIH username and password to connect.

Hit enter to supply the password.

The following message appears for those logging on to Biowulf for the first time. Respond with "yes" to proceed.

> The authenticity of host 'biowulf.nih.gov (128.231.2.9)' can't be established. ECDSA key fingerprint is SHA256:BoP/KLS17g+gUuQ7mrCHa9oPPO+MHi/h8WML44iA1dw. Are you sure you want to continue connecting (yes/no)? yes

Once logged onto Biowulf, type the following to see the present working directory.

```
pwd
```

Users will start at the home directory upon signing onto Biowulf. Again, replace username with the student account ID for this class.

```
/home/username
```

The following command will take the user to the data directory. More on the home and data directories in Lesson 2.

```
cd /data/username
```

Biowulf website

The Biowulf webiste (https://hpc.nih.gov *(https://hpc.nih.gov)*) has a menu that allows users to find useful information regarding the cluster (see Figure 1). Some of the useful tabs in this menu are discussed below.

Figure 1: Menu on the Biowulf website.

Applications

Need to find out what software are available on Biowulf? Then click on the Applications tab (Figure 2). The softwares are classified according to discipline.

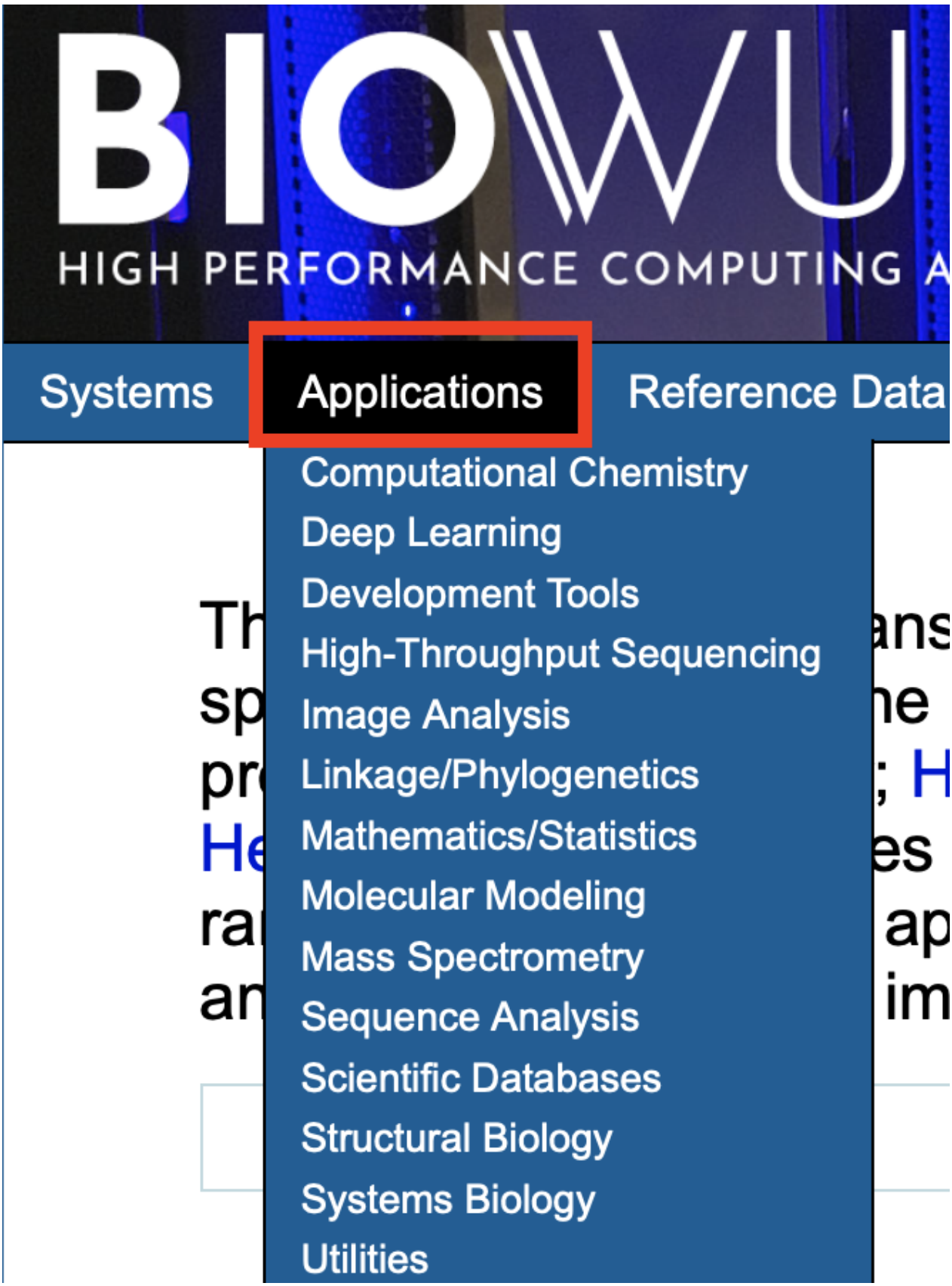
BIOWU
HIGH PERFORMANCE COMPUTING A
Systems
Applications
Reference Data
Computational Chemistry
Deep Learning
Development Tools
High-Throughput Sequencing
Image Analysis
Linkage/Phylogenetics
Mathematics/Statistics
Molecular Modeling
Mass Spectrometry
Sequence Analysis
Scientific Databases
Structural Biology
Systems Biology
Utilities

Figure 2:

Reference data

Analysis of NGS data requires reference genomes and annotations. Click on the Reference Data tab to see which are installed on Biowulf.

Training

Biowulf staff offer extensive trainings. To see what is available, click on the Training tab (Figure 3).

Figure 3: Find training offered by Biowulf staff

User dashboard

The User Dashboard provides

- Account information including group affiliations
- Disk usage and link to increase storage quota for user's data directory
- Information on submitted jobs
- Usage report

There is also a student dashboard *(https://hpcnihapps.cit.nih.gov/auth/)* for the student accounts.

Lesson 2 sneak peak

Below are the commands that will be introduced in Lesson 2.

- `ls`: list directory content
- `chmod`: change file and directory permissions
- `pwd`: get present working directory
- `cd`: change directory
- `cp`: copy

- `rm`: delete

Lesson 2: Biowulf directory structure

Quick review

Lesson 1 introduced the benefits of working on Biowulf and method for connecting to Biowulf from a personal computer by using the `ssh` command.

Learning objectives

After this lesson, participants should be able to

- Understand the Biowulf directory structure
- Describe the home and data directories on Biowulf
- Find help for Unix commands
- List directory content
- Describe file and folder permissions
- Change into one directory from another
- Copy files and folders
- Download files from the web
- Unpack tar files
- View file contents

Commands that will be discussed

- `ls`: list directory content
- `chmod`: change file and directory permissions
- `pwd`: get present working directory
- `cd`: change directory
- `cp`: copy
- `zcat`: to view compressed files
- `cat`: to view file content

Before getting started

Sign onto Biowulf using the assigned student account. Remember, Windows users will need to open the Command Prompt and Mac users will need to open the Terminal. Also remember to connect to the NIH network either by being on campus or through VPN before attempting to sign in. The command to sign in to Biowulf is below, where username should be replaced by the student ID.

```
ssh username@biowulf.nih.gov
```

See here *(https://nih-my.sharepoint.com/:x:/g/personal/fitzgepe_nih_gov/Eb3cc5ZoaWBOjRB_ec49cIMBBvNbuYF9XCQ0A1CZjO-HNw?e=bpaHxi)* for student account assignment. Enter NIH credentials to see the student account assignment sheet after clicking the link.

Biowulf directory path structure

The first step to navigating around Biowulf is to understand the directory path structure. The root folder is the top level folder in the Biowulf file system (Figure 1). In Unix systems, the root folder is designated by "/". Inside the root are the home and data directories. As an example, the data directory contains a folder P, which contains folders P_in and P_out (Figure 1).

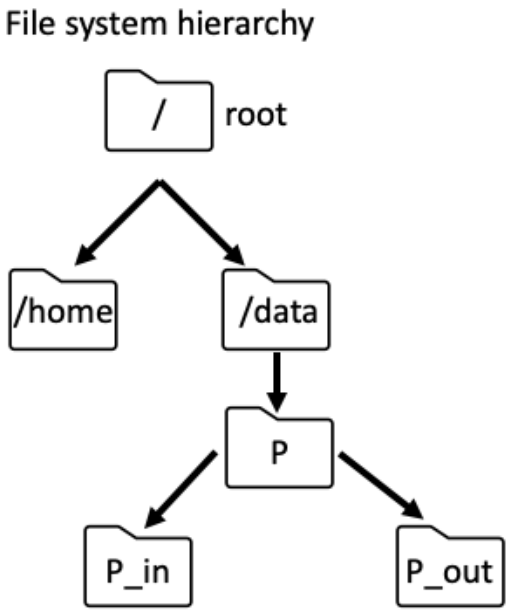

Figure 1: Example of Biowulf file system hierarchy.

Listing the contents of the root folder in Biowulf shows that the home and data directories reside within it. To list directory content, use the `ls` command followed by the name of the directory (in this case /, which denotes the root folder).

```
ls /
```

```
home
data
```

Home and data directories

Upon signing onto Biowulf, the prompt below appears (replace username with the assigned student ID). The ~ at the prompt denotes the home directory. The `pwd` command can be used to print the path to the present working directory, which should be home.

```
[username@biowulf ~]$
```

```
pwd
```

```
/home/username
```

The home directory is limited to 16 gb of storage space and cannot be expanded. Use the data directory, which has more default storage space *(https://hpc.nih.gov/storage/index.html)* and can be expanded when for data intensive analysis. . To change into the data directory use the `cd` command followed by the name of the folder (in this case it is /data/username). Replace username with the student account ID.

```
cd /data/username
```

The `pwd` command can confirm that change of directory was successful.

```
pwd
```

```
/data/username
```

> **Note**
>
> When `pwd` is used, the directory path retrieved starts at the "/" or the root. For instance, /home/username and /data/username. A directory path that starts from the root is known as an absolute path.

Finding help

To get help with Unix commands, use the `man` command, which pulls up the manual. Another option, which is command specific is to append either `-h` or `--help` to the command.

```
man man
```

```
man pwd
```

Hit q to exit the manual.

```
man ls
```

OR

```
ls --help
```

Viewing detailed directory content

Take a look at the /data/classes/BTEP folder using `ls -l` where the `-l` option lists directory contents in the detailed form

```
ls -l /data/classes/BTEP
```

Among the items in /data/classes/BTEP is a folder called unix_on_biowulf_2023_documents. The permission block (ie. the string drwxrwsr-x) for unix_on_biowulf_2023_documents begins with a "d", which denotes that it is a folder. A "-" at the beginning of block indicates a file. Figure 2 explains the permission block. File and folder permissions are important in Unix because it determines who can view and modify content.

```
drwxrwsr-x.  4 wuz8        GAU               4096 Feb  9 21:28 unix_
```

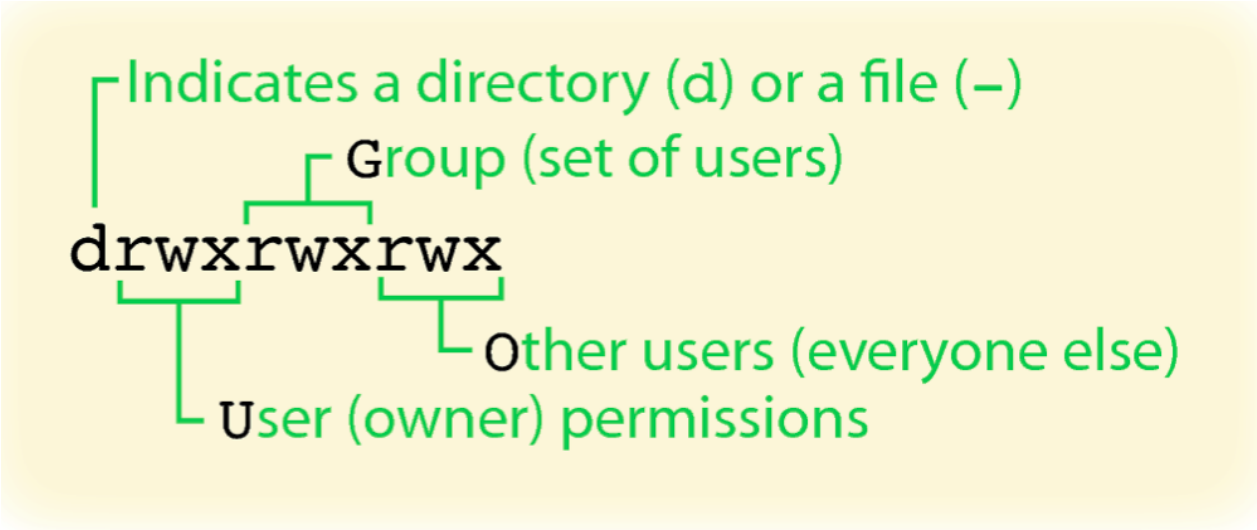

Figure 2: Unix permission block. The permissions are divided into three chunks of "rwx", corresponding to read, write, and execution privileges of the file or directory for owner, others in the group, and everyone else. If the permission begins with "d", then we are looking at a directory. If the permission begins with "-", then we are looking at file. Source: UF Research Computing *(https://training.it.ufl.edu/media/trainingitufledu/documents/research-computing/RC_UpAndRunning.pdf)*.

The `chmod` command enables users to change file and folder permissions. To learn how to use this command use one of the following.

```
chmod --help
```

OR

```
man chmod
```

Copying a directory

For this part of the class, be sure to stay in the /data/username folder. If unsure, use `pwd` to check and use `cd /data/username` to change into if not in the folder.

Copy the unix_on_biowulf_2023_documents in /data/classes/BTEP to /data/username using the following `cp` command construct where the options and arguments are as follows.

- Option: `-r` indicates to copy a folder
    - `-r`: copy directories recursively
- Argument: name of of the folder to be copied (ie. /data/classes/BTEP/unix_on_biowulf_2023_documents)

- Argument: destination to copy the folder to (ie. /data/username, again, replace username with the assigned student ID)

```
cp -r /data/classes/BTEP/unix_on_biowulf_2023_documents /data/usernar
```

Note

The present working directory can be denoted by "."

Now, change into the unix_on_biowulf_2023_documents and look at the content using `ls -l`, which shows two files and two folders.

```
cd unix_on_biowulf_2023_documents
```

```
ls -l
```

```
-rwxr-x---. 1 wuz8 wuz8   368 Sep  5 11:26 SRP045416.swarm
drwxr-x---. 2 wuz8 wuz8  4096 Sep  5 11:26 SRR1553606
drwxr-x---. 2 wuz8 wuz8  4096 Sep  5 11:26 unix_on_biowulf_2023
-rwxr-x---. 1 wuz8 wuz8 41734 Sep  5 11:26 unix_on_biowulf_2023.zip
```

To go back to the /data/username folder (ie. one folder up) use `cd` with the `..` notation.

```
cd ..
```

Copying a file

For this exercise, go back to the unix_on_biowulf_2023_documents folder in the data directory.

```
cd unix_on_biowulf_2023_documents
```

Make a copy of SRP045416.swarm and call it SRP045416_copy_1.swarm. To do this use the `cp` command where the arguments are

- File to make a copy of (ie. SRP045416.swarm)
- Name of the copy (ie. SRP045416_copy_1.swarm)

```
cp SRP045416.swarm SRP045416_copy_1.swarm
```

Go back to the data folder by doing

```
cd ..
```

Copy the SRP045416.swarm file in unix_on_biowulf_2023_documents here using the `cp` command where the arguments are

- File to copy (ie. SRP045416.swarm; here the relative path of unix_on_biowulf_2023_documents/SRP045416.swarm to the file is provided)
- Destination to copy the file (ie. "." which denotes here in the current directory)

```
cp unix_on_biowulf_2023_documents/SRP045416.swarm .
```

Note

Relative path is defined as the path related to the present working directory (pwd). It starts at your current directory and never starts with a / ." -- https://www.geeksforgeeks.org/absolute-relative-pathnames-unix/ *(https://www.geeksforgeeks.org/absolute-relative-pathnames-unix/)*

Downloading from the web

Change back to the /data/username directory for this exercise. Replace username with the student account ID.

```
cd /data/username
```

There maybe times when it is necessary to download a data from the web. Use either `wget` or `curl` to download from the web. Here, `curl` will be shown where the options and arguments are

- Option: `-o` to specify filename of the download
- Argument: url for the file (ie. http://genomedata.org/rnaseq-tutorial/practical.tar). Note that the last part of the url (pratical.tar) is the filename but the `-o` option in `curl` enables saving of this file as something else.

```
curl -o hcc1395_fastq.tar http://genomedata.org/rnaseq-tutorial/pract
```

List the directory content after download to confirm that hcc1395_fastq.tar is there.

Unpacking tar files

The hcc1395_fastq.tar is actually known as a tape archive (it has the .tar extension), which is a bundle of files and folders. The `tar` command is used to unpack its contents. The following are options and arguments used in the `tar` command to extract items.

- Options:
- `-x`: extract files from an archive
- `-v`: verbosely list files processed
- `-f`: use archive file or device ARCHIVE
- Argument: name of the file to unpack (ie. hcc1395_fastq.tar)

```
tar -xvf hcc1395_fastq.tar
```

The `tar` command should have unpacked the contents of hcc1395_fastq.tar into the /data/username directory. These are fastq files containing sequences derived from NGS experiment. These fastq files were compressed (.gz extension) to reduce storage space. Many bioinformatics algorithms can take fastq.gz as input so no need to uncompressed these.

```
hcc1395_normal_rep1_r1.fastq.gz
hcc1395_normal_rep1_r2.fastq.gz
hcc1395_normal_rep2_r1.fastq.gz
hcc1395_normal_rep2_r2.fastq.gz
hcc1395_normal_rep3_r1.fastq.gz
hcc1395_normal_rep3_r2.fastq.gz
hcc1395_tumor_rep1_r1.fastq.gz
hcc1395_tumor_rep1_r2.fastq.gz
hcc1395_tumor_rep2_r1.fastq.gz
hcc1395_tumor_rep2_r2.fastq.gz
hcc1395_tumor_rep3_r1.fastq.gz
hcc1395_tumor_rep3_r2.fastq.gz
```

Viewing file content

Stay in the /data/username folder and take a look at hcc1395_normal_rep1_r1.fastq.gz using the command `zcat`, which is used to view compressed files.

```
zcat hcc1395_normal_rep1_r1.fastq.gz
```

FASTQ file store the sequencing reads derived from Next Generation Sequencing. Each read is composed of four lines.

1. Metadata header that starts with @
2. The actual sequence
3. "+"
4. Quality score of each base in that read

```
@K00193:38:H3MYFBBXX:4:1101:10003:44458/1
TTCCTTATGAAACAGGAAGAGTCCCTGGGCCCAGGCCTGGCCCACGGTTGTCAAGGCACATCATTGCCA
+
AAFFFKKKKKKKKKKKKKKKKKKKKKKKKFKKFKKKKF<AAKKKKKKKKKKKKKKKKFKKKFKKKKKKI
```

Hit `control c` to exit `zcat`

Change into the unix_on_biowulf_2023_documents folder.

If `pwd` is /data/username then do the following. Remember to replace username with the student account ID.

```
cd unix_on_biowulf_2023_documents
```

If `pwd` is not /data/username then do the following

```
cd /data/username/unix_on_biowulf_2023_documents
```

To look at file content, use the `cat` command.

```
cat SRP045416.swarm
```

```
#SWARM --job-name SRP045416
#SWARM --sbatch "--mail-type=ALL --mail-user=wuz8@nih.gov"
#SWARM --gres=lscratch:15
#SWARM --module sratoolkit

fastq-dump --split-files -X 10000 SRR1553606
fastq-dump --split-files -X 10000 SRR1553416
fastq-dump --split-files -X 10000 SRR1553417
fastq-dump --split-files -X 10000 SRR1553418
fastq-dump --split-files -X 10000 SRR1553419
```

Note that `cat` can be used to view hcc1395_normal_rep1_r1.fastq.gz if it was uncompressed.

Change into the unix_on_biowulf_2023 folder.

```
cd unix_on_biowulf_2023
```

```
cat text_1.txt
```

```
oranges
blue
bananas
cats
dogs
apple
florida
gators
gainesville
alachua
county
btep
```

Lesson 3: Working with files and directories, interactive sessions, and exploring Next Generation Sequencing data

Quick review

Lesson 2 introduced the Biowulf directory structure and distinguished the difference between the home and data directories. It introduced commands for

- listing the present working directory (`pwd`)
- listing directory content in short format (`ls`)
- listing directory content in detailed format (`ls -l`)
- moving from one directory to another (`cd`)
- copying (`cp -r`) of a folder.

Learning objectives

After this lesson, the learner should be able to

- Make new directories
- Move and rename files and folders
- Delete files
- Search for patterns in files
- Describe the difference between the Biowulf log-in node and compute nodes
- Request an interactive session
- Explore next generation sequencing data using Unix commands

Commands that will be discussed

- `mkdir`: make new directory
- `mv`: move or rename file or directory
- `rm`: delete
- `sinteractive`: request an interactive session
- `head`: view content at the beginning of a file
- `tail`: view content at the end of a file
- `less`: page through a file
- `grep`: search for pattern in a file

- `wc`: word count

Before getting started

Sign onto Biowulf using the assigned student account. Remember, Windows users will need to open the Command Prompt and Mac users will need to open the Terminal. Also remember to connect to the NIH network either by being on campus or through VPN before attempting to sign in. The command to sign in to Biowulf is below, where username should be replaced by the student ID.

```
ssh username@biowulf.nih.gov
```

See here *(https://nih-my.sharepoint.com/:x:/g/personal/fitzgepe_nih_gov/Eb3cc5ZoaWBOjRB_ec49clMBBvNbuYF9XCQ0A1CZjO-HNw?e=bpaHxi)* for student account assignment. Enter NIH credentials to see the student account assignment sheet after clicking the link.

After connecting to Biowulf, change into the data directory. Again, replace username with the student account ID.

```
cd /data/username
```

Biowulf does not keep data in the student accounts after class, so copy the folder unix_on_biowulf_2023_documents in /data/classes/BTEP to the present working directory, which should be /data/username.

```
cp -r /data/classes/BTEP/unix_on_biowulf_2023_documents .
```

Change into unix_on_biowulf_2023_documents.

```
cd unix_on_biowulf_2023_documents
```

Next, change into unix_on_biowulf_2023.

```
cd unix_on_biowulf_2023
```

Make a new directory called lesson3

```
mkdir lesson3
```

Moving files

Move the file counts.csv to lesson3 using the `mv` command, where the arguments are

- item to move (ie. the file counts.csv)
- destination (ie. the folder lesson3)

```
mv counts.csv lesson3
```

Move the counts.csv file from the folder lesson3 back to the present working directory, which should be /data/username/unix_on_biowulf_2023_documents/unix_on_biowulf_2023.

```
mv lesson3/counts.csv .
```

Moving folders

The `mv` command can also be used to move a folder to another. To demonstrate this, make a copy of the folder lesson3 and call it lesson3_copy.

```
cp -r lesson3 lesson3_copy
```

Move lesson3_copy to lesson3

```
mv lesson3_copy lesson3
```

Renaming files

Rename results.csv to deg_results.csv.

```
mv results.csv deg_results.csv
```

Starting an interactive session

Upon logging on to Biowulf, the user is taken to a log in node, which should not be used for computation intensive tasks. To perform computation intensive tasks, the user should work on a compute node.

To request an interactive session

```
sinteractive
```

The `jobhist` command can be used to look at compute allocations for an interactive session. The argument for `jobhist` is the job id, which can be referenced using the variable SLURM_JOBID. Note that "$" is used to reference variables in Unix.

```
jobhist $SLURM_JOBID
```

```
JobId              : 65090666
User               : wuz8
Submitted          : 20230509 15:40:56
Started            : 20230509 15:45:46
Ended              :

Jobid        Partition       State  Nodes  CPUs      Walltime       F
65090666    interactive     RUNNING      1     2       8:00:00
```

Note

The default sinteractive allocation is 1 core (2 CPUs) and 0.768 GB/CPU of memory and a walltime of 8 hours. Resource allocations can be adjusted depending on the task.

Working with next generation sequencing files

Go back up one directory to the unix_on_biowulf_2023_documents folder and then change into the SRR1553606 folder.

```
cd ..
```

```
cd SRR1553606
```

There are two next generation sequencing data files (fastq files) in this folder.

```
ls
```

```
SRR1553606_1.fastq  SRR1553606_2.fastq
```

It is possible to use Unix commands to learn about the content of fastq files prior to analyzing them with more advanced tools that are available on Biowulf.

The `head` command will print out the first 10 rows (default) of files in Unix and can be applied to fastq files.

```
head SRR1553606_1.fastq
```

```
ATACACATCTCCGAGCCCACGAGACCTCTCTACATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAAC/
+SRR1553606.1 1 length=101
@@@FDFDFHHHHHIJGIJJHHIIJIGHGHIJGFI9DDFH?FFHIGGGH>EHGIJEECCABBDABD###/
@SRR1553606.2 2 length=101
CAACAACAACACTCATCACCAAGATACCGGAGAAGAGAGTGCCAGCAGCGGGAAGCTAGGCTTAATTA(
+SRR1553606.2 2 length=101
CCCFFFFFHHGHHJJIJJJJIJJIDIJJJJHIGIJJJJIFHIJJJJJIJJJJGFFFDEEEEDDDDEEE(
@SRR1553606.3 3 length=101
CTTGCATACTGCACTGGATTGAATTGCGGGACGGTCTGGATCGTCAGGCGCTCGATATTCCACGCTGC(
```

To print out more or less than the default 10 lines, use the `-n` option followed by the number of lines desired. The `head` command below will print the first sequence of the fastq file.

```
head -n 4 SRR1553606_1.fastq
```

```
@SRR1553606.1 1 length=101
ATACACATCTCCGAGCCCACGAGACCTCTCTACATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAAC/
+SRR1553606.1 1 length=101
@@@FDFDFHHHHHIJGIJJHHIIJIGHGHIJGFI9DDFH?FFHIGGGH>EHGIJEECCABBDABD###/
```

The `tail` command can be used to view contents at the end of documents. By default it shows the last 10 lines and the `-n` option can be used to change the default behavior.

```
tail SRR1553606_1.fastq
```

There is also the `less` command, which allows for paging up and down through file contents. Hit the up and down arrow keys to scroll up and down line by line or hit the space bar to scroll down page by page. Hit `q` to exit the `less` command.

```
less SRR1553606_1.fastq
```

The `grep` command is used to search for patterns with in files. For instance, we can search for the sequencing data header that corresponds to every sequence in a fastq file.

```
grep @SRR1553606 SRR1553606_1.fastq
```

Because the each sequence in a fastq file has a header line, it follows that searching for and then counting the occurence of the header lines is a plausible way to obtain the number of sequences. Again, `grep` can be used to search for the header and then the pipe or "|" can be used to send the results to `wc -l` to count the number of lines. The `wc` command can be used to count number of characters and words in a file in addition to the number of lines.

```
grep @SRR1553606 SRR1553606_1.fastq | wc -l
```

Deleting files or folders

Go back up one folder to unix_on_biowulf_2023_documents.

```
cd /data/username/unix_on_biowulf_2023_documents
```

Make a copy of SRP045416.swarm and call it SRP045416_copy_1.swarm

```
cp SRP045416.swarm SRP045416_copy_1.swarm
```

Delete SRP045416_copy_1.swarm.

```
rm SRP045416_copy_1.swarm
```

To remove folders, use `rm` with the `-r` option.

Lesson 4: Biowulf modules, swarm, and batch jobs

Quick review

The previous lessons have taught participants how to connect to Biowulf and navigagte through the environment.

Learning objectives

After this lesson, participants should be able to

- Find bioinformatics applications that are installed on Biowulf
- Load applications that are installed on Biowulf
- Describe the Biowulf batch system
- Use nano to edit files
- Use swarm to submit a group of commands to the Biowulf batch system
- Submit a script to the Biowlf batch system

Commands that will be discussed

- `module avail`: list available applications on Biowulf
- `module spider`: list available applications on Biowulf
- `module what is`: get application description
- `module load`: load an application
- `nano`: open the Unix text editor to edit files
- `touch`: create a blank text file

Before getting started

Sign onto Biowulf using the assigned student account. Remember, Windows users will need to open the Command Prompt and Mac users will need to open the Terminal. Also remember to connect to the NIH network either by being on campus or through VPN before attempting to sign in. The command to sign in to Biowulf is below, where username should be replaced by the student ID.

```
ssh username@biowulf.nih.gov
```

See here *(https://nih-my.sharepoint.com/:x:/g/personal/fitzgepe_nih_gov/Eb3cc5ZoaWBOjRB_ec49clMBBvNbuYF9XCQ0A1CZjO-HNw?e=bpaHxi)* for student account assignment. Enter NIH credentials to see the student account assignment sheet after clicking the link.

After connecting to Biowulf, change into the data directory. Again, replace username with the student account ID.

```
cd /data/username
```

Biowulf does not keep data in the student accounts after class, so copy the folder unix_on_biowulf_2023_documents in /data/classes/BTEP to the present working directory, which should be /data/username.

```
cp -r /data/classes/BTEP/unix_on_biowulf_2023_documents .
```

Change into unix_on_biowulf_2023_documents.

```
cd unix_on_biowulf_2023_documents
```

Bioinformatics applications on Biowulf

Biowulf houses thousands of applications. To get a list of applications that are available on Biowulf use the `module` command its `avail` subcommand.

```
module avail
```

Use the up and down arrows keys to scroll through the list or use the space bar to scroll one page at a time. Hit `q` to exit the modules list and return to the prompt.

`module spider` also lists applications but displays results in a different format. Hit `q` to exit `module spider`.

Use `module spider` followed by the application name to search for a specific application. For instance, fastqc, which is used to assess quality of Next Generation Sequencing data.

```
module spider fastqc
```

Biowulf keeps the current version and previous versions of an application. The default is to load the current version. By default, Biowulf loads the latest version of a tool.

```
-------------------------------------------------------------------------
  fastqc:
-------------------------------------------------------------------------
     Versions:
        fastqc/0.11.8
        fastqc/0.11.9

-------------------------------------------------------------------------
  For detailed information about a specific "fastqc" package (includ
  Note that names that have a trailing (E) are extensions provided by
  For example:

     $ module spider fastqc/0.11.9
-------------------------------------------------------------------------
```

To find out how to load FASTQC

```
module spider fastqc/0.11.9
```

```
-------------------------------------------------------------------------
  fastqc: fastqc/0.11.9
-------------------------------------------------------------------------

    This module can be loaded directly: module load fastqc/0.11.9

    Help:
      This module sets up the environment for using fastqc.
```

To find out what FASTQC does

```
module whatis fastqc
```

```
fastqc/0.11.9       : fastqc: It provide quality control functions to
fastqc/0.11.9       : Version: 0.11.9
```

Working with Biowulf bioinformatics applications

This exercise will demonstrate how to use a Biowulf bioinformatics application called seqkit. The skills can be used for running other applications that are available on Biowulf.

What is seqkit?

```
module whatis seqkit
```

```
seqkit/2.1.0        : A cross-platform and ultrafast toolkit for FAS
```

Before doing anything computationally intensive, request an interactive session.

```
sinteractive
```

Load seqkit or any other tool (tools will not load in the login node)

```
module load seqkit
```

Change into the SRR1553606 directory

```
cd SRR1553606
```

```
ls
```

There are two Next Generation Sequencing fastq files in this folder

```
SRR1553606_1.fastq  SRR1553606_2.fastq
```

Use the `stat` subcommand of seqkit to get some statistics about the SRR1553606_1.fastq.

```
seqkit stat SRR1553606_1.fastq
```

```
file                format  type  num_seqs    sum_len  min_len  avg_
SRR1553606_1.fastq  FASTQ   DNA     10,000  1,010,000      101      
```

Convert SRR1553606_1.fastq to fasta using the `fq2fa` subcommand of seqkit.

```
seqkit fq2fa SRR1553606_1.fastq
```

```
>SRR1553606.1 1 length=101
ATACACATCTCCGAGCCCACGAGACCTCTCTACATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAAC/
>SRR1553606.2 2 length=101
CAACAACAACACTCATCACCAAGATACCGGAGAAGAGAGTGCCAGCAGCGGGAAGCTAGGCTTAATTA(
```

Submitting jobs to the Biowulf batch system

For this portion of the class, change back to the /data/username folder

```
cd /data/username
```

Then make a new directory called SRP045416 and change into it.

```
mkdir SRP045416
```

```
cd SRP045416
```

In Biowulf, a swarm script can help with parallelization of tasks such as downloading multiple sequencing data files from the NCBI SRA study Zaire ebolavirus sample sequencing from the 2014 outbreak in Sierra Leone, West Africa *(https://trace.ncbi.nlm.nih.gov/Traces/?view=study&acc=SRP045416)* in parallel, rather than one file after another. The example here will download the first 10000 reads the following sequencing data files in this study.

- SRR1553606
- SRR1553416
- SRR1553417
- SRR1553418
- SRR1553419

Create up a file called SRP045416.swarm in the nano editor

```
nano SRP045416.swarm
```

Copy and paste the following script into the editor.

```
#SWARM --job-name SRP045416
#SWARM --sbatch "--mail-type=ALL --mail-user=username@nih.gov"
#SWARM --partition=student
#SWARM --gres=lscratch:15
#SWARM --module sratoolkit
```

```
fastq-dump --split-files -X 10000 SRR1553606
fastq-dump --split-files -X 10000 SRR1553416
fastq-dump --split-files -X 10000 SRR1553417
fastq-dump --split-files -X 10000 SRR1553418
fastq-dump --split-files -X 10000 SRR1553419
```

In the swarm script above, the first four lines in the script start with #SWARM are not run as part of the script and are directives for requesting resources on Biowulf. The four swarm directives are interpreted as below:

- --job-name
  - assigns job name (ie. SRP045416)
- --sbatch "--mail-type=ALL --mail-user=username@nih.gov"
  - asks Biowulf to email all job notifications (replace username with NIH username)
- --gres
  - asks for generic resource (ie. local temporary storage space of 15 gb by specifying lscratch:15)
- --module
  - loads modules (ie. sratoolkit which houses `fastq-dump` for downloading sequencing data from the Sequence Read Archive)

After editing a file using nano, hit control-x to exit. When prompted to save, choose hit "y" to save.

To submit SRP045416.swarm

```
swarm -f SRP045416.swarm
```

Use `sjob` to check job status and resource allocation. Figure 1 shows the information provided by `sjob` when SRP045416.swarm was submitted.

```
sjobs
```

Some important columns in Figure 1 include the following.

- JobID
- St, which provides the job status
  - R for running
  - PD for pending
- Walltime, which indicates how much time was allocated for the job
- Number of CPUs and memory assigned

Note that the swarm script was assigned job ID 1436172 and there are five sub-jobs as indicated by [0-4], which concords with the five commands in the script. Biowulf assigned 5 cpus (see cpus queued) and 7.5 gb of memory or 1.5 gb per sub-job (see mem queued) for the swarm script.

```
User  JobId          JobName     Part         St  Reason  Runtime     Walltime     Nodes  CPUs   Memory  Dependency  Nodelist
=============================================================================================================================
wuz8  1428430        sinteracti  interactive  R               59:16      8:00:00      1      2    2 GB              cn4282
wuz8  1436172_[0-4]  SRP045416   norm         PD                         2:00:00            5    2 GB
=============================================================================================================================
cpus queued  = 5
cpus running = 0 / 2
mem  queued  = 7.5 GB
mem  running = 2.0 MB / 1.5 GB
jobs queued  = 5
jobs running = 1
```

Figure 1: Use `sjobs` to check status and resource allocation after submitting a job to Biowulf.

After the swarm script finishes, use `ls` to list the contents of the directory. Use the `-1` option to show one item per line.

```
ls -1
```

There are swarm log files with *.e and* .o extensions. Importantly, the fastq files were downloaded.

```
SRP045416_65452913_0.e
SRP045416_65452913_0.o
SRP045416_65452913_1.e
SRP045416_65452913_1.o
SRP045416_65452913_2.e
SRP045416_65452913_2.o
SRP045416_65452913_3.e
SRP045416_65452913_3.o
SRP045416_65452913_4.e
SRP045416_65452913_4.o
SRP045416.swarm
SRR1553416_1.fastq
SRR1553416_2.fastq
SRR1553417_1.fastq
SRR1553417_2.fastq
SRR1553418_1.fastq
SRR1553418_2.fastq
SRR1553419_1.fastq
SRR1553419_2.fastq
SRR1553606_1.fastq
```

An advantage of using command line and scripting to analyze data is the ability to automate, which is desired when working with multiple input files such as fastq files derived from

sequencing experiments. A bash script can help obtain stats using seqkit for the fastq files that were just downloaded. Create a script called SRP045416_stats.sh.

```
nano SRP045416_stats.sh
```

Copy and paste the following into the editor.

```
#!/bin/bash
#SBATCH --job-name=SRP045416_stats
#SBATCH --mail-type=ALL
#SBATCH --mail-user=username@nih.gov
#SBATCH --mem=1gb
#SBATCH --partition=student
#SBATCH --time=00:02:00
#SBATCH --output=SRR045416_stats_log

#LOAD REQUIRED MODULES
module load seqkit

#CREATE TEXT FILE TO STORE THE seqkit stat OUTPUT
touch SRP045416_stats.txt

#CREATE A FOR LOOP TO LOOP THROUGH THE FASTQ FILES AND GENERATE STAT
#Use ">>" to redirect and append output to a file
for file in *.fastq;
do seqkit stat $file >> SRP045416_stats.txt;
done
```

To submit this script

```
sbatch SRP045416_stats.sh
```

```
cat SRP045416_stats.txt
```

```
file                format  type  num_seqs    sum_len  min_len  avg_
SRR1553416_1.fastq  FASTQ   DNA     10,000  1,010,000      101
file                format  type  num_seqs    sum_len  min_len  avg_
SRR1553416_2.fastq  FASTQ   DNA     10,000  1,010,000      101
file                format  type  num_seqs    sum_len  min_len  avg_
SRR1553417_1.fastq  FASTQ   DNA     10,000  1,010,000      101
file                format  type  num_seqs    sum_len  min_len  avg_
SRR1553417_2.fastq  FASTQ   DNA     10,000  1,010,000      101
```

```
file                format  type  num_seqs    sum_len  min_len  avg_
SRR1553418_1.fastq  FASTQ   DNA     10,000  1,010,000      101      1
file                format  type  num_seqs    sum_len  min_len  avg_
SRR1553418_2.fastq  FASTQ   DNA     10,000  1,010,000      101      1
file                format  type  num_seqs    sum_len  min_len  avg_
SRR1553419_1.fastq  FASTQ   DNA     10,000  1,010,000      101      1
file                format  type  num_seqs    sum_len  min_len  avg_
SRR1553419_2.fastq  FASTQ   DNA     10,000  1,010,000      101      1
file                format  type  num_seqs    sum_len  min_len  avg_
SRR1553606_1.fastq  FASTQ   DNA     10,000  1,010,000      101      1
file                format  type  num_seqs    sum_len  min_len  avg_
SRR1553606_2.fastq  FASTQ   DNA     10,000  1,010,000      101      1
```

Explanation of the SRP045416_stats.sh script.

- Lines that start with "#" are comments and are not run as a part of the script
- A shell script starts with #!/bin/bash, where "#!" is known as the sha-bang following "#!", is the path to the command interpreter (ie. /bin/bash)
- Lines that start with #SBATCH are directives. Because these lines start with "#", they will not be run as a part of the script. However, these lines are important because they instruct Biowulf on when and where to send job notification as well as what resources need to be allocated.
    - job-name: (name of the job)
    - mail-type: (type of notification emails to receive, ALL will send all notifications including begin, end, cancel)
    - mail-user: (where to send notification emails, replace with NIH email)
    - mem: (RAM or memory required for the job)
    - partition: (which partition to use; student accounts will need to use the student partition)
    - time: (how much time should be alloted for the job, we want 10 minutes)
    - output: (name of the log file)

Practice questions

Lesson 1

Lesson 1 practice

Name the softwares for Windows and Macs are used to connect to Biowulf.

{{Sdet}}

Solution{{Esum}}

```
Windows users will use the Command Prompt
```

```
Mac users will use the Terminal
```

{{Edet}}

True or False: Connection to the NIH network by being on campus or through VPN is required to log into Biowulf.

{{Sdet}}

Solution{{Esum}}

True

{{Edet}}

What command is used to connect to Biowulf?

{{Sdet}}

Solution{{Esum}}

```
ssh username@biowulf.nih.gov
```

{{Edet}}

What is the Unix command for checking present working directory?

{{Sdet}}

Solution{{Esum}}

```
pwd
```

{{Edet}}

What is the Unix command for changing directory?

{{Sdet}}

Solution{{Esum}}

```
cd
```

{{Edet}}

Lesson 2

Lesson 2 practice

For these practice questions, check the present working directory and if needed, change into the /data/username folder (username is the student account ID).

What command is used to check present working directory?

{{Sdet}}

Solution{{Esum}}

```
pwd
```

{{Edet}}

If not in the /data/username folder, then what is the approach to change into it?

{{Sdet}}

Solution{{Esum}}

```
cd /data/username
```

{{Edet}}

Copy the lesson_2_practice folder from /data/classes/BTEP/unix_on_biowulf_2023_practice_sessions to the /data/username folder, which should be the present working directory.

{{Sdet}}

Solution{{Esum}}

```
cp -r /data/classes/BTEP/unix_on_biowulf_2023_practice_sessions/lesso
```

{{Edet}}

Change into lesson2_practice.

{{Sdet}}

Solution{{Esum}}

```
cd lesson2_practice
```

{{Edet}}

How many files and how many directories are in lesson2_practice?

{{Sdet}}

Solution{{Esum}}

There are four files and no directories.

```
ls -l
```

```
-rw-r----- 1 wuz8 wuz8 19336 May 11 16:34 nc_002549_1.fasta
-rw-r----- 1 wuz8 wuz8 30429 May 11 16:34 nc_0045512_2.fasta
-rw-r----- 1 wuz8 wuz8  1468 May 11 16:34 OK572970_1.fasta
-rw-r----- 1 wuz8 wuz8   265 May 11 16:34 OQ946980_1.fasta
```

{{Edet}}

Take a look at OQ946980_1.fasta, what organism did this sequence come from?

{{Sdet}}

Solution{{Esum}}

```
cat OQ946980_1.fasta
```

```
>OQ946980.1 Severe acute respiratory syndrome coronavirus 2 isolate S
TCTACTCTTGCGCAGAATGAATTCTCGTAACTACATAGCACAAGTAGATGTAGTTAACTTTAATCTCAC
TAGCAATCTTTAATCAGTGTGTAACATTAGGGAGGACGTGAAAGAACCAC
```

{{Edet}}

Change back to /data/username. Again, replace username with the student account ID.

Download the fastq files from http://genomedata.org/rnaseq-tutorial/HBR_UHR_ERCC_ds_5pc.tar *(http://genomedata.org/rnaseq-tutorial/HBR_UHR_ERCC_ds_5pc.tar)*. Save the output as HBR_UHR_ERCC_ds_5pc.tar.

{{Sdet}}

Solution{{Esum}}

```
curl -o HBR_UHR_ERCC_ds_5pc.tar http://genomedata.org/rnaseq-tutoria
```

{{Edet}}

Unpack HBR_UHR_ERCC_ds_5pc.tar, how many fastq files are there?

{{Sdet}}

Solution{{Esum}}

```
tar -xvf HBR_UHR_ERCC_ds_5pc.tar
```

There are 12 fastq files.

```
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
HBR_Rep3_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep3_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
UHR_Rep1_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep1_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
UHR_Rep2_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep2_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
UHR_Rep3_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep3_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
```

{{Edet}}

Lesson 3

Lesson 3 practice

For these practice questions, check the present working directory and if needed, change into the /data/username folder (username is the student account ID).

What command is used to check present working directory?

{{Sdet}}

Solution{{Esum}}

```
pwd
```

{{Edet}}

If not in the /data/username folder, then what is the approach to change into it?

{{Sdet}}

Solution{{Esum}}

```
cd /data/username
```

{{Edet}}

Copy the lesson3_practice folder from /data/classes/BTEP/unix_on_biowulf_2023_practice_sessions to the present working directory, which should be /data/username.

{{Sdet}}

Solution{{Esum}}

```
cp -r /data/classes/BTEP/unix_on_biowulf_2023_practice_sessions/lesso
```

{{Edet}}

Change into the lesson3_practice folder.

{{Sdet}}

Solution{{Esum}}

```
cd lesson3_practice
```

{{Edet}}

How many files and directories are in the lesson3_practice folder?

{{Sdet}}

Solution{{Esum}}

```
ls -l
```

There is one folder and one file.

```
drwxr-x--- 2 wuz8 wuz8 4096 May 11 17:29 sample_sequence_data
-rw-r----- 1 wuz8 wuz8   46 May 11 17:29 text1.txt
```

{{Edet}}

Change into the sample_sequence_data folder. How many files are there?

{{Sdet}}

Solution{{Esum}}

```
cd sample_sequence_data
```

```
ls -l
```

There are 12 fq or fastq files.

```
-rwx------ 1 wuz8 wuz8 29108029 May 11 17:18 HBR_1_R1.fq
-rwx------ 1 wuz8 wuz8 29108029 May 11 17:19 HBR_1_R2.fq
-rwx------ 1 wuz8 wuz8 35553295 May 11 17:18 HBR_2_R1.fq
-rwx------ 1 wuz8 wuz8 35553295 May 11 17:18 HBR_2_R2.fq
-rwx------ 1 wuz8 wuz8 31861169 May 11 17:18 HBR_3_R1.fq
-rwx------ 1 wuz8 wuz8 31861169 May 11 17:19 HBR_3_R2.fq
-rwx------ 1 wuz8 wuz8 55822508 May 11 17:19 UHR_1_R1.fq
-rwx------ 1 wuz8 wuz8 55822508 May 11 17:18 UHR_1_R2.fq
-rwx------ 1 wuz8 wuz8 39860900 May 11 17:18 UHR_2_R1.fq
```

```
-rwx------ 1 wuz8 wuz8 39860900 May 11 17:18 UHR_2_R2.fq
-rwx------ 1 wuz8 wuz8 45524396 May 11 17:19 UHR_3_R1.fq
-rwx------ 1 wuz8 wuz8 45524396 May 11 17:18 UHR_3_R2.fq
```

{{Edet}}

How many sequencing reads are in HBR_1_R1.fq?

{{Sdet}}

Solutions{{Esum}}

```
grep @HWI HBR_1_R1.fq | wc -l
```

There are 118571 sequencing reads.

{{Edet}}

Go back up one directory to the lesson3_practice folder.

{{Sdet}}

Solutions{{Esum}}

```
cd ..
```

{{Edet}}

Rename text1.txt to text_file1.txt.

{{Sdet}}

Solutions{{Esum}}

```
mv text1.txt text_file1.txt
```

{{Edet}}

Lesson 4 Practice

Author: Stephan Sanders, PhD (UCSF)

For today's practice, we are going to embark on a Unix treasure hunt created by the Sanders Lab *(https://sanderslab.github.io/code/)* at the University of California San Francisco. Note: the treasure hunt materials can be obtained directly from the Sanders lab code repository linked above.

UNIX treasure hunt tutorial

This perl script will install a series of directories and clues that teaches basic UNIX command line skills including `cd`, `ls`, `grep`, `less`, `head`, `tail`, and `nano`. Run the perl script from the command line on a UNIX based machine (e.g. Mac or Linux) using the command: `perl treasureHunt_v2.pl`. Then use `ls` to find the first clue. A PDF of command line commands is also available to download.

- Source
- Manual

Note to start at the `/data/username` folder for this exercise (replace username with the student account ID). To begin create a directory called `treasure_hunt` in your home directory and run the perl script in `/data/classes/BTEP/unix_on_biowulf_2023_practice_sessions/lesson4_practice/` from the `treasure_hunt` directory.

{{Sdet}}

Solution{{Esum}}

```
mkdir treasure_hunt
cd treasure_hunt
perl /data/classes/BTEP/unix_on_biowulf_2023_practice_sessions/lesson
ls -l
```

{{Edet}}

Read the first clue and begin.

Recommendation: Create an environment variable to store the path to the treasure hunt directory to facilitate movement through the directory.

{{Sdet}}

Solution{{Esum}}

```
THUNT=`pwd`
echo $THUNT
```

{{Edet}}

When you have found the treasure, answer or do the following:

1. How many words are in the last line of the file containing the teasure?

   {{Sdet}}

   Solution{{Esum}}

   ```
   tail -n 1 openTheBox.txt | wc -w
   ```

   {{Edet}} 2. Save the last line to a new file called `finallyfinished.txt` without copying and pasting.

   {{Sdet}}

   Solution{{Esum}}

   ```
   tail -n 1 openTheBox.txt > finallyfinished.txt
   ```

   {{Edet}}
   3. Now append the first line to the same file that you just saved the last line.

   {{Sdet}}

   Solution{{Esum}}

   ```
   head -n 1 openTheBox.txt >> finallyfinished.txt
   ```

   {{Edet}}

Congratulations! You have found the treasure and have gained some useful unix practice throughout your hunt.