Introduction to Unix on Biowulf - January 2024

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

Student accounts

Lesson 1 (Overview of Unix command line and signing onto Biowulf)

Lesson 2 (Unix command structure and navigating between Biowulf directories)

Lesson 3 (Copy, move, rename, and delete files and folders using Unix commands)

Lesson 4 (Working with bioinformatics software on Biowulf)

Lesson 5 (Submitting shell and swarm scripts to the Biowulf batch system)

Lesson 6 (Working with text files and wrangling tabular data using Unix)

Help Sessions

Connecting to Biowulf (additional methods)

Self learning resources

Course overview

Biowulf is the high-performance compute cluster (HPC) at NIH and runs Linux, a Unix-like operating system. It offers more compute power than a personal computer and has over 900 software applications installed, including those for bioinformatics. Using Biowulf requires working knowledge of a command line interface. This course series will teach the basic Unix commands needed to get started working with your NGS data on Biowulf.

Course Expectations / Learning Objectives

After this course, participants will be able to

- Log onto the NIH High Performance Compute Cluster Biowulf
- Navigate the folder and file (directory) structure on a Unix system
- View directory content
- Copy, move, rename, and delete files and folders
- Find and work with bioinformatics applications that are installed on Biowulf
- Run interactive, swarm and batch jobs on Biowulf
- Work with and perform basic wrangling tasks on text files and tabular data

Course schedule and topical outline

- Lesson 1 (Monday, January 22, 2024):
  - Overview of Unix and Biowulf
  - Logging into Biowulf
  - Lesson 1 recording *(https://cbiit.webex.com/cbiit/ldr.php?RCID=5096751e878bda0321ea0a34339ec38f)*
- Lesson 2 (Wednesday, January 24, 2024):
  - Navigating the Biowulf directory structure
  - Lesson 2 recording *(https://cbiit.webex.com/cbiit/ldr.php?RCID=b000d2ca20316ed3d98fe09118ff0605)*
- Lesson 3 (Monday, January 29, 2024):
  - Copy, move,rename, and deleting files and folders using Unix commands
  - Lesson 3 recording *(https://cbiit.webex.com/cbiit/ldr.php?RCID=e47a62f16895360c70cd5afd43706eff)*
- Lesson 4 (Wednesday, January 31, 2024):
  - Working on interactive sessions
  - File transfer on Helix
  - Working with software installed on Biowulf
  - Lesson 4 recording *(https://cbiit.webex.com/cbiit/ldr.php?RCID=87f9caacc5b3171d47f3ed841b002621)*
- Lesson 5 (Monday, February 5, 2024):
  - Submitting shell and swarm scripts to the Biowulf batch system

    - Lesson 5 recording *(https://cbiit.webex.com/cbiit/ldr.php?RCID=6cf97587736a970861ce065c6f84f1ca)*
- Lesson 6 (Wednesday, February 7, 2024):
    - Using Unix commands to work with and performing basic wrangling tasks for text files and tabular data.
    - Lesson 6 recording *(https://cbiit.webex.com/cbiit/ldr.php?RCID=ff3a620ec00bf1e2975728bdf9285956)*

Participants will use student accounts provided by Biowulf for this course series. See student assignments.

Student accounts

Name	Student account ID
Wu, Zhuoxi	student1
Zhang, Ting	student2
Zhang, Pei	student3
Zargar, Shabir	student4
Yavuz, Bengi	student5
Wu, Xueyao	student6
Wang, Limin	student7
Wang, Kevin	student8
WANG, HERUI	student9
Wagh, Kaustubh	student10
So, Jae Young	student11
Scales, Jessica	student12
Sang, Xueyu	student13
Samdin, Tuan	student14
Sait, Shaimaa	student15
Rahman, Naimur	student16
Nie, Zuqin	student17
Morato Rafet, Sergio	student18
Miraftab, Mona	student19
Martinez Fructuoso, Lucero	student20
Kaul, Sunil	student21
Honec, Rob	student22
Fatema, Kaniz	student23
Fan, Lixin	student24
ezennia, somayina	student25
El Meskini, Rajaa	student26
Cheng, Jason	student27

Name	Student account ID
Chen, Xuemin	student28
Carney, Christine	student29
Brownmiller, Tayvia	student30
Blechter, Batel	student31
BHATT, BHARAT	student32
Betrapally padmanabhan, Naga sridhar	student33
Barry, Madeline	student34
Aljabri, Ashwaq	student35
Agarwala, Neha	student36
	student37
	student38
	student39
	student40

Lesson 1: Overview of Unix command line and signing onto Biowulf

Learning objectives

After this lesson, participants will be able to

- Describe the Unix operating system
- Describe Biowulf
- Connect onto Biowulf via local computer

Overview of Unix

In Windows and MacOS, we interact with the computer through a graphical user interface (GUI). On the contrary, in Unix, we interact with the computer by typing commands.

Basic Unix command syntax

The Unix command syntax is composed of

- The command
- Option(s) that will alter how a command functions
- Argument(s), what you want the command to operate on

```
command options argument
```

For instance, to make a new folder in Unix, we use the command `mkdir` . Here, we enter the command followed by the argument(s) that we want the command to operate on. In this case, the argument is the name of the folder that we would like to create. This is different from the graphical based approach that we use to create new folders in Windows *(https://support.microsoft.com/en-us/office/create-a-new-folder-cbbfb6f5-59dd-4e5d-95f6-a12577952e17)* or MacOS *(https://support.apple.com/guide/mac-help/organize-files-with-folders-mh26885/mac)*

```
mkdir new_folder
```

Above, we just learned our first Unix command, which is just one of many. Before moving further, we should clarify the rationale for using Unix. While there is a steep learning curve, once we have mastered working in Unix, we can perform many of our computing processes. Unix

allows for easy file management, editing of text files, and allows us to view tabular data that is too large for Excel. Further, many of the applications used in bioinformatics are made to work in Unix.

Overview of Biowulf

Biowulf is the high performance and Unix-based computing system at NIH. Below are some rationale for using Biowulf.

- Biowulf offers more computing power and space for data storage compared to our local machine.
- Biowulf also houses many applications for bioinformatics *(https://hpc.nih.gov/apps/)*, which are installed and updated by their staff.
- The GUI-based bioinformatics package, Partek Flow *(https://partekflow.cit.nih.gov)* runs on Biowulf.

Visit https://hpc.nih.gov/docs/accounts.html *(https://hpc.nih.gov/docs/accounts.html)* to learn how to obtain a Biowulf account.

Figure 1 shows the an example of high performance computing clusters hierarchy. This is useful to know so that we know what we are asking for when requesting compute resources.

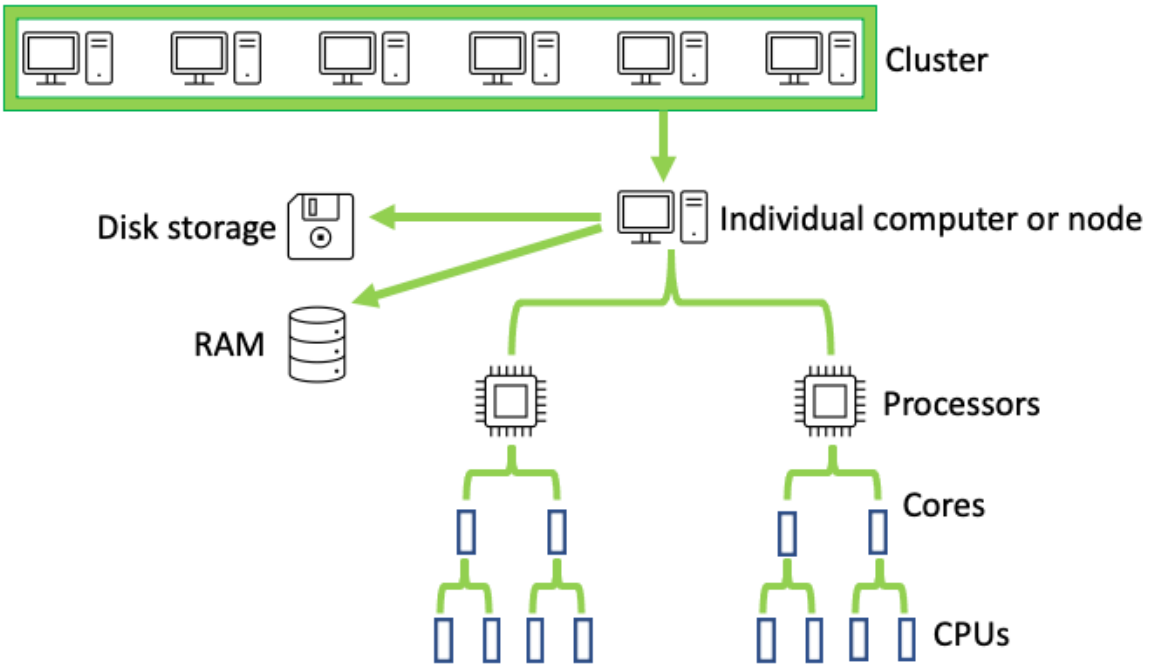


Figure 1: In Biowulf, many computers make up a cluster. Each individual computer or node has disk space for storage and random access memory (RAM) for running tasks. The individual computer is composed of processors, which are further divided into cores, and cores are divided into CPUs. In this example, the individual computer has 2 processors, 4 cores, and 8 CPUs.

Biowulf student accounts

For this course series, participants will be using one of the student accounts (see student assignments) provided by Biowulf staff. See the course overview section student account ID assignment.

Signing onto Biowulf

When working on Biowulf, we are working on a remote computer; thus, we need a way to connect to it. We can use Secure Shell Protocol (ssh) to connect to Biowulf. When connecting to Biowulf, we need to either be connected to the NIH network by being on campus or via VPN.

Signing onto Biowulf with a PC

For those using Windows 10 or newer, ssh is built into the command prompt (Figure 2 and Figure 3).

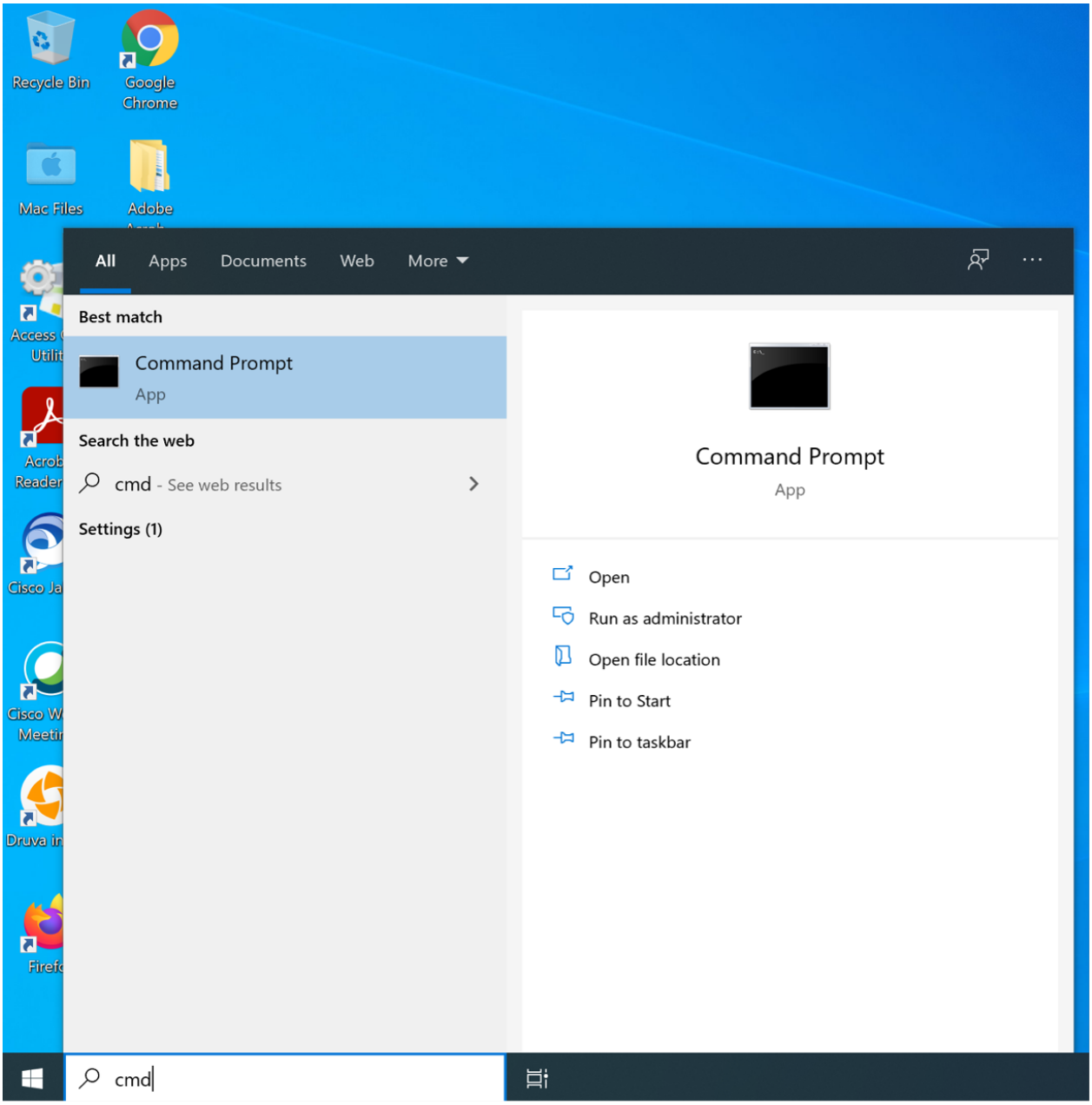


Figure 2: At the search box next to the Windows start menu, type cmd and click on the command prompt application.

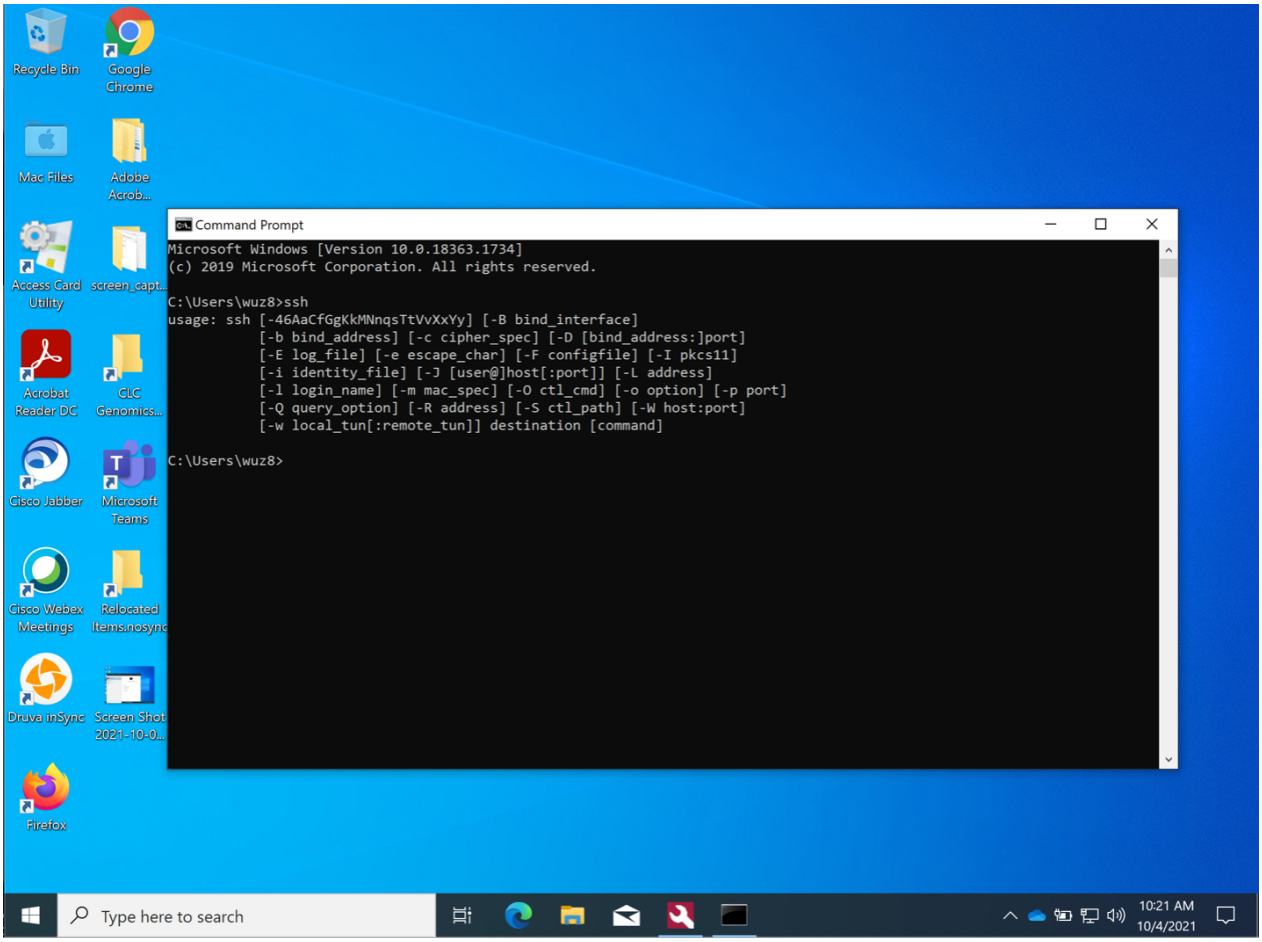


Figure 3: When the command prompt opens, you can type ssh to confirm that it is available

Signing onto Biowulf with a Mac

The best way to sign onto Biowulf from a Mac is to use the built-in terminal (Figure 4). Use the Spot Light search at the Mac menu bar to search for the Terminal application. Click on it to open the Terminal.

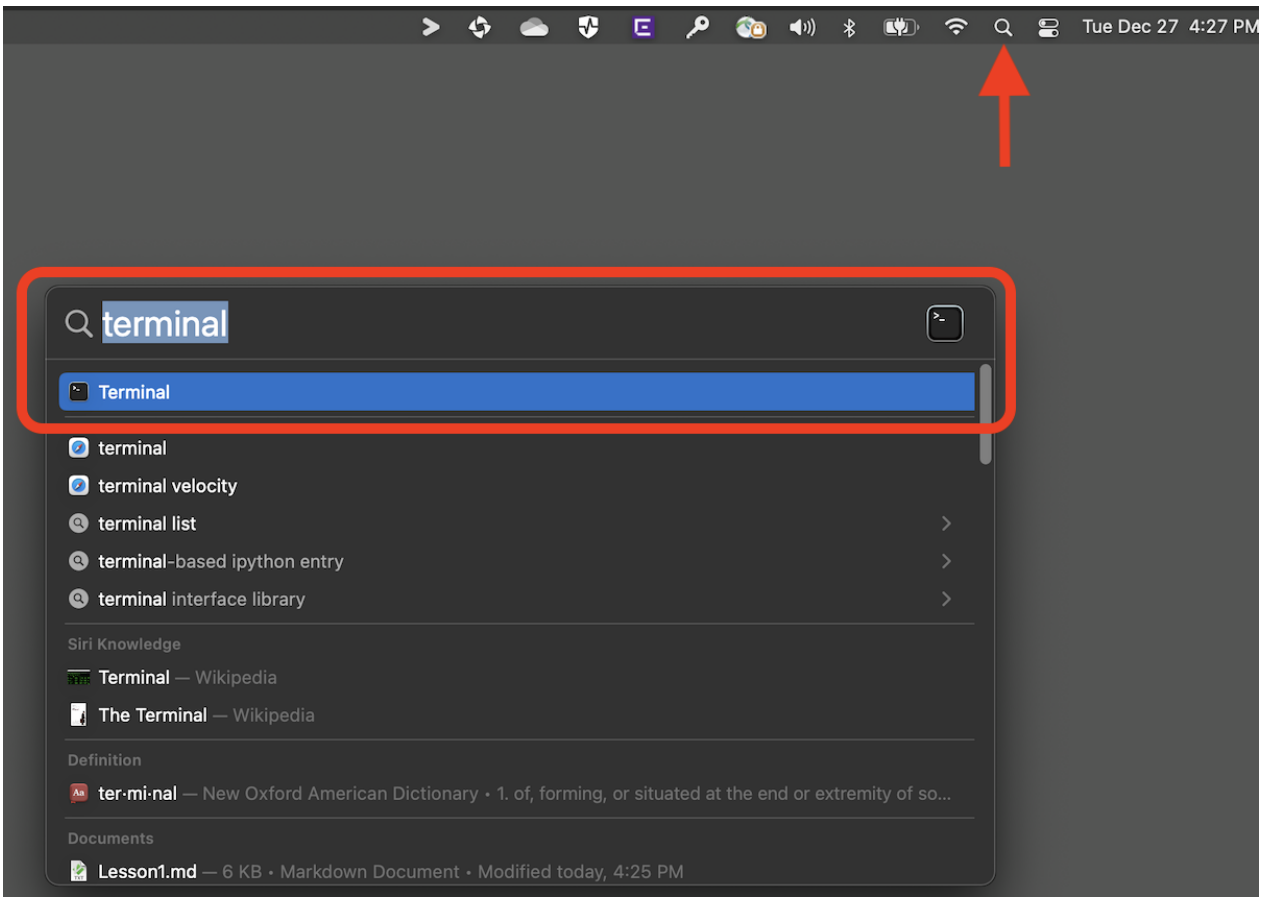


Figure 4: Use the Mac Spot Light search to find the Terminal.

Connect to Biowulf

Remember that if you are not on campus, then you need to connect to the NIH network through VPN. Regardless whether you are using the Windows Command Prompt or Mac Terminal, the construct for ssh to connect to Biowulf is (see Figure 5).

The username in the ssh command is either

- your NIH username if you are using your own Biowulf account for this course **OR**
- one of the student accounts

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

For first time users, when connecting you may see the message below. Respond with yes.

> 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

Next, you will see a message warning you that you are accessing a government computer system and that you should not do anything suspicious. At the end of the message, you will be

asked to enter your password, which is either your NIH password (if you are using your own Biowulf account) or the password for the student accounts. The cursor will not move and nothing will be displayed when entering your password, but keep typing.

```
Warning: Permanently added 'biowulf.nih.gov' (ED25519) to the
list of known hosts.
                          ***WARNING***

You are accessing a U.S. Government information system, which
includes (1) this computer, (2) this computer network, (3) all
computers connected to this network, and (4) all devices
and storage media attached to this network or to a computer on
this network. This information system is provided for U.S.
Government-authorized use only.

Unauthorized or improper use of this system may result in
disciplinary action, as well as civil and criminal penalties.

By using this information system, you understand and consent to the
following:

* You have no reasonable expectation of privacy regarding any
communications or data transiting or stored on this information
system. At any time, and for any lawful Government purpose,
the government may monitor, intercept, record, and search and
seize any communication or data transiting or stored on this
information system.

* Any communication or data transiting or stored on this information
system may be disclosed or used for any lawful Government purpose.

--
Notice to users:  This system is rebooted for patches and
maintenance on the first Sunday of every month at 8:00 pm unless
Monday is a holiday, in which case it is rebooted the following
Sunday evening at 8:00 pm.  Running cluster jobs are not
affected by the monthly reboot.

username@biowulf.nih.gov's password:
```

You will be taken to the prompt after successfully entering your password (see below). It is at the prompt where we type commands and interact with Biowulf. Again, replace username with the student ID in which you were assigned.

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

Finding group affiliation on a high performance computing cluster

The `id` command informs groups that the user might be affiliated with. This is important when collaborating with others Biowulf such that our affiliation with groups will indicate that we have access to the data.

```
id
```

Running the `id` command we see my user id (uid) and primary group id (gid). We also see that I am a part of the GAU and LCP_Omics groups.

```
uid=58740(wuz8) gid=58740(wuz8) groups=58740(wuz8),57888(GAU)
```

Log-in node

Upon signing onto Biowulf, users will land in the log-in node. Later on in this series, compute nodes will be introduced but essentially, the log-in nodes should not be used to perform compute intensive tasks.

> **Definition**
>
> "The log in node is your point of access to the Biowulf cluster" -- Biowulf accounts and log in node *(https://youtu.be/qiWGxrLl6AY?t=207)*

The log in node is meant for the following (Source: Biowulf accounts and log in node *(https://youtu.be/qiWGxrLl6AY?t=217)*)

- Submitting jobs (main purpose)
- Editing/compiling code
- File management
- File transfer
- Brief testing of code or debugging (under 20 minutes)

Biowulf directory spaces

Upon signing onto Biowulf, users will land in the `home` directory, which is denoted by `home/username` or ~. Again, replace username with your assigned student ID or NIH username when you get a personal Biowulf account.

> **Note**
>
> "Each user has a home directory called /home/username which is accessible from every HPC system. The /home area has a quota of 16 GB which cannot be increased. It is commonly used for configuration files (aka dotfiles), code, notes, executables, state files, and caches." -- Biowulf *(https://hpc.nih.gov/storage/)*.

The `pwd` command is used to find which directory the user is currently in.

```
pwd
```

Again, upon log into Biowulf, the current directory should be home.

```
/home/username
```

Data directory

The data directory is much larger and quota can be increased. The path to the data directory is /data/username. To change in to the data directory use the following. The data directory to can be used to store analysis input and output.

```
cd /data/username
```

lscratch

In Biowulf, lscratch is local storage space available on individual nodes. This can be helpful and used for jobs that read or write a lot of temporary files. We will further discuss lscratch in a future lesson.

Scratch

The scratch area is a shared storage space accessible to users for storing temporary files. The path to this is `/scratch/username`. A word of caution is that files in scratch are deleted after 10 days. While each user can store up to 10 TB (terabyte) of data in scratch, it is not guaranteed that this amount will always be available. Finally, Biowulf staff will delete files if scratch becomes more than 80% full.

Snapshots

When working in Unix, we need to keep in mind that there is no Recycling Bin (Windows) or Trash can (Mac) that hold deleted items and allow us to recover it. Once we delete something in Unix, it is gone. Fortunately, Biowulf keeps snapshots, which are read-only copy of data at a certain time and we can use these to restore content that we deleted. See here *(https://hpc.nih.gov/storage/backups.html)* for snapshots on Biowulf. To change in the snapshot directory from the data folder, use the following.

```
cd /data/username/.snapshot
```

The home directory snapshot is located at `/home/username/.snapshot`.

Lesson 2: Unix command structure, navigating Biowulf directories, and tools for data transfer

After this lesson, participants will

- Know how to get help with Unix commands
- Know the tools for transferring data from local computer to the cluster
- Be able to navigate the Unix file systems
- Be able to list directory content
- Be able to describe file and directory permissions as well as know how to modify them

Connecting to Biowulf

To get started, open the Command Prompt (Windows) or the Terminal (Mac) and connect to Biowulf. Remember you need to be connected to the NIH network either by being on campus or through VPN. Recall from lesson 1 that you use the `ssh` command below to connect to Biowulf, where username is the student account ID that was assigned to you (see student assignments). Remember that when prompted to enter your password, you are not going to be able to see it, but keep typing.

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

Navigating the Biowulf directory structure

Unix file system hierarchy

Figure 1 shows an example hierarchy of Unix file system hierarchy. At the very top, there is the root folder and every subfolder branches of from this. The root folder is denoted as `/`.

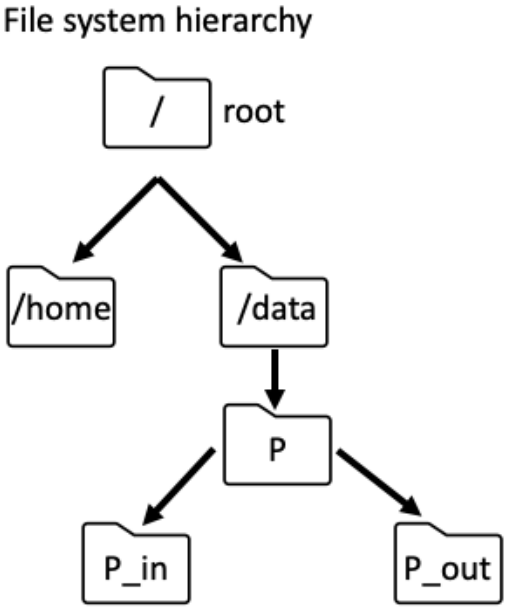


Figure 1: Example of file system hierarchy structure.

In Biowulf, the home and data folders stem from the root and this is evident by typing `ls /` at the command line. The `ls` command is used to list directory content.

```
data
home
```

Note

A file path that starts with the root or `/` is known as an aboslute path. One that does not start with a root is called a relative path. For example, in Unix, `.` is used to denote here in the present working directory and `..` is used to denote one directory back. Thus, a path that starts with `.` or `..` is a relative path.

Recall that upon signing on to Biowulf, you will land in the home directory (`/home/username` or ~). Use `pwd` to confirm the directory in which you are in.

```
pwd
```

This should return `/home/username`. Again, replace username with the student account ID that was assigned to you.

To change into the data directory, use `cd /data/username` (note the absolute path to the data folder was provided to the `cd` command).

Make a new directory

Once in the data folder, use the `mkdir` command to create a directory called lesson2.

```
mkdir lesson2
```

Then change into it. Because we are in the data folder already, we can just do `cd lesson2` without providing the absolute path the directory. Note that `cd ./lesson2` works as well where `.` denotes here in the present working directory (ie. the data folder) but it this not needed. Parts of a Unix file path are separated by "/" or forward slash.

```
cd lesson2
```

To go back to the data folder, which is one directory up, just do `cd ..`.

```
cd ..
```

Listing directory content

The `ls` command is used to list directory content.

```
ls
```

```
lesson2
```

Make a new directory called lesson2a.

```
mkdir lesson2a
```

```
ls
```

```
lesson2
lesson2a
```

Tip

If there are many items in a directory, use the `-1` option in `ls` to list the items one line at a time.

To get a detailed view of directory content, use the `-l` option with `ls`.

```
ls -l
```

```
drwxr-x---. 2 wuz8     wuz8            4096 Jan 17 17:18 lesson2
drwxr-x---. 2 wuz8     wuz8            4096 Jan 17 17:24 lesson2a
```

Unix file and directory permissions

The column "drwr-x---" in the above results from `ls -1` tells us the permission (ie. who can read - r, write - w, or execute - x contents of the file or directory), which is an important aspect of work in Unix systems like Biowulf. Figures 1 and 2 gives a breakdown of the information provided in the permission block.

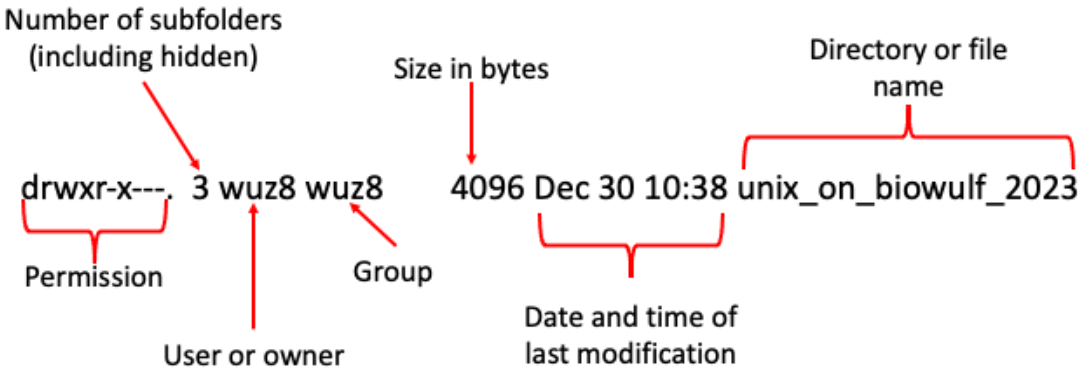


Figure 1

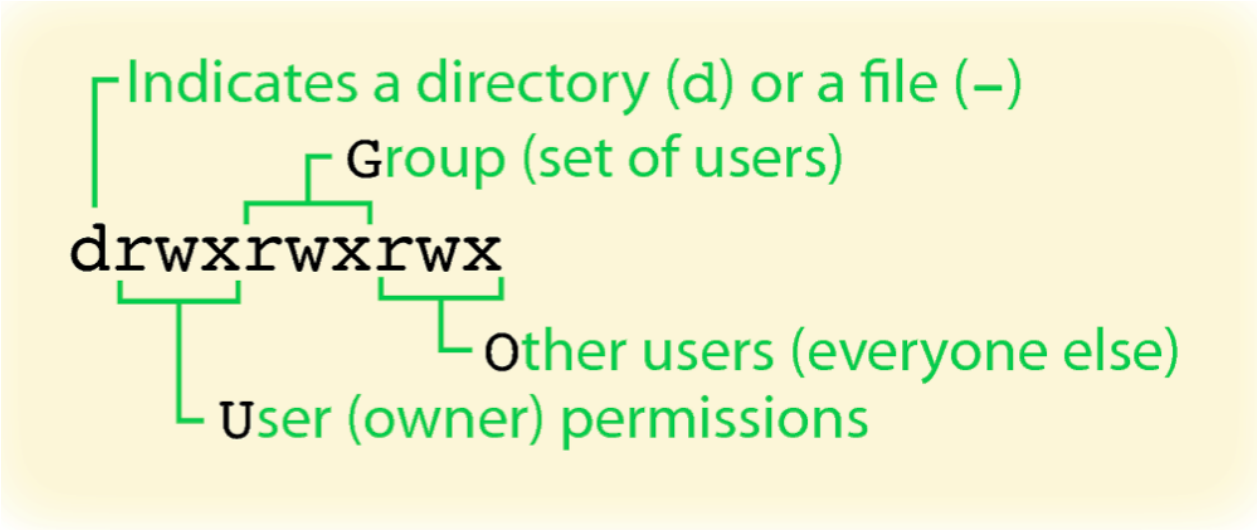


Figure 2

The command for modifying permissions is `chmod`. If we append `--help` to `chmod`, then we can see how to use it.

```
chmod --help
```

```
Usage: chmod [OPTION]... MODE[,MODE]... FILE...
  or:  chmod [OPTION]... OCTAL-MODE FILE...
  or:  chmod [OPTION]... --reference=RFILE FILE...
Change the mode of each FILE to MODE.
With --reference, change the mode of each FILE to that of RFILE.

  -c, --changes   like verbose but report only when a change is made
  -f, --silent, --quiet  suppress most error messages
  -v, --verbose          output a diagnostic for every file processe
      --no-preserve-root  do not treat '/' specially (the default)
      --preserve-root    fail to operate recursively on '/'
      --reference=RFILE  use RFILE's mode instead of MODE values
  -R, --recursive        change files and directories recursively
      --help     display this help and exit
      --version  output version information and exit

Each MODE is of the form
'[ugoa]*([-+=]([rwxXst]*|[ugo]))+|[-+=][0-7]+'.

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
For complete documentation, run: info coreutils 'chmod invocation'
```

Tip

> The `man` command can be used to pull up manuals for various Unix command. The `--help` option may sometimes be shorten as `-h` but these are command specific (ie. not every Unix command will provide help documentation using `--help` and/or `-h`).

To use `chmod`, we need to be aware that

- u is user or owner
- g is group
- o is others
- "-" is used to remove a permission
- "+" is used to add a permission
- "=" sets permission

We can also numerically set permissions where

- 0: No permission
- 1: Execute permission
- 2: Write permission
- 3: Execute and write permission (1+2=3)
- 4: Read permission
- 5: Read and execute permission (1+4=5)
- 6: Read and write permission (2+4=6)
- 7: All permission (1+2+4=7)

For instance, to change the permission for the lesson2 folder to group writable do the following.

```
chmod g+w lesson2
```

```
drwxrwx---. 2 wuz8     wuz8            4096 Jan 17 17:18 lesson2
```

Tools for transferring data between local computer and Biowulf.

See the Biowulf guide for transferring data to and from the cluster *(https://hpc.nih.gov/docs/transfer.html)* for options.

Globus

The preferred method for transferring large data files (ie. FASTQ generated from high throughput sequencing) is Globus and instructions can be found at https://hpc.nih.gov/docs/

globus/ *(https://hpc.nih.gov/docs/globus/)*. Users will need to download a Globus desktop client and set up the appropriate end points for data transfer.

Helix

Definition

"Helix (helix.nih.gov) is the interactive data transfer and file management node for the NIH HPC Systems." -- Biowulf *(https://hpc.nih.gov/systems/)*.

Tip

"Interactive Data Transfers should be performed on helix.nih.gov, the designated system for interactive data transfers and large-scale file manipulation. (An interactive session on a Biowulf compute node is also appropriate). Such processes should not be run on the Biowulf login node. For example, tarring and gzipping a large directory, or rsyncing data to another server, are examples of such interactive data transfer tasks" -- Biowulf *(https://hpc.nih.gov/docs/transfer.html)*.

To sign on to Helix, do

```
ssh username@helix.nih.gov
```

See https://bioinformatics.ccr.cancer.gov/docs/intro-to-bioinformatics-ss2023/Lesson4/HPCintro/ *(https://bioinformatics.ccr.cancer.gov/docs/intro-to-bioinformatics-ss2023/Lesson4/HPCintro/)* for useful tips on when to use Helix.

- Transferring >100 GB using scp
- gzipping a directory containing >5K files, or > 50 GB
- copying > 150 GB of data from one directory to another
- uploading or downloading data from the cloud

SCP

The `scp` command can be used to securely copy files between local and Biowulf. For instance, the command below can used to secure copy the lesson2 folder from Biowulf to local. Note that a connection to Helix was used and that you will be prompted to enter your Biowulf password.

```
scp -r username@helix.nih.gov:/data/username/lesson2 .
```

To copy the lesson2 folder from local back to Biowulf do the following.

```
scp -r ./lesson2 username@helix.nih.gov:/data/username/lesson2
```

Lesson 3: Copy, move, rename, and delete files and folders using Unix commands

Learning objectives

After this lesson, participants will become proficent with copying, moving, renaming, and removing files as well as folders using Unix commands.

Connecting to Biowulf

To get started, open the Command Prompt (Windows) or the Terminal (Mac) and connect to Biowulf. Remember you need to be connected to the NIH network either by being on campus or through VPN. Recall from lesson 1 that you use the `ssh` command below to connect to Biowulf, where username is the student account ID that was assigned to you (see student assignments). Remember that when prompted to enter your password, you are not going to be able to see it, but keep typing.

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

Change into the data directory after connecting. Replace username with the student account ID.

```
cd /data/username
```

Copying of files or folders

Copying folders

There is a folder called unix_on_biowulf_2024_documents in the /data/classes/BTEP folder. We can see it using the `ls` command with the options and arguments below.

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

```
drwxrwsr-x.  4 wuz8        GAU               4096 May 22  2023 unix_
```

The persmission for the unix_on_biowulf_2024_documents folder is set that so others can read it.

Caution

Do not change into and work in `/data/classes/BTEP/unix_on_biowulf_2024_documents`.

To copy this folder to the data directory, use the command construct below where

- `-r`: tells Unix to copy the folder and items that are in it.
- `/data/classes/BTEP/unix_on_biowulf_2024_documents`: is the argument for the `cp` command (ie. this is the item that we want to make a copy of)
- `.`: the folder where the copy should be stored. Recall that `.` denotes here in the present working directory, which should the `/data/username` folder

```
cp -r /data/classes/BTEP/unix_on_biowulf_2024_documents .
```

Use `ls` to confirm that the unix_on_biowulf_2024_documents has been copied.

Change into the unix_on_biowulf_2024_documents folder and use `ls -l` to get a detailed view of what is in it.

```
cd unix_on_biowulf_2024_documents
```

```
ls -l
```

```
-rwxr-x---. 1 wuz8 wuz8   368 Jan 18 11:20 SRP045416.swarm
drwxr-x---. 2 wuz8 wuz8  4096 Jan 18 11:20 SRR1553606
drwxr-x---. 2 wuz8 wuz8  4096 Jan 18 11:20 unix_on_biowulf_2024
-rwxr-x---. 1 wuz8 wuz8 41734 Jan 18 11:20 unix_on_biowulf_2024.zip
```

The unix_on_biowulf_2024_documents has two files and two directories.

Change into unix_on_biowulf_2024.

```
cd unix_on_biowulf_2024
```

List contents of unix_on_biowulf_2024.

```
ls -l
```

```
-rwxr-x---. 1 wuz8 wuz8  31666 Jan 18 11:20 counts.csv
-rwxr-x---. 1 wuz8 wuz8 104473 Jan 18 11:20 results.csv
-rwxr-x---. 1 wuz8 wuz8     84 Jan 18 11:20 text_1.txt
```

Copy a file

Earlier, we used `cp` with the `-r` option to recursively copy the unix_on_biowulf_2024_documents directory and all of its contents to the data directory. Suppose we want to make a copy of just one file (the counts.csv file) in the unix_on_biowulf_2024 subfolder of unix_on_biowulf_2024_documents, how would we do this? We could use the `cp` command with the following arguments.

- Argument: File to make a copy of (ie. counts.csv)
- Argument: Name of the copy (ie. counts_copy.csv)

```
cp counts.csv counts_copy.csv
```

Supposed that we want to make a copy of text_1.txt and call it text_1_copy.txt, we can use the `cp` command again.

```
cp text_1.txt text_1_copy.txt
```

Use `ls -1` to list the items in unix_on_biowulf_2024.

```
ls -1
```

The copies of counts.csv and text_1.txt have been made.

```
counts.csv
counts_copy.csv
results.csv
text_1.txt
text_1_copy.txt
```

Now, if we wanted to make a copy of counts.csv and place it one directory up in the unix_on_biowulf_2024_documents folder then we can use the command below, where `../` represents go back one directory.

```
cp counts.csv ../counts_copy_1.csv
```

```
ls -1 ../
```

```
SRP045416.swarm
SRR1553606
counts_copy_1.csv
unix_on_biowulf_2024
unix_on_biowulf_2024.zip
```

Tip

Note that the size of directory contents are listed as bytes. We can get a more human readable form of the size by appending the `-h` option to `ls -l`. Again we can use `ls --help` to find out about the `-h` option. Of interest is that options for Unix commands can also be written in the long form preceded by "--" (ie. `-h` is the same as `--human-readable`). This an example of a command that does not use `-h` as a short notation for `--help`.

```
ls --help
```

```
-h, --human-readable  with -l, print sizes in human readable format
                        (e.g., 1K 234M 2G)
```

```
ls -lh
```

```
-rwxr-x---. 1 wuz8 wuz8  31K Jan 18 11:20 counts.csv
-rwxr-x---. 1 wuz8 wuz8  31K Jan 18 11:58 counts_copy.csv
-rwxr-x---. 1 wuz8 wuz8 103K Jan 18 11:20 results.csv
-rwxr-x---. 1 wuz8 wuz8   84 Jan 18 11:20 text_1.txt
-rwxr-x---. 1 wuz8 wuz8   84 Jan 18 12:06 text_1_copy.txt
```

Tip

Speaking of file sizes, we can use the `checkquota` command to check disk usage. This should give the same result as that shown in the user dashboard.

```
checkquota
```

```
Mount                   Used      Quota  Percent    Files    Limit  I
/data:                1.5 TB     2.0 TB   75.27%   127141 32000000
/home:                3.2 GB    16.0 GB   19.87%    63871      n/a
```

Rename and moving

Stay in `unix_on_biowulf_2024` for this exercise.

To rename a file, use the `mv` command. Below, counts_copy.csv is renamed as rna_seq_counts.csv.

```
mv counts_copy.csv rna_seq_counts.csv
```

To move a file into a different directory, we can also rely on the `mv` command. Below, text_1_copy.txt is moved one directory up to the unix_on_biowulf_2024_documents folder.

```
mv text_1_copy.txt ../
```

Confirm that the move was successful.

```
ls -1 ../
```

```
SRP045416.swarm
SRR1553606
counts_copy_1.csv
text_1_copy.txt
unix_on_biowulf_2024
unix_on_biowulf_2024.zip
```

The `mv` command can be used to rename and move folders as well.

Make a new folder called lesson2.

```
mkdir lesson3
```

Rename the folder lesson2 to lesson3_january_2024_unix_class

```
mv lesson3 lesson3_january_2024_unix_class
```

Deleting files and folders

To delete a folder, use the `rm` command with the `-r` options, which enables deletion of the folder and everything that is in it.

To delete the lesson2_january_2024_unix_class directory, use the following.

```
rm -r lesson2_january_2024_unix_class
```

Tip

The `rmdir` command can be used to delete empty folders.

To delete a file, just use `rm` followed by the file to be removed.

```
rm rna_seq_counts.csv
```

Snapshots

Biowulf and other high performance computing clusters do not have a recycling bin or trash can to hold deleted files or folders and allow users to restore if they removed by accident. However, users are able to restore from snaphots. See https://hpc.nih.gov/storage/backups.html *(https://hpc.nih.gov/storage/backups.html)* to learn how to restore from snapshot on Biowulf.

Definition

"A 'snapshot' is a read-only copy of the data at a particular point in time." -- Biowulf *(https://hpc.nih.gov/storage/backups.html)*

A snapshot of the files or folders have to be made in order for restoration to be possible.

Caution

Be care when deleting items when working on Biowulf.

The way to access snapshots for the data directory depends on the user's storage system. Use `ls -ld /data/username` to find out. Replace username with your specific Biowulf account ID. In these examples, my Biowulf account ID (wuz8) will be used to demonstrate.

```
ls -ld /data/wuz8
```

From the results below, my data directory is on a VAST storage system evident by it beginning with `/vf`.

```
lrwxrwxrwx. 1 root root 14 May  5  2023 /data/wuz8 -> /vf/users/wuz8
```

Note

The `/data/wuz8` path is just symlink which points to an actual file storage location. Symlinks are useful as they provide shortcuts for referencing things like really long file paths.

The data directory has daily and weekly snapshots. Choose the the appropriate one and append your username to goto your user specific data folder snapshot.

```
ls -1 /vf/users/.snapshot/
```

```
daily-snap._2024-01-16_04_00_00_UTC
daily-snap._2024-01-17_04_00_00_UTC
daily-snap._2024-01-18_04_00_00_UTC
weekly-snap._2024-01-07_07_00_00_UTC
weekly-snap._2024-01-14_07_00_00_UTC
```

```
ls -1 /vf/users/.snapshot/daily-snap._2024-01-18_04_00_00_UTC/wuz8
```

Suppose that I want to restore a file example_text.txt from snapshot back to my data folder, the following can be used.

```
cp /vf/users/.snapshot/daily-snap._2024-01-18_04_00_00_UTC/wuz8/exam
```

Snapshots are also made for the home directory.

```
ls ~/.snapshot
```

```
Hourly.2024-01-17_1400  Hourly.2024-01-18_0800   Nightly.2024-01-17_(
Hourly.2024-01-17_1700  Hourly.2024-01-18_1100   Nightly.2024-01-18_(
Hourly.2024-01-17_2000  Nightly.2024-01-12_0010  Weekly.2023-11-26_0(
Hourly.2024-01-17_2300  Nightly.2024-01-13_0010  Weekly.2023-12-03_0(
Hourly.2024-01-18_0200  Nightly.2024-01-15_0010  Weekly.2023-12-10_0(
Hourly.2024-01-18_0500  Nightly.2024-01-16_0010  Weekly.2023-12-17_0(
```

Lesson 4 (Working with bioinformatics software on Biowulf)

Lesson 4: Working with bioinformatics software on Biowulf

Learning objectives

After this lesson, we participants will

- Know how to request an interactive session on Biowulf
- Know how to software that are installed on Biowulf
- Be able to sign onto Helix and download sequencing data from SRA.
- Load software that are installed on Biowulf and become familiar with running some bioinformatics applications using Unix command line

Connecting to Biowulf

To get started, open the Command Prompt (Windows) or the Terminal (Mac) and connect to Biowulf. Remember you need to be connected to the NIH network either by being on campus or through VPN. Recall from lesson 1 that you use the `ssh` command below to connect to Biowulf, where username is the student account ID that was assigned to you (see student assignments). Remember that when prompted to enter your password, you are not going to be able to see it, but keep typing.

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

Requesting an interactive session

> **Recall**
>
> The Biowulf login node is meant for job submission to the batch system and should not be used to perform any computation intensive tasks. For testing computation intensive tasks without submitting a job, request an interactive session to work on one of Biowulf's compute nodes.

To request an interactive session do the following.

```
```

```
sinteractive
```

```
salloc: Pending job allocation 17385251
salloc: job 17385251 queued and waiting for resources
salloc: job 17385251 has been allocated resources
salloc: Granted job allocation 17385251
salloc: Waiting for resource configuration
salloc: Nodes cn4298 are ready for job
srun: error: x11: no local DISPLAY defined, skipping
error: unable to open file /tmp/slurm-spank-x11.17385251.0
slurmstepd: error: x11: unable to read DISPLAY value
```

Note

The number 17385251 in the `sinteractive` output is the job ID. This important because users can reference it to view job details and cancel jobs if submitting to the batch system.

Important

The prompt changes to username@cn#### from username@biowulf when successfully connected to an interactive session, where "cn####" is the name of one of the Biowulf compute nodes.

Above, `sinteractive` was run without options (ie. with the defaults).

```
jobhist 17385251
```

```
JobId              : 17385251
User               : wuz8
Submitted          : 20240118 17:33:17
Started            : 20240118 17:33:25
Ended              :

Jobid        Partition       State  Nodes  CPUs      Walltime       M
17385251    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. While the MemReq shows 2 GB of RAM was requested, it is actually 1.5 GB of RAM (0.768 GB x 2 CPU). Biowulf just rounded to the nearest integer.

Partitions

> "Partitions define limitations that restrict the resources that can be requested for a job submitted to that partition. The limitations affect the maximum run time, the amount of memory, and the number of available CPU cores (which are called CPUs in Slurm)." -- https://wiki.hpcuser.uni-oldenburg.de/index.php?title=Partitions *(https://wiki.hpcuser.uni-oldenburg.de/index.php?title=Partitions)*. "Jobs should be submitted to the partition that best matches the required resources." -- https://wiki.hpcuser.uni-oldenburg.de/index.php?title=Partitions *(https://wiki.hpcuser.uni-oldenburg.de/index.php?title=Partitions)*.

NCI-CCR, NHLBI and NINDS, and NIHM have buy-in nodes (partitions). To request an interactive session for the NCI-CCR partition, use `sinteractive --constraint=ccr`. See the following links from Biowulf regarding the buy-in nodes.

- NCI-CCR: https://hpc.nih.gov/docs/ccr.html *(https://hpc.nih.gov/docs/ccr.html)*
- NHLBI and NINDS: https://hpc.nih.gov/docs/forgo.html *(https://hpc.nih.gov/docs/forgo.html)*
- NIMH: https://hpc.nih.gov/docs/nimh.html *(https://hpc.nih.gov/docs/nimh.html)*

Software on Biowulf

Biowulf staff has installed many applications, including those used in genomic data analysis. In general, to view the applications that are available on Biowulf, we can use the module command, with its avail subcommand. This will essentially print out a list of applications that are on Biowulf and we can use the up and down arrows to navigate and view the list. We hit "q" to exit this list.

```
module avail
```

To list only the default version of each application, include the -d option in module avail.

```
module -d avail
```

To check if a specific application is available, you can append the name of the module after `module avail. For instance, we do that with the genomic sequencing Star aligner Bowtie below.

```
module avail star
```

We can use the whatis subcommand to see information regarding a specific tool and also to confirm if Biowulf has it installed. For instance, we can check for fastqc, which is an application used to assess quality of high throughput sequencing data. The output provides a description

of what the tool does and the default version if we load the tool. The whatis subcommand is case sensitive.

```
module whatis fastqc
```

To load an application we can use module load. Let's load the sratool kit and fastqc. By default, the latest version of an application is loaded.

```
module load fastqc
```

```
[+] Loading fastqc  0.11.9
```

```
module load sratoolkit
```

The following error is obtained when loading sratoolkit. This is triggered because sratoolkit write temporary files and requires local temporary storage space.

```
Lmod has detected the following error:

This module requires allocation of /lscratch. Please see

   https://hpc.nih.gov/docs/userguide.html#local

or contact staff@hpc.nih.gov for more information.

While processing the following module(s):
    Module fullname    Module Filename
    ---------------    ---------------
    sratoolkit/3.0.10  /usr/local/lmod/modulefiles/sratoolkit/3.0.10
```

To resolve the above issue with loading sratoolkit, exit the interactive session.

```
exit
```

```
srun: error: cn4298: task 0: Exited with exit code 1
salloc: Relinquishing job allocation 17385251
salloc: Job allocation 17385251 has been revoked.
```

Request another interactive session with local temporary storage on the assigned Biowulf compute node. To this append the `--gres` option to `sinteractive`, where `gres` stands for generic resources. Set `gres` to `lscratch` (ie. local temporary storage) and indicate the size in gigabytes. For instance, the command construct below asks for 10 gigabytes of local temporary storage.

```
sinteractive --gres=lscratch:15
```

The `module load sratoolkit` and `module load fastqc`.

Exploring bioinformatics tools

Here, we will download some high throughput genomic sequences from NCBI SRA. The data that we will download were derived from sequencing of the Zaire Ebola virus. See the NCBI SRA page for this study *(https://www.ncbi.nlm.nih.gov/sra/?term=SRR1553606)* for more details. For this part of the exercise, sign onto Helix.

Start a new Terminal (Mac) or Command Prompt (Window).

```
ssh username@Helix.nih.gov
```

We will use a command called `fastq-dump`` within the sratoolkit to grab the first 10000 reads for this sequencing run. In the syntax for fastq-dump

- `--split-files` will generate two files that contains the forward and reverse reads from paired-end sequencing.
- `-X` allows us to input how many reads we want to obtain (here, we just want the first 10000 reads to save time and computation resources for this class)
- Finally, we enter the SRA accession number of the sequencing data that we want to download (SRR1553606 in this example).
- Create a directory called SRR1553606 to store the sequencing data.

```
mkdir SRR1553606
```

```
cd /data/username/SRR1553606
```

Module load sratoolkit.

```
module load sratoolkit
```

```
fastq-dump --split-files -X 10000 SRR1553606
```

After download has completed, there should be two fastq files.

```
ls
```

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

Go back to the Terminal or Command Prompt with the Biowulf interactive session and stay in the /data/username/SRR1553606 folder.

The first task in analyzing high throughput sequencing data is to perform quality check using tools such as FASTQC. Look at the help documents to learn how to run FASTQC.

```
fastqc --help
```

The command construct starts with `fastqc` followed by the arguments, which are the files that the user wants to perform quality check on.

```
fastqc seqfile1 seqfile2 .. seqfileN
```

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

Listin the content of the directory will reveal the FASTQC results in html and zip format. The html file can be viewed locally in a web browser while the zip file when expanded contains text summaries and individual quality metric images presented in the html file.

```
SRR1553606_1.fastq
SRR1553606_1_fastqc.html
SRR1553606_1_fastqc.zip
SRR1553606_2.fastq
SRR1553606_2_fastqc.html
SRR1553606_2_fastqc.zip
SRR1553606_fastqc_log
SRR1553606_fastqc.sh
```

Seqkit is a package that enables users to find and work with sequences. The `stats` function can be used to obtain fastq file statistics.

```
```

```
module load seqkit
```

```
seqkit stats SRR1553606_1.fastq SRR1553606_2.fastq
```

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

Lesson 5: Submitting jobs to the Biowulf batch system

Learning objectives

After this lesson, participants will

- Be able to describe shell and swarm scripts
- Use the Nano editor to edit scripts
- Submit shell and swarm scripts to the Biowulf batch system

Connecting to Biowulf

To get started, open the Command Prompt (Windows) or the Terminal (Mac) and connect to Biowulf. Remember you need to be connected to the NIH network either by being on campus or through VPN. Recall from lesson 1 that you use the `ssh` command below to connect to Biowulf, where username is the student account ID that was assigned to you (see student assignments). Remember that when prompted to enter your password, you are not going to be able to see it, but keep typing.

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

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

```
cd /data/username
```

Swarm scripts

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

Make a folder called SRP045416.

```
mkdir SRP045416
```

```
cd SRP045416
```

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 17387605 and there are five sub-jobs as indicated by [0-4], which concords with the five commands in the script.

> "By default, each subjob is allocated a 1.5 gb of memory and 1 core (consisting of 2 cpus)." -- Biowulf swarm documentation *(https://hpc.nih.gov/apps/swarm.html)*

```
[wuz8@cn4285 wuz8]$ sjobs
User  JobId       JobName     Part         St  Reason  Runtime     Walltime     Nodes  CPUs   Memory  Dependency  Nodelist
=============================================================================================================================
=
wuz8  17387605    sinteracti  interactive  R               48:02      8:00:00      1      2    2 GB             cn4285
wuz8  17389173_4  SRP045416   norm         R                1:10      2:00:00      1      2    2 GB             cn4330
wuz8  17389173_0  SRP045416   norm         R                1:10      2:00:00      1      2    2 GB             cn4310
wuz8  17389173_1  SRP045416   norm         R                1:10      2:00:00      1      2    2 GB             cn4328
wuz8  17389173_2  SRP045416   norm         R                1:10      2:00:00      1      2    2 GB             cn4330
wuz8  17389173_3  SRP045416   norm         R                1:10      2:00:00      1      2    2 GB             cn4330
=============================================================================================================================
=
cpus queued  = 0
cpus running = 0 / 12
mem  queued  = 0.0 B
mem  running = 48.0 MB / 9.0 GB
jobs queued  = 0
jobs running = 6
```

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
```

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)

To submit this script

```
sbatch SRP045416_stats.sh
```

To view the output file SRP045416_stats.txt

```
```

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

Lesson 6: Using Unix commands to work with text files and tabular data

Learning objectives

After this lesson, participants will

- Know how to download data from the web
- Work with tape archives (tar files)
- Be able to create text files and tabular data
- Be able to view and page through file content
- Know how to use Unix commands to perform basic wrangling tasks such as
    - Pattern searching
    - Substitution
    - Subsetting
    - Sorting

To get started, open the Command Prompt (Windows) or the Terminal (Mac) and connect to Biowulf. Remember you need to be connected to the NIH network either by being on campus or through VPN. Recall from lesson 1 that you use the `ssh` command below to connect to Biowulf, where username is the student account ID that was assigned to you (see student assignments). Remember that when prompted to enter your password, you are not going to be able to see it, but keep typing.

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

Change into the data directory when connected.

```
cd /data/username
```

Getting example data

Copy the example_rna_sequencing folder in `/data/classes/BTEP` to the data directory and then change into it.

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

```

```

```
cd example_rna_sequencing
```

See what is in the example_rna_sequencing folder.

```
ls -l
```

There are two folders, hbr_uhr_fastq and hcc1395_fastq that contains sequencing data for the human brain and universal human reference (HBR/UHR) and the HCC1395 studies as documented in the RNA sequencing tutorial provided by the Griffith lab *(https://rnabio.org/module-01-inputs/0001/05/01/RNAseq_Data/)*. The csv or comma separate files contain gene expression counts and differential gene expression results from these two studies.

```
-rwxr-x--- 1 wuz8 wuz8  31666 Jan 19 12:04 hbr_uhr_counts.csv
-rwxr-x--- 1 wuz8 wuz8 104473 Jan 19 12:04 hbr_uhr_deg_results.csv
drwxr-x--- 2 wuz8 wuz8   4096 Jan 19 12:04 hbr_uhr_fastq
drwxr-x--- 2 wuz8 wuz8   4096 Jan 19 12:04 hcc1359_fastq
-rwxr-x--- 1 wuz8 wuz8  33230 Jan 19 12:04 hcc1395_counts.csv
-rwxr-x--- 1 wuz8 wuz8 106687 Jan 19 12:04 hcc1395_deg_results.csv
```

Downloading data from the web

Go back to the data directory, which is one folder up.

```
cd ../
```

Suppose you need to download the fastq files for the HCC1395 study from the Griffith lab RNA sequencing tutorial page using this URL http://genomedata.org/rnaseq-tutorial/practical.tar *(http://genomedata.org/rnaseq-tutorial/practical.tar)*, there are two Unix commands to this. These are `wget` and `curl`. Note in the URL for HCC1395 data that the file name is practical.tar.

Make a folder called hcc1395_fastq_download.

```
mkdir hcc1395_fastq_download
```

```
cd hcc1395_fastq_download
```

To download using `wget` just provide the URL to the data as an argument. Include `-O` specify a file name. The name practical.tar is not informative, so change download it as hcc1395_fastq.tar. Use `wget` to download for this class.

```
wget -O hcc1395_fastq.tar http://genomedata.org/rnaseq-tutorial/prac
```

To download using `curl`, provide the data's URL and name of the output using `-o` or '-O'. The `-o` option enables users to supply a name other than is in the data's URL (ie. practical.tar). The `-O` option downloads the data and names it using the that shown in the data's URL.

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

```
curl -O http://genomedata.org/rnaseq-tutorial/practical.tar
```

After download

```
ls
```

```
hcc1395_fastq.tar
```

Tape archive or tar files

The `wget` command above downloaded the HCC1395 fastq files in the form of a tape archive or tar file. Tape archives are essentially packages or collections of files and folders and enable easy transfer and sharing.

To get the HCC1395 fastq files, this file needs to be unpacked using the `tar` command with the options below.

- `-x`: extract files from an archive
- `-v`: verbosely list files processed
- `-f`: use archive file or device ARCHIVE

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

The contents that are unpacked are displayed because the `-v` option was included.

```
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
```

Go back to the date directory and make a folder called hbr_uhr_fastq. Download the fastq files for HBR/UHR from http://genomedata.org/rnaseq-tutorial/HBR_UHR_ERCC_ds_5pc.tar *(http://genomedata.org/rnaseq-tutorial/HBR_UHR_ERCC_ds_5pc.tar)* and name the tar file hbr_uhr_fastq.tar. Finish the task by unpacking.

```
cd ../
```

```
mkdir hbr_uhr_fastq_download
```

```
cd hbr_uhr_fastq_download
```

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
curl -o hbr_uhr_fastq.tar http://genomedata.org/rnaseq-tutorial/HBR_l
```

```
tar -xvf hbr_uhr_fastq.tar
```

{{Edet}}

Viewing and working with fastq files using Unix commands

Change back to `/data/username/hcc1395_fastq_download` for this exercise.

```
cd /data/username/hcc1395_fastq_download
```

The fastq files were compressed to save on storage space as evident by the extension "gz", which stands for gzip. Usually, `cat` can be used to view fastq files in addition to text files and tabular data. However, it will not work with compressed files. Fortunately, there is a work around

for this without having to uncompress the files. Hit control-c or control-z to exit `zcat` and return to the command prompt.

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

While `zcat` will print all of the sequences for a fastq file to the terminal, it is not a convenient way to view large files. The `|` or pipe can be used to send output of one command to another. Here, it is used to send the output of `zcat` to `less`, which allows users to scroll through file content line by line using the up and down arrows or page by page using the space bar. Hit `q` to exit `less` and return to the prompt.

```
zcat hcc1395_normal_rep1_r1.fastq.gz | less
```

Alternatively, the `head` command can be used to look at the first couple of lines (defaults to first 10 lines).

```
zcat hcc1395_normal_rep1_r1.fastq.gz | head
```

Including the `-n` option in head enables users specify the number of lines to print. Because sequences in a fastq file come in four lines (ie. header, sequence, "+", quality score), `-n 4` could be used to view the first sequnence in a fastq file.

```
zcat hcc1395_normal_rep1_r1.fastq.gz | head -n 4
```

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

The `tail` command is used to look at lines at the bottom of a file. Again, it has `-n` option for users to specify how many lines.

Creating text files and tabular data

For this exercise, change back into the data directory.

```
cd /data/username
```

Make a directory called text_files_and_tabular_data.

```
mkdir text_files_and_tabular_data
```

```
cd text_files_and_tabular_data
```

The `touch` command can be used to create empty files.

```
touch example.txt
```

If we do `cat example.txt`, nothing will be printed to the terminal. To edit and add stuff to example.txt do the following. The `-L` option prevents the addition of a new line at the end of the file.

```
nano -L example.txt
```

Add the following to example.txt. Hit control-x and then "y" to save and exit nano and return to the prompt.

```
DNA sequencing
RNA sequencing
ChIp sequencing
ATAC sequencing
```

Now, doing `cat example.txt` will print the file contents to the terminal. To add a new line that says scRNA sequencing, do the follwing.

```
echo "scRNA sequencing" >> example.txt
```

```
cat example.txt
```

```
DNA sequencing
RNA sequencing
ChIp sequencing
ATAC sequencing
scRNA sequencing
```

Pattern searching

The command `grep` can be used to search for patterns in a file. For instance, to get the sequencing modalities for RNA in example.txt use

```
grep RNA example.txt
```

```
RNA sequencing
scRNA sequencing
```

The `grep` output can be saved using the redirect >.

```
grep RNA example.txt > rna_sequencing.txt
```

```
cat rna_sequencing.txt
```

```
RNA sequencing
scRNA sequencing
```

Including the `-o` option in `grep` will print only the pattern.

```
grep -o ATAC example.txt
```

```
ATAC
```

Note

`grep` is case senstive by defaul. Use the `-i` option to ignore case.

The `-v` option will return lines that do not contain the pattern.

```
grep -v RNA example.txt
```

```
DNA sequencing
ChIp sequencing
ATAC sequencing
```

Deleting and adding lines

The `sed` command can be used to perform various transformations on files. For instance, to delete the first line corresponding to DNA sequencing in example.txt use the following where the option `d` denotes delete and `1` is added to indicate delete line 1. To delete the second line, replace `1` with `2`. Note the output of the `sed` command below was not saved but only printed to the terminal.

```
sed '1d' example.txt
```

```
RNA sequencing
ChIp sequencing
ATAC sequencing
scRNA sequencing
```

To insert a line with `sed` use the `i` option followed by the text to insert. To code below inserts "Spatial transcriptomics" into the first line of example.txt.

```
sed '1i Spatial transcriptomics' example.txt
```

Pattern subsitution

The `sed` command can also be used for substitution. For instance, to change all of the instance of sequencing to "seq" in example.txt the following can be used. The pattern, sequencing is given first, followed by the substitute (ie. seq). These are separated by "/".

```
sed 's/sequencing/seq/' example.txt
```

```
DNA seq
RNA seq
ChIp seq
ATAC seq
scRNA seq
```

Working with tabular data

Data analysis requires investigators to work with tabular data either in comma separated (csv) or tab separated format. Columns in a comma separated file are separated by commas. Those in tab separated files are separated by tabs.

Change into the /data/username/example_rna_sequencing folder.

```
cd /data/username/example_rna_sequencing
```

Then take look at the first few lines of hbr_uhr_counts.csv, which contains gene expression counts for the HBR/UHR study.

```
cat hbr_uhr_counts.csv | head
```

The columns Geneid, HBR_1.bam,HBR_2.bam,HBR_3.bam,UHR_1.bam,UHR_2.bam,UHR_3.bam are separated by commas. The way `cat` displays the file content is not nice.

```
Geneid,HBR_1,HBR_2,HBR_3,UHR_1,UHR_2,UHR_3
U2,0,0,0,0,0,0
CU459211.1,0,0,0,0,0,0
CU104787.1,0,0,0,0,0,0
BAGE5,0,0,0,0,0,0
ACTR3BP6,0,0,0,0,0,0
5_8S_rRNA,0,0,0,0,0,0
AC137488.1,0,0,0,0,0,0
AC137488.2,0,0,0,0,0,0
CU013544.1,0,0,0,0,0,0
```

Sending the output of `head` to `column` allows the columns in the file to be printed nicely aligned. In the `column` command, `-t` tells is to create a table, and `-s` is used specify the column separator (ie. comma).

```
head hbr_uhr_counts.csv | column -t -s ','
```

```
Geneid       HBR_1  HBR_2  HBR_3  UHR_1  UHR_2  UHR_3
U2           0      0      0      0      0      0
CU459211.1   0      0      0      0      0      0
CU104787.1   0      0      0      0      0      0
BAGE5        0      0      0      0      0      0
```

```
ACTR3BP6    0       0       0       0       0       0
5_8S_rRNA   0       0       0       0       0       0
AC137488.1  0       0       0       0       0       0
AC137488.2  0       0       0       0       0       0
CU013544.1  0       0       0       0       0       0
```

Subset tabular data by column

The `cut` command can be used to subset tabular data by column. To do this, specify the column number using `-f` option and the column separator using the `-d` option. For instance, to subset out the Geneid, HBR_1, and UHR_1 columns which corresponds to columns 1, 2, and 5 the following can be used. The column numbers will be separated by commas. The column separator `-d` will be set to ',' for comma.

```
cut -f1,2,5 -d ',' hbr_uhr_counts.csv | head
```

```
Geneid,HBR_1,UHR_1
U2,0,0
CU459211.1,0,0
CU104787.1,0,0
BAGE5,0,0
ACTR3BP6,0,0
5_8S_rRNA,0,0
AC137488.1,0,0
AC137488.2,0,0
CU013544.1,0,0
```

To subset a range of columns, for instance 1 thru 4 in hbr_uhr_counts.csv, set `-f` to 1-4.

```
cut -f1-4 -d ',' hbr_uhr_counts.csv | head
```

```
Geneid,HBR_1,HBR_2,HBR_3
U2,0,0,0
CU459211.1,0,0,0
CU104787.1,0,0,0
BAGE5,0,0,0
ACTR3BP6,0,0,0
5_8S_rRNA,0,0,0
AC137488.1,0,0,0
```

```
AC137488.2,0,0,0
CU013544.1,0,0,0
```

Subset tabular data by row

To subset the counts for gene RABL2B in hbr_uhr_counts.csv, the `awk` command can be used. `awk` is one of the many Unix commands that are suitable for data processing. In the construct below

- `-F` is the `awk` option to specify the field separator (a comma in this clase)
- The commands within `awk` are enclosed in ''
    - `FNR==1` tells `awk` to print the first row, which is the column heading
    - `||` is the OR operator
    - `$1` is column 1 (the Geneid) and this is set using == to the gene RABL2B
    - Essentially, this command tells `awk` to print the first row or the row where the Gene ID matches RABL2B in hbr_uhr_counts.csv

```
awk -F, 'FNR==1 || $1=="RABL2B"' hbr_uhr_counts.csv
```

```
Geneid,HBR_1,HBR_2,HBR_3,UHR_1,UHR_2,UHR_3
RABL2B,74,62,54,68,50,47
```

Sorting

The final exercise in this lesson is to sort. To do this create text file called sorting_example.txt in the data directory.

```
cd /data/username
```

```
nano -L sorting_example.csv
```

Then enter the following. Hit control-x and "y" to save to exit nano and return to the prompt.

```
HIF1a,35
EPAS1,40
ADA,25
AMPD3,15
ADSS,20
```

```
ADSL,50
HPRT,75
```

To sort column 2 of this file from the lowest value to highest value the following can be used. `sort` is the Unix command for sorting.

- `-t` prompts for the column separator (comma in this case)
- `-k` prompts for the column number to sort on (column 2, so the construct is `-k2`)
- `-n` indicates to sort numerically

```
sort -t ',' -k2 -n sorting_example.csv
```

```
AMPD3,15
ADSS,20
ADA,25
HIF1a,35
EPAS1,40
ADSL,50
HPRT,75
```

The `-r` option will sort in reverse.

```
sort -t ',' -k2 -n -r sorting_example.csv
```

```
HPRT,75
ADSL,50
EPAS1,40
HIF1a,35
ADA,25
ADSS,20
AMPD3,15
```

Help Sessions

Lesson 1: Practice questions

Question 1:

Name the softwares for Windows and Macs that we use to connect to Biowulf.

{{Sdet}}{{Ssum}}Solution{{Esum}}

For Windows 10 or beyond users, we have the Command Prompt application. Alternatively, Windows users can use PuTTy or MobaXterm.

Mac users can use the built in Terminal application.

{{Edet}}

Question 2:

Connect to your Biowulf account, how do we do this?

{{Sdet}}{{Ssum}}Solution{{Esum}}

Via the Windows Command Prompt or Mac Terminal

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

{{Edet}}

Question 3:

What is the hierarchical architecture of Biowulf?

{{Sdet}}{{Ssum}}Solution{{Esum}}

- Cluster
- Computer/node
- Processor
- Core
- CPU

{{Edet}}

Question 4:

What is the Unix command for checking your username and group affiliation?

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
id
```

{{Edet}}

Question 5:

What is the Unix command for making a new directory?

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
mkdir
```

{{Edet}}

Question 6:

Upon signing on to Biowulf, users will land in the log-in node and should not be used for compute intensive tasks (True or False).

{{Sdet}}{{Ssum}}Solution{{Esum}}

True

{{Edet}}

Question 7:

The Biowulf home directory is where users should store data and analysis output and it's storage limit can be increased (True or False).

{{Sdet}}{{Ssum}}Solution{{Esum}}

False

{{Edet}}

Lesson 2: Practice questions

Question 1:

What command is used to check the directory in which the user is in?

{{Sdet}}{{Ssum}}Solution{{Esum}}

`pwd`

{{Edet}}

Question 2:

What command is used to change to another directory.

{{Sdet}}{{Ssum}}Solution{{Esum}}

`cd` {{Edet}}

Question 3:

Give an absolute and relative file path

{{Sdet}}{{Ssum}}Solution{{Esum}}

An absolute file path starts at the root (ie. /data/username/example.txt)

A relative file path references the current directory (ie. ./example.txt) where `.` denotes here in the current folder.

{{Edet}}

Question 4:

Given that you are in the /data/username/lesson2 folder, how do you go back up one directory back to /data/username.

{{Sdet}}{{Ssum}}Solution{{Esum}}

`cd ..`

{{Edet}}

Question 5:

How do we list directory content?

{{Sdet}}{{Ssum}}Solution{{Esum}}

`ls` where the `-1` option prints the items one line at a time and `-l` gives a detailed view of the directory content include perfmissions.

{{Edet}}

Question 6:

What command is used to change file or folder permissions?

{{Sdet}}{{Ssum}}Solution{{Esum}}

`chmod`

{{Edet}}

Question 7:

What is the preferred method for transferring large data on Biowulf?

{{Sdet}}{{Ssum}}Solution{{Esum}}

Globus

{{Edet}}

Lesson 3: Practice questions

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

Question 1

Copy the lesson3_practice folder from /data/classes/BTEP/unix_on_biowulf_2024_practice_sessions to the present working directory, which should be /data/username.

{{Sdet}}{{Ssum}}Solution{{Esum}}

cp -r /data/classes/BTEP/unix_on_biowulf_2024_practice_sessions/lesson3_practice .

{{Edet}}

Question 2

Change into the lesson3_practice folder.

{{Sdet}}{{Ssum}}Solution{{Esum}}

cd lesson3_practice

{{Edet}}

Question 3

How many files and directories are in the lesson3_practice folder?

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
ls -l
```

```
drwxr-x--- 2 wuz8 wuz8 4096 Jan 18 21:34 sample_sequence_data
-rw-r----- 1 wuz8 wuz8   46 Jan 18 21:34 text1.txt
```

One director and one file.

{{Edet}}

Question 4

Rename text1.txt to text_file1.txt.

{{Sdet}}{{Ssum}}Solution{{Esum}}

mv text1.txt text_file1.txt

{{Edet}}

Question 5

Make a copy of text_file1.txt and call it text_file2.txt.

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
cp text_file1.txt text_file2.txt
```

{{Edet}}

Question 6

Delete text_file2.txt.

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
rm text_file2.txt
```

{{Edet}}

Lesson 4: Practice questions

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

Question 1

Copy the lesson4_practice folder from /data/classes/BTEP/unix_on_biowulf_2024_practice_sessions to the present working directory, which should be /data/username.

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
cp -r /data/classes/BTEP/unix_on_biowulf_2024_practice_sessions/lesso
```

{{Edet}}

Question 2

Change into the lesson4_practice folder.

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
cd lesson4_practice
```

{{Edet}}

Question 3

How many folders are in this directory and what is the name of this folder?

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
ls -l
```

There is one folder called sample_sequence_data.

{{Edet}}

Question 4

Change into the folder sample_sequence_data.

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
cd sample_sequence_data
```

There is one folder called sample_sequence_data.

{{Edet}}

Question 5

Request an interactive session with defaults.

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
sinteractive
```

{{Edet}}

Question 6

Load the package seqkit

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
module load seqkit
```

{{Edet}}

Question 7

How many sequences are in the file HBR_1_R1.fq?

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
seqkit stats HBR_1_R1.fq
```

```
file         format  type  num_seqs     sum_len  min_len  avg_len  ma
HBR_1_R1.fq  FASTQ   DNA    118,571  11,857,100      100      100
```

{{Edet}}

Question 8

Is there an application called salmon installed on Biowulf?

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
module avail salmon
```

```
salmon/1.7.0    salmon/1.10.0    salmon/1.10.1 (D)    salmonte/0.4

  Where:
   D:  Default Module

Module defaults are chosen based on Find First Rules due to Name/Vers
See https://lmod.readthedocs.io/en/latest/060_locating.html for deta

If the avail list is too long consider trying:

"module --default avail" or "ml -d av" to just list the default modu
"module overview" or "ml ov" to display the number of modules for ea

Use "module spider" to find all possible modules and extensions.
Use "module keyword key1 key2 ..." to search for all possible module
```

{{Edet}}

Question 9

What does the package salmon do?

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
module whatis salmon
```

```
salmon/1.10.1       : Estimating transcript-level expression from RN
salmon/1.10.1       : URL =>  http://combine-lab.github.io/salmon
```

{{Edet}}

Question 10

Sign on to Helix and download the first 1000 sequences for SRA SRR27044741. This was sequence in pair end mode.

{{Sdet}}{{Ssum}}{{Esum}}

```
ssh username@helix.nih.gov
```

```
module load sratoolkit
```

```
fastq-dump --split-files -X 1000 SRR27044741
```

{{Edet}}

Lesson 5: Practice questions

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

Question 1

Make a directory called SRP475677 and change into it.

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
mkdir SRP475677
```

```
cd SRP475677
```

{{Edet}}

Question 2

Submit a swarm script (name it SRP475677.swarm) to download the first 1000 sequences for the following accessions from SRA. Paired end mode was used.

- SRR27044727
- SRR27044728
- SRR27044729
- SRR27044733
- SRR27044734

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
nano SRP475677.swarm
```

```
#SWARM --job-name SRP475677
#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 1000 SRR27044727
```

```
fastq-dump --split-files -X 1000 SRR27044728
fastq-dump --split-files -X 1000 SRR27044729
fastq-dump --split-files -X 1000 SRR27044733
fastq-dump --split-files -X 1000 SRR27044734
```

Hit control-x and then "y" to save and exit nano.

```
swarm -f SRP475677.swarm
```

{{Edet}}

Question 3

Submit a shell script (name it SRP475677.sh) to run `seqkit stats` for the FASTQ files that were just downloaded.

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
nano SRP475677.sh
```

```
#!/bin/bash
#SBATCH --job-name=SRP475677_stats
#SBATCH --mail-type=ALL
#SBATCH --mail-user=username@nih.gov
#SBATCH --mem=1gb
#SBATCH --partition=student
#SBATCH --time=00:02:00
#SBATCH --output=SRP475677_stats_log

#LOAD REQUIRED MODULES
module load seqkit

#CREATE TEXT FILE TO STORE THE seqkit stat OUTPUT
touch SRP475677_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 >> SRP475677_stats.txt;
done
```

```
sbatch SRP475677.sh
```

{{Edet}}

Question 4

What command is used to view the text file containing the `seqkit stats` results for the FASTQ files downloaded?

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
cat SRP475677_stats.txt
```

{{Edet}}

Question 5

How can the shell script in Question 3 be changed to obtain FASTQC results?

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
nano SRP475677_fastqc.sh
```

```
#!/bin/bash
#SBATCH --job-name=SRP475677_fastqc
#SBATCH --mail-type=ALL
#SBATCH --mail-user=username@nih.gov
#SBATCH --mem=1gb
#SBATCH --partition=student
#SBATCH --time=01:00:00
#SBATCH --output=SRP475677_fastqc_log

#LOAD REQUIRED MODULES
module load fastqc

#CREATE A FOR LOOP TO LOOP THROUGH THE FASTQ FILES AND RUN FASTQC

for file in *.fastq;
do fastqc $file;
done
```

```
sbatch SRP475677_fastqc.sh
```

{{Edet}}

Lesson 6: Practice questions

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 data directory (ie. `/data/username`) and change into it. Next, run the perl script in `/data/classes/BTEP/unix_on_biowulf_2024_practice_sessions/lesson6_practice/`.

{{Sdet}}{{Ssum}}Solution{{Esum}}

```
mkdir treasure_hunt
cd treasure_hunt
perl /data/classes/BTEP/unix_on_biowulf_2024_practice_sessions/lesso
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}}{{Ssum}}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}}{{Ssum}}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}}{{Ssum}}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}}{{Ssum}}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.

Connecting to Biowulf (additional methods)

Interfacing with Biowulf using Putty

Putty is an open source and grpahical based ssh client that is also capable of scp and sftp. It is one of the ways to interface with Biowulf for Windows users. To obtin Putty, goto https://www.putty.org *(https://www.putty.org)* (Figure 1). At the Putty website, click on the Download Putty link (Figure 1).

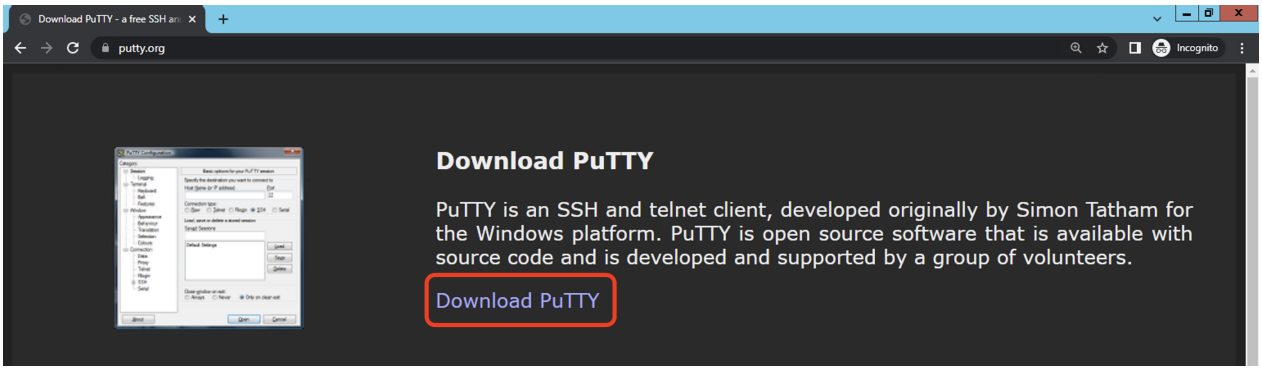


Figure 1

Subsequently, we will be taken to a page that houses several download options. To avoid having to install anything, we can grab the ".exe" files under the "Alternative binary files" section (Figure 2). Make sure to get the 64-bit x86 versions. Download putty.exe (the ssh client), pscp.exe (for scp), and psftp.exe (for sftp).

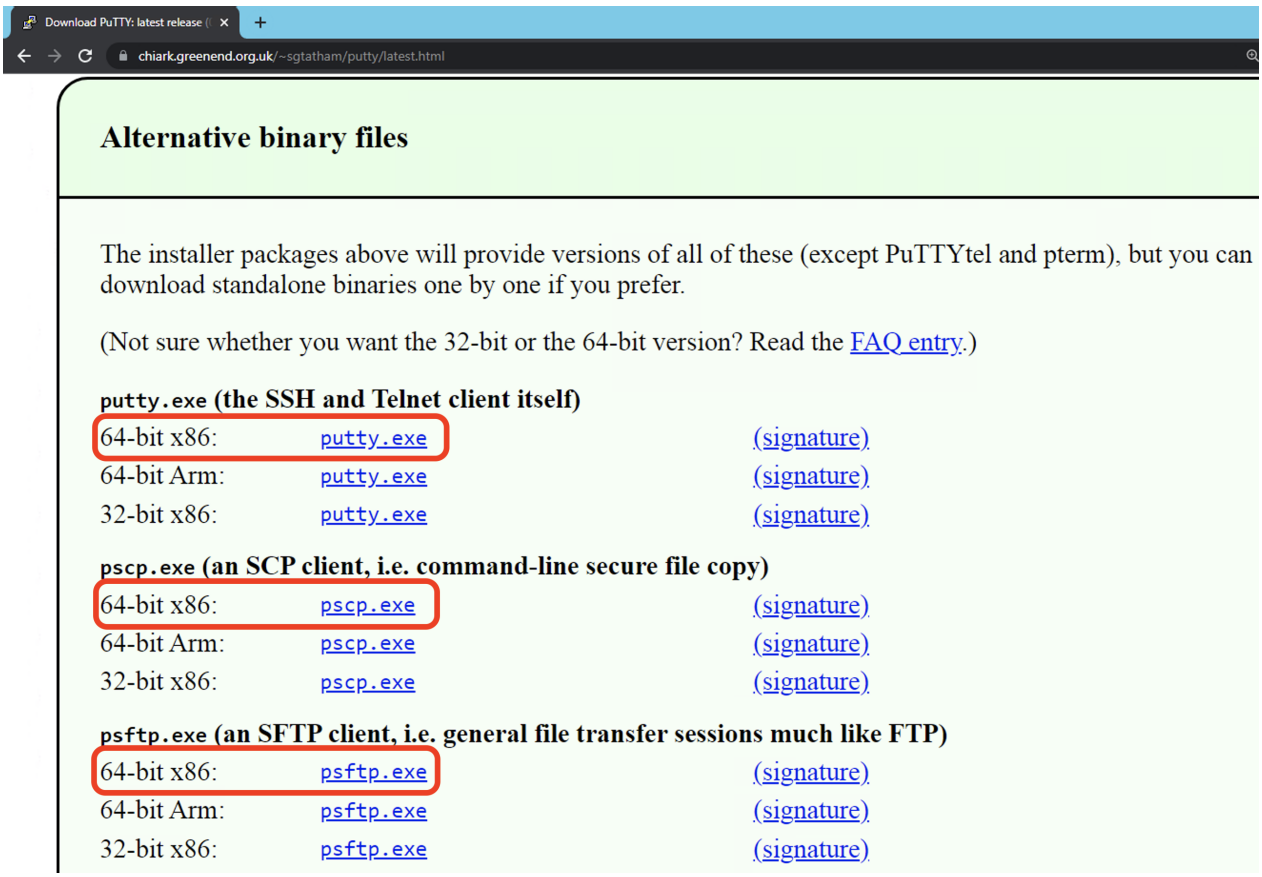


Figure 2

In this example, I have downloaded putty.exe, pscp.exe, and psftp.exe onto the Windows desktop.

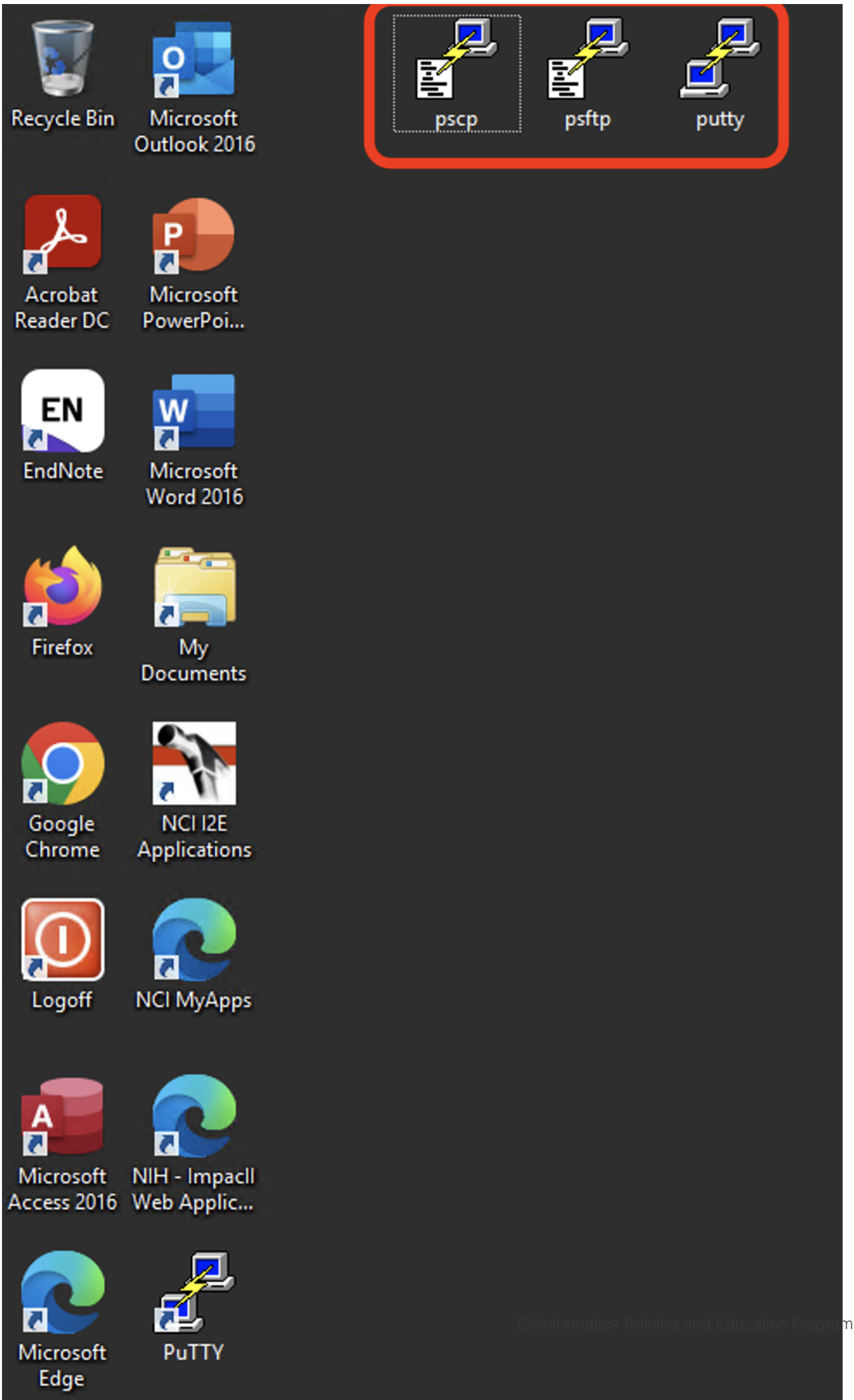

pscp
psftp
putty
Recycle Bin
Microsoft Outlook 2016
Acrobat Reader DC
Microsoft PowerPoi...
EndNote
Microsoft Word 2016
Firefox
My Documents
Google Chrome
NCI I2E Applications
Logoff
NCI MyApps
Microsoft Access 2016
NIH - ImpacII Web Applic...
Microsoft Edge
PuTTY

Figure 3

To connect to Biowulf, open putty and in the dialogue box that appears, enter biowulf.nih.gov in the box labeled "Host Name (or IP address), make sure the "Port" is set to 22, and that we choose SSH as the "Connection type". See Figure 4. Once the information has been entered, hit "Open".

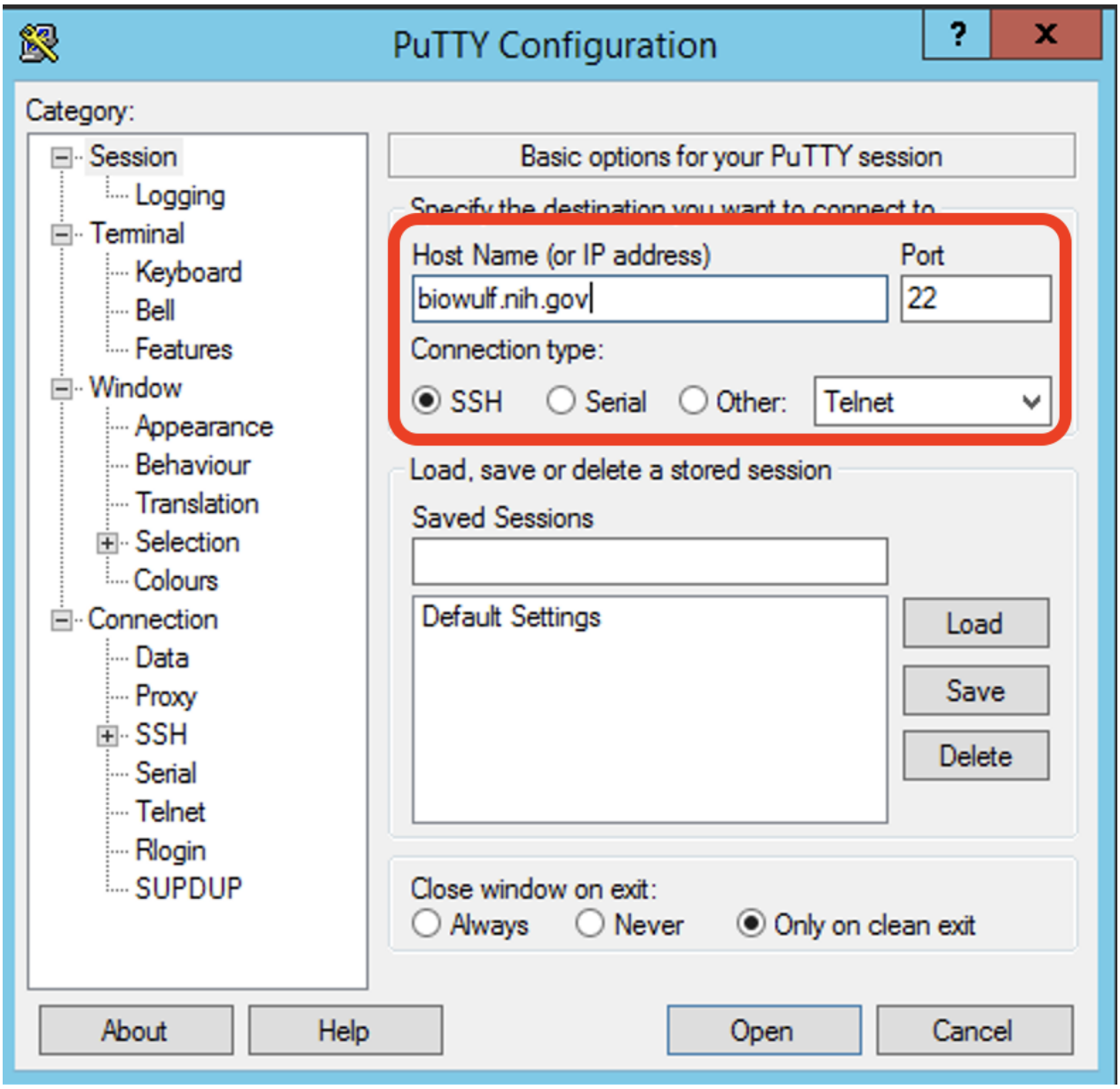


Figure 4

We will then be taken to a terminal where we entering our login credentials (Figur 5 and Figure 6)

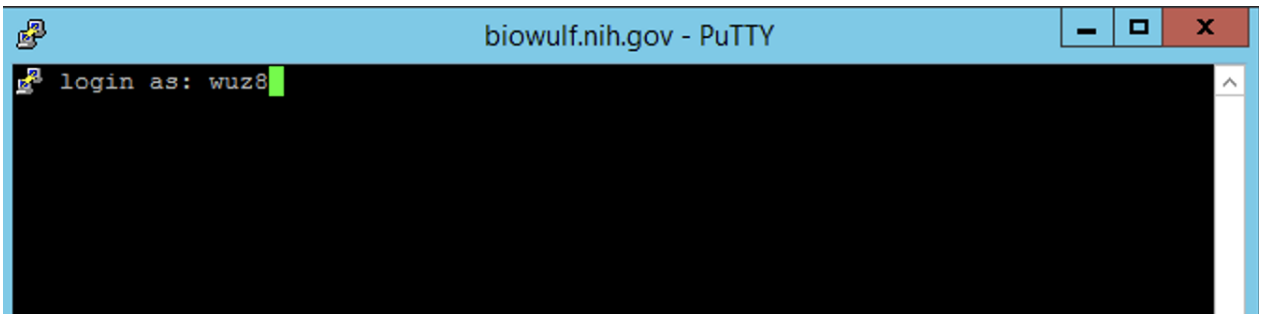


Figure 5

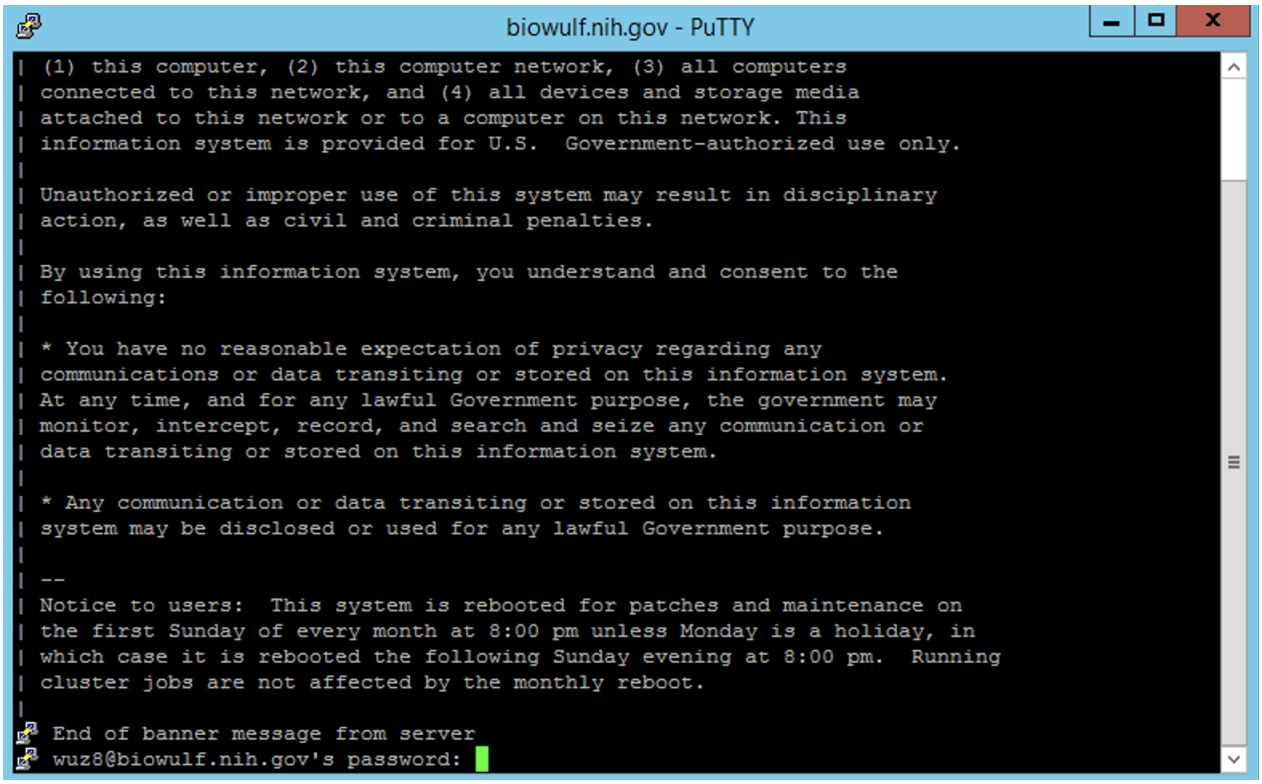


Figure 6

We will reach the Biowulf prompt after successfully logging in (Figure 7).

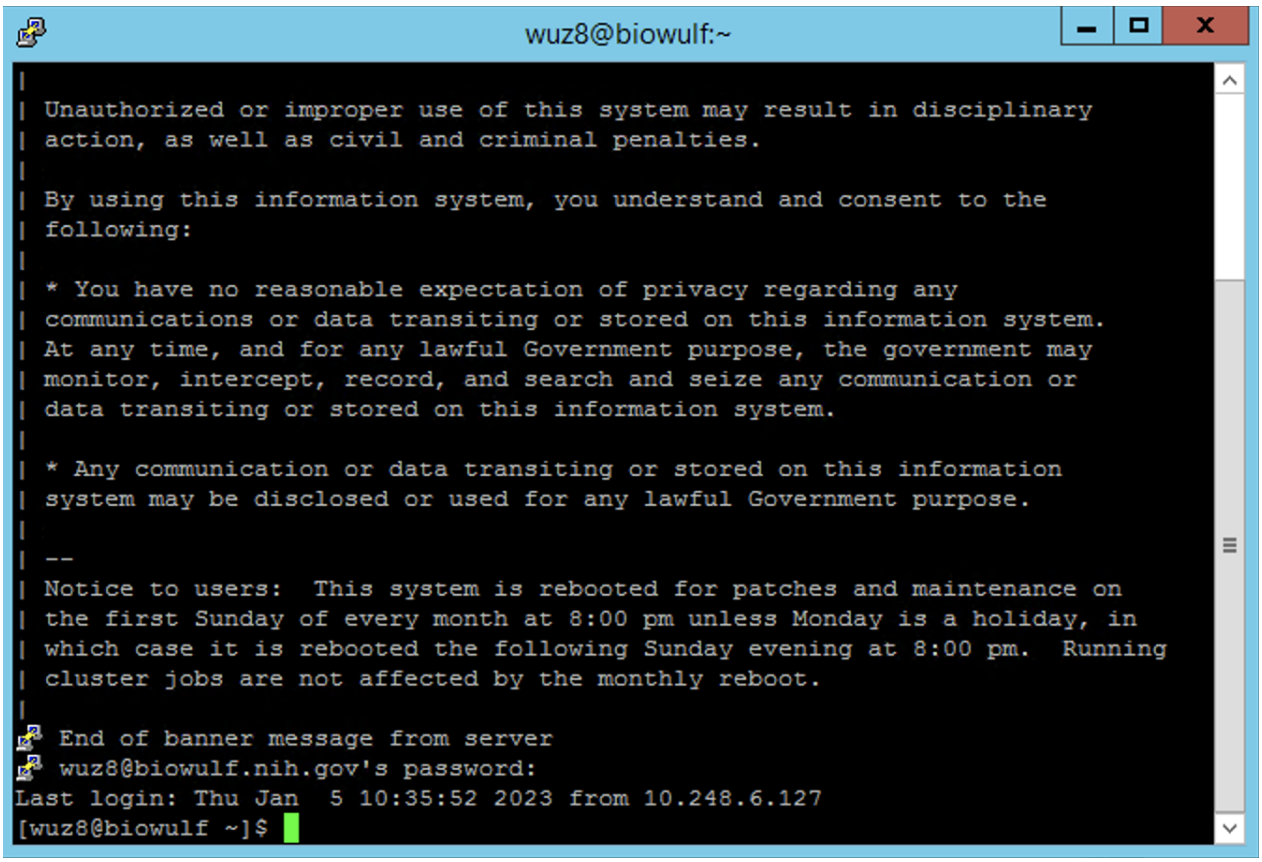


Figure 7

If we open a Windows Command Prompt, and change into the directory where pscp.exe was downloaded (in this case O:\Users\wuz8\Desktop), we can transfer files from Biowulf to our local machine using the Putty version of scp, which is pscp (Figure 8).

Figure 8

Interfacing with Biowulf using Mobaxterm

Mobaxterm is another open source and graphical based ssh client that Windows users can use to interact with Biowulf. To obtain Mobaxterm, goto https://mobaxterm.mobatek.net *(https://mobaxterm.mobatek.net)* and click the Download tab at the top (Figure 1).

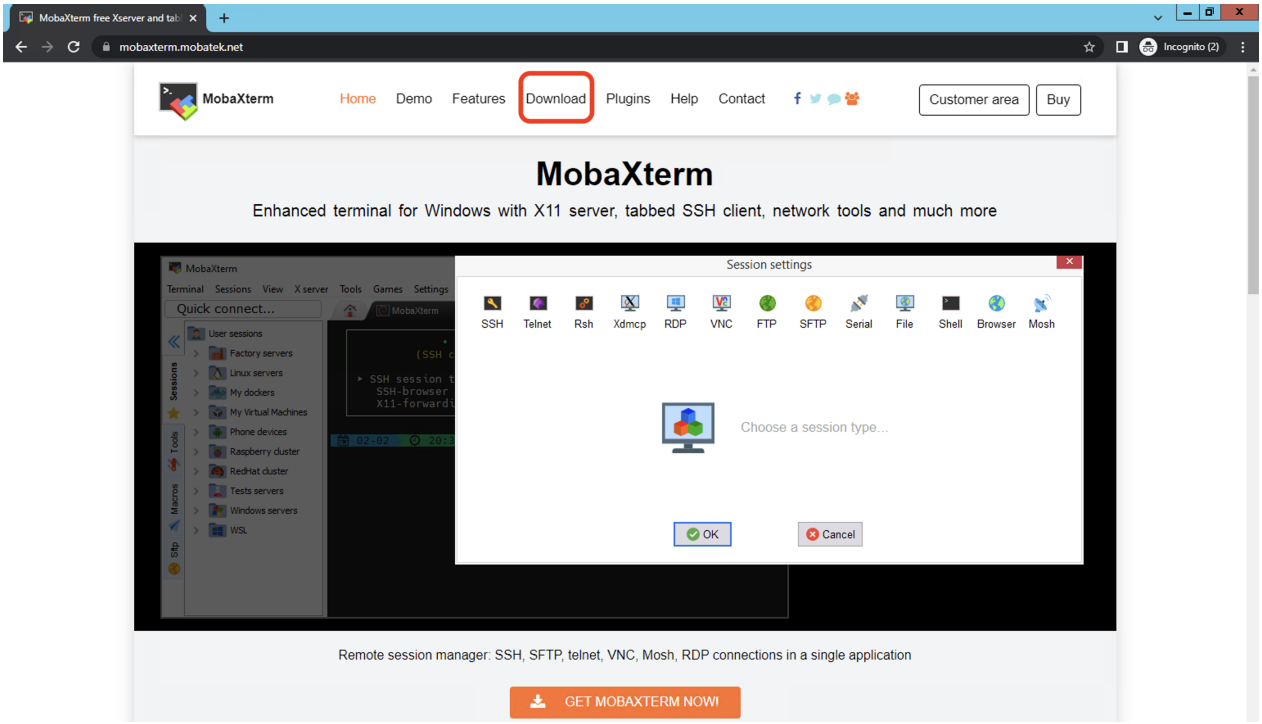


Figure 1

Subsequently, select to download the Home Edition, which is free (Figure 2).

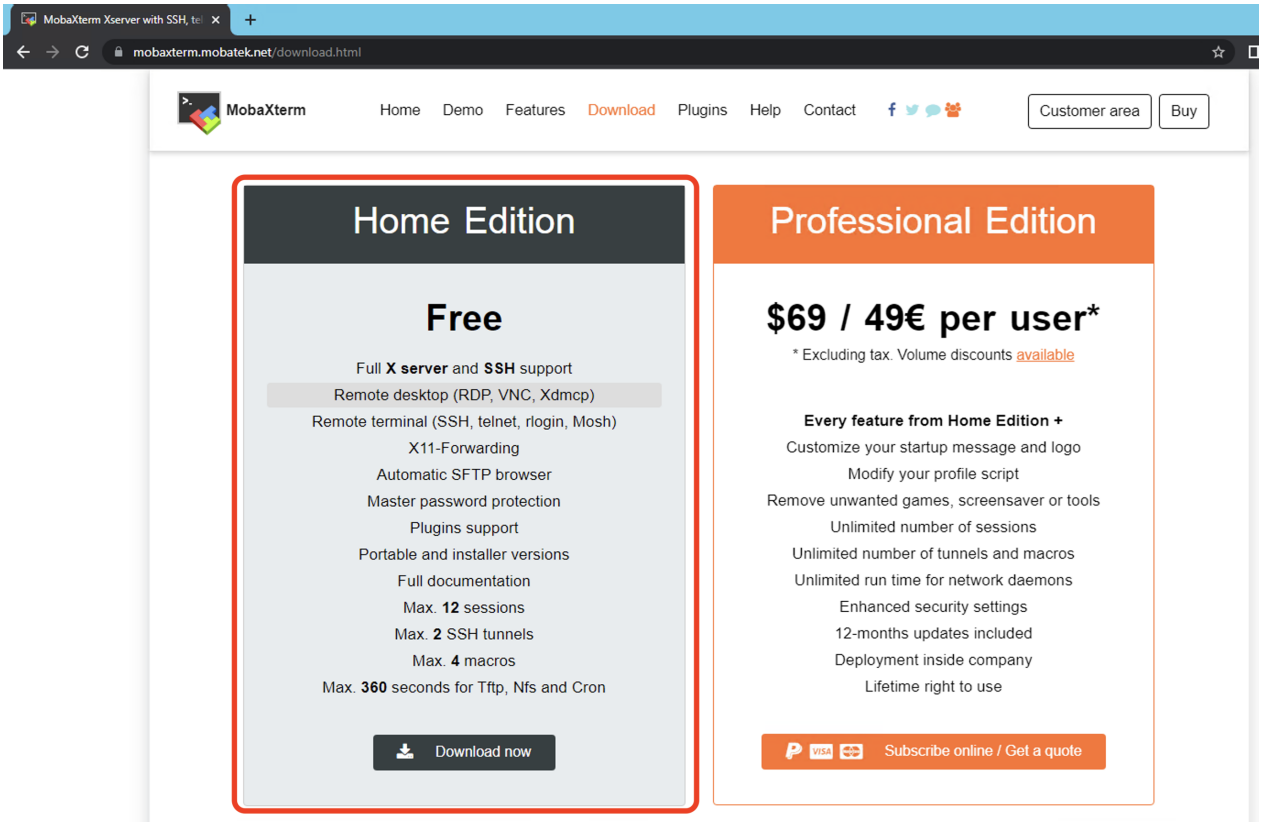


Figure 2

To avoid installing, choose to download the Portabl edition (Figure 3).

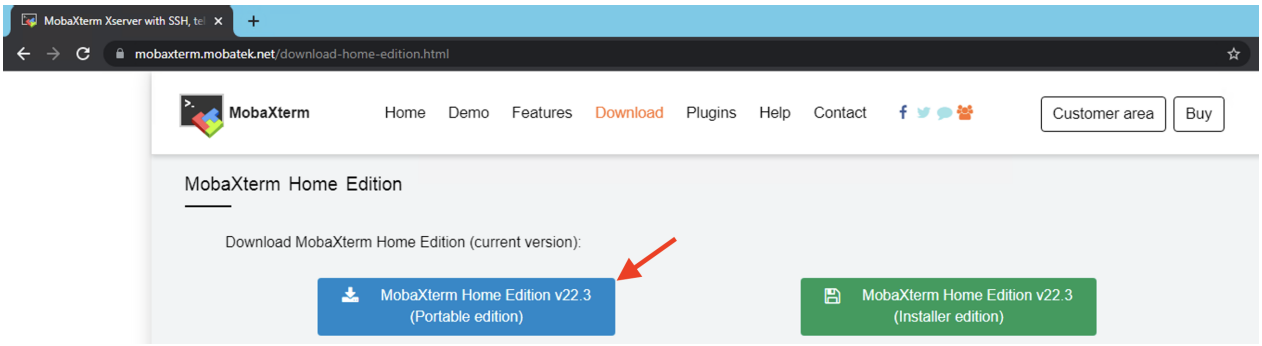


Figure 3

The Portable version was downloaded onto Windows desktop in zip folder (Figure 4).

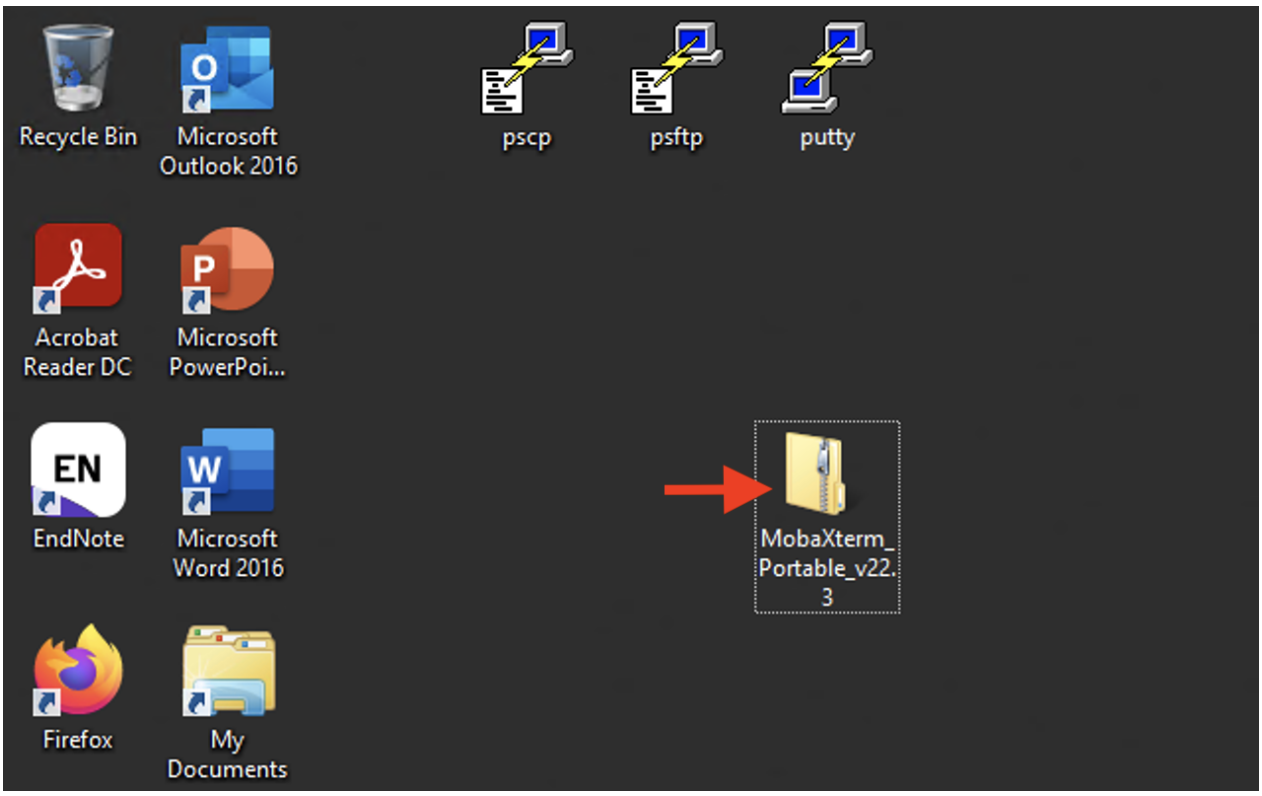


Figure 4

Right click on the zip folder and choose Extract All (Figure 5).

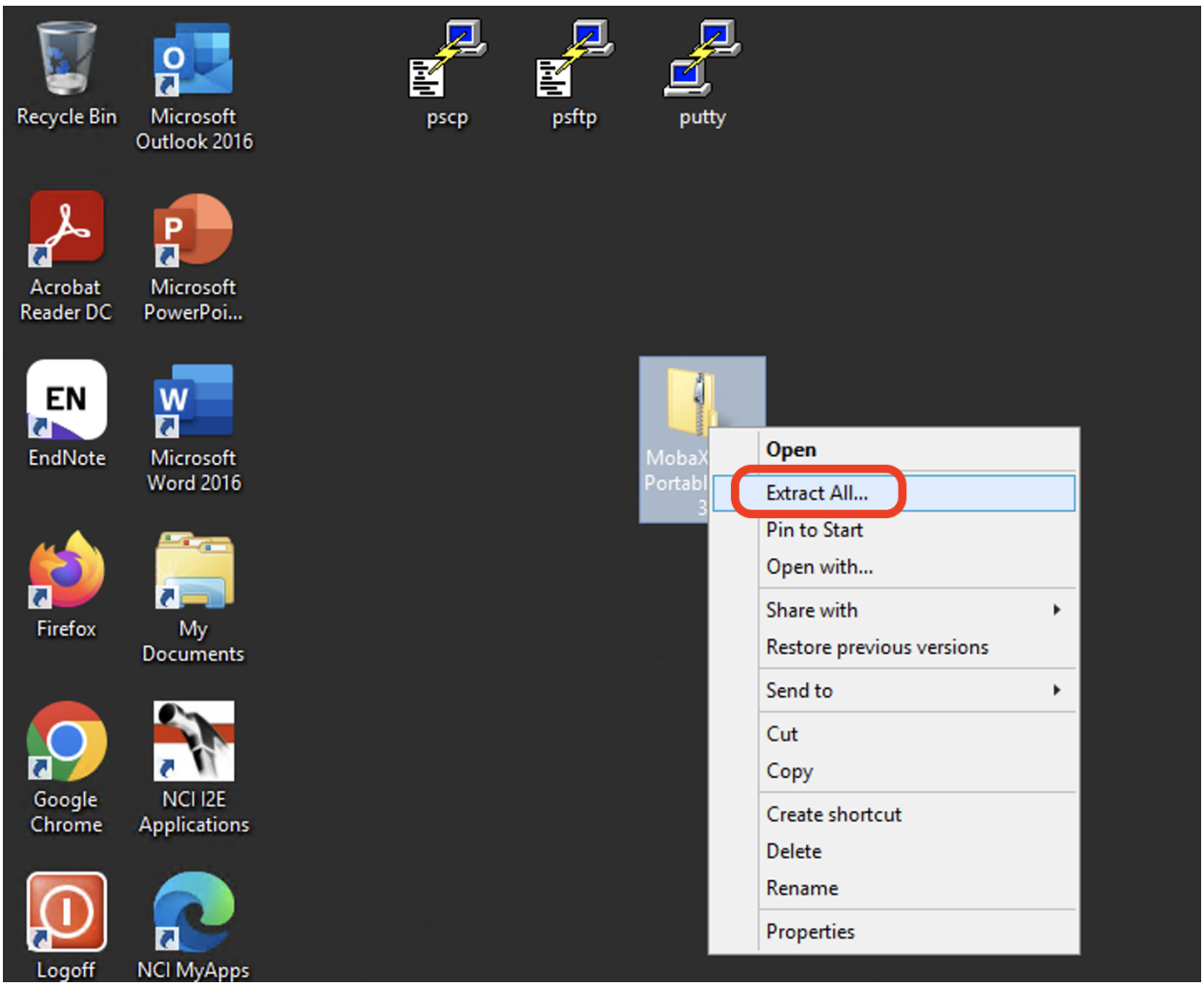


Figure 5

Confirm where you want the unzipped folder to go and then select Extract (Figure 6). In this example, the contents will be extracted onto the Windows desktop.

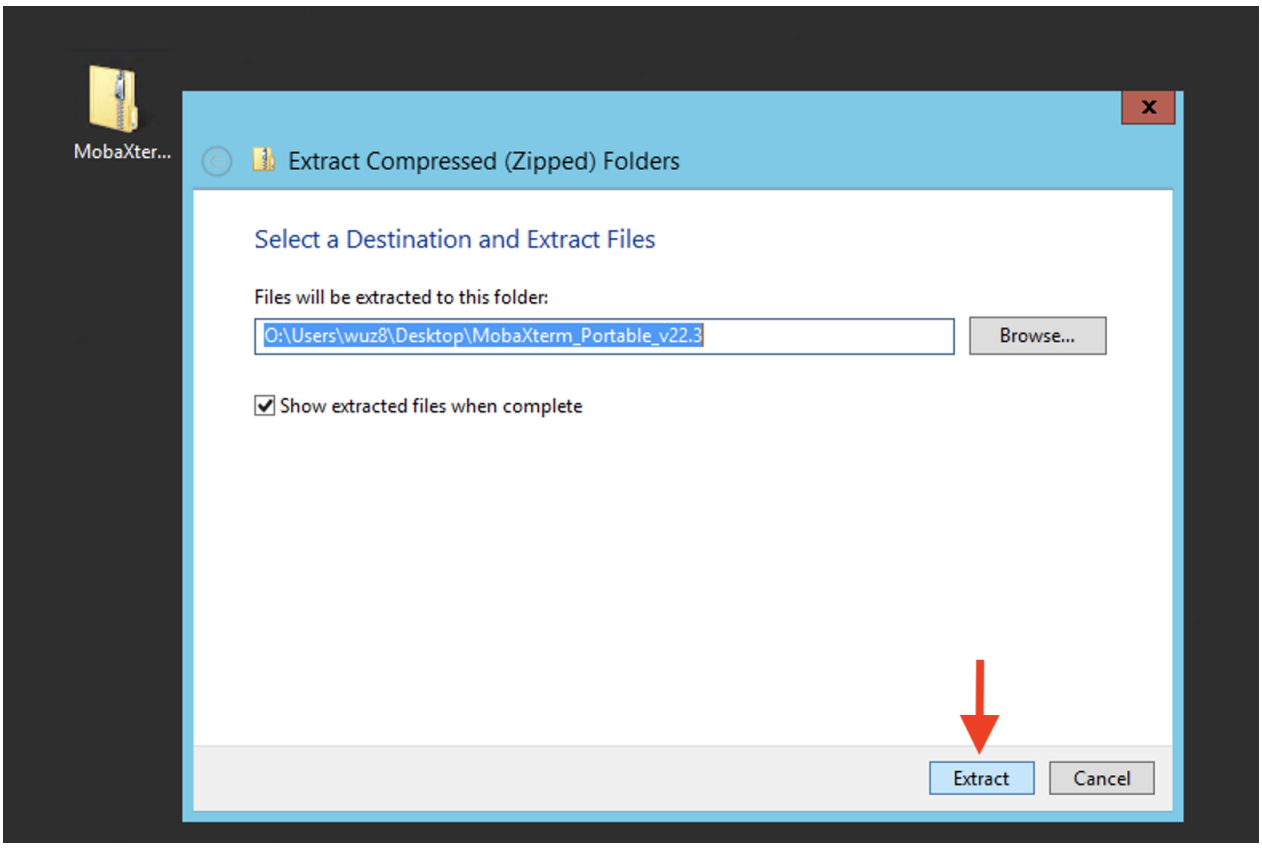


Figure 6

Once extracted, we will see a folder with the Mobaxterm icon (Figure 7).

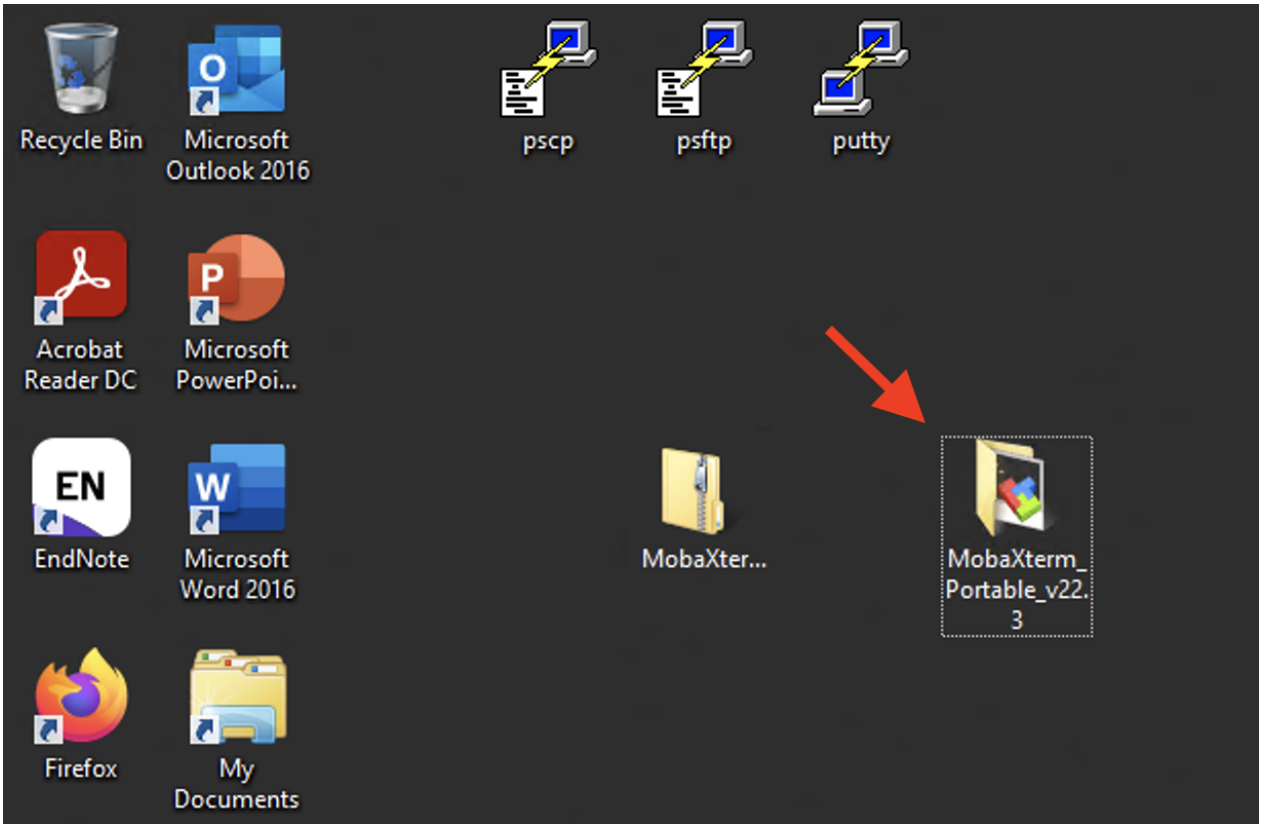


Figure 7

Open the unzipped folder and click on the Mobaxterm application file (MobaXterm_Personal_22.3) (Figure 8).

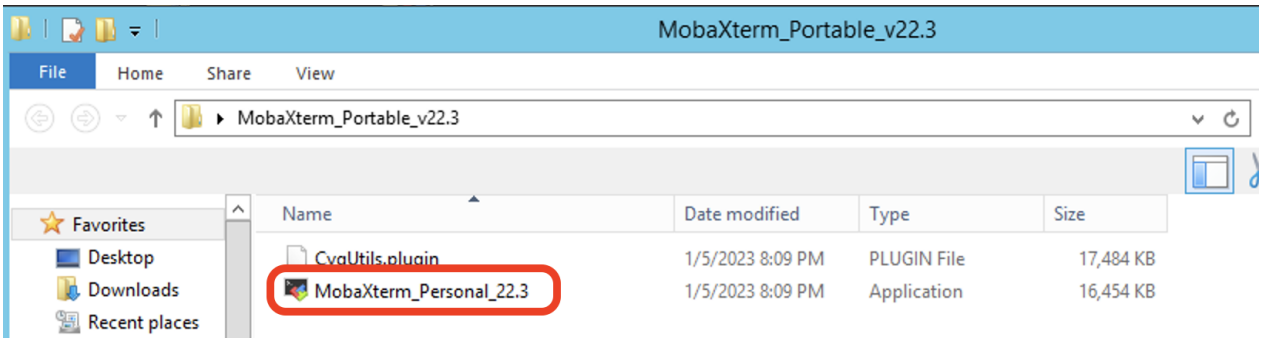


Figure 8

We will see the Mobaxterm client interface. To connect to Biowulf, we can click on Terminal and select "Open new tab" (Figure 9).

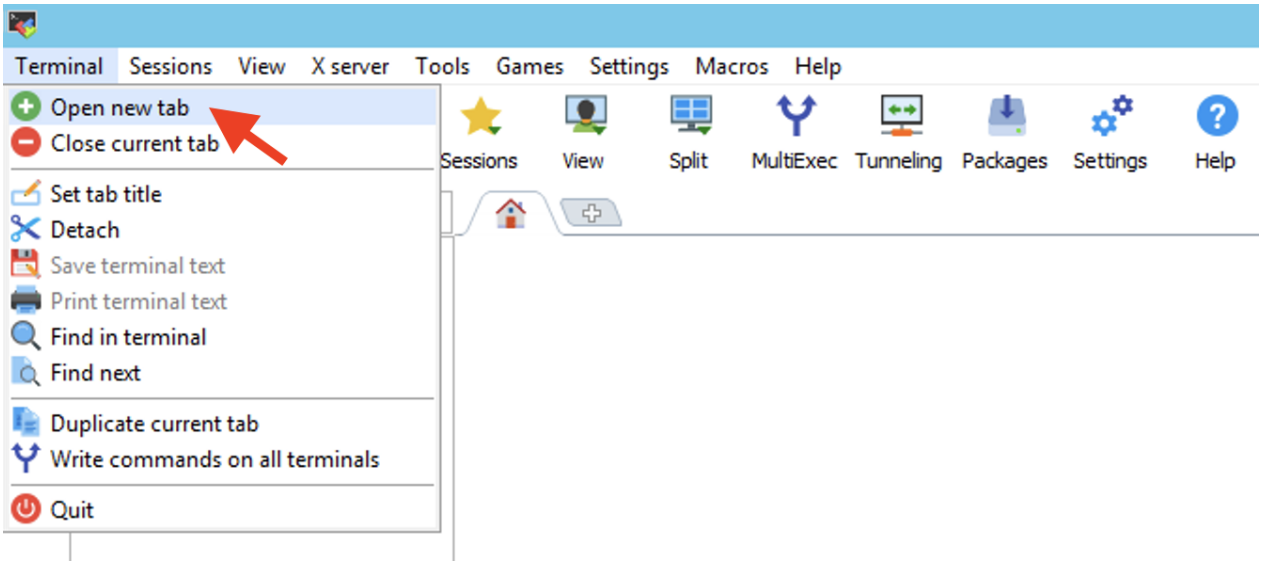


Figure 9

A local Unix terminal should open. From here, we can use the `ssh` command to connect to Biowulf (Figure 10).

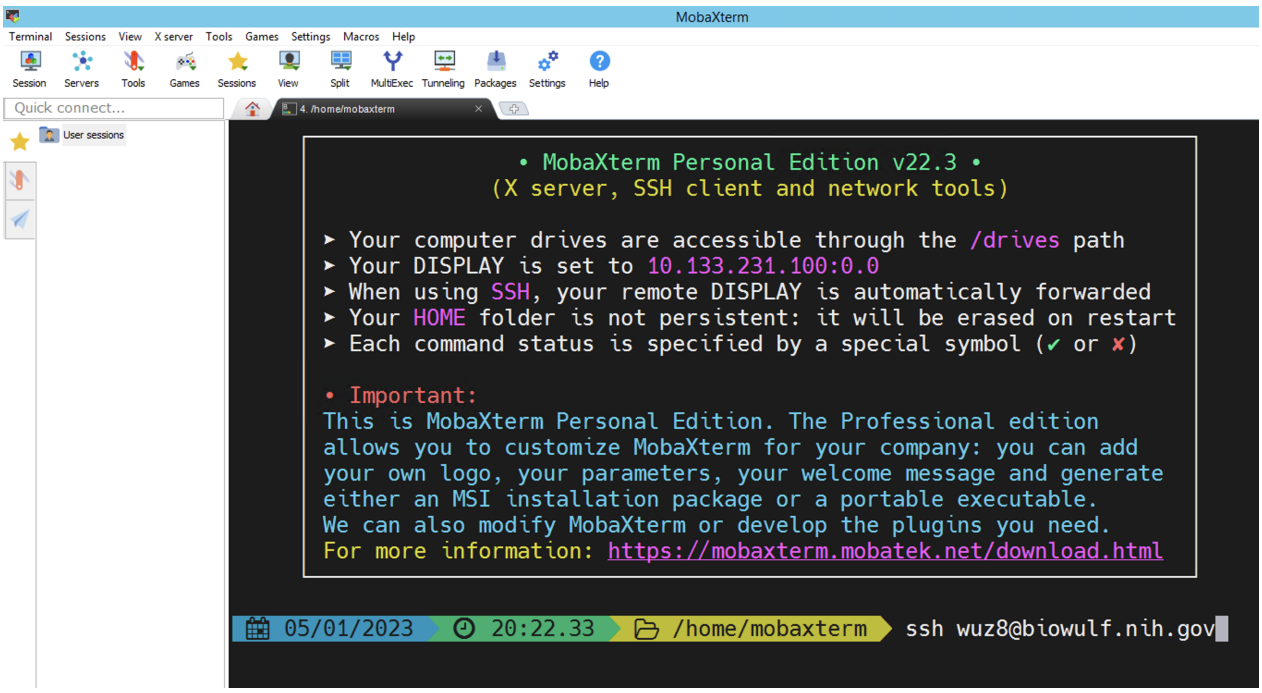


Figure 10

When asked whether we want to continue, select yes (Figure 11). The message in Figure 11 appears when your machine connects to Biowulf for the first time.

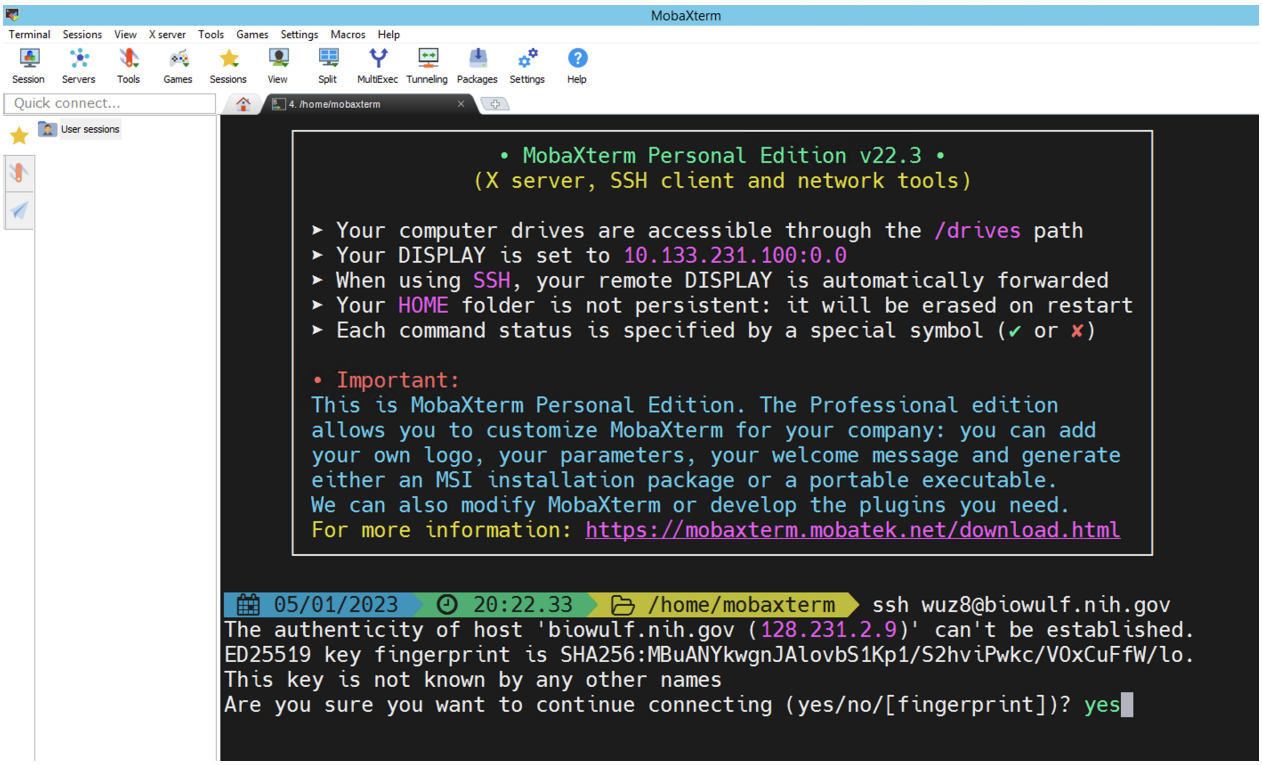


Figure 11

Enter Biowulf password (Figure 12).

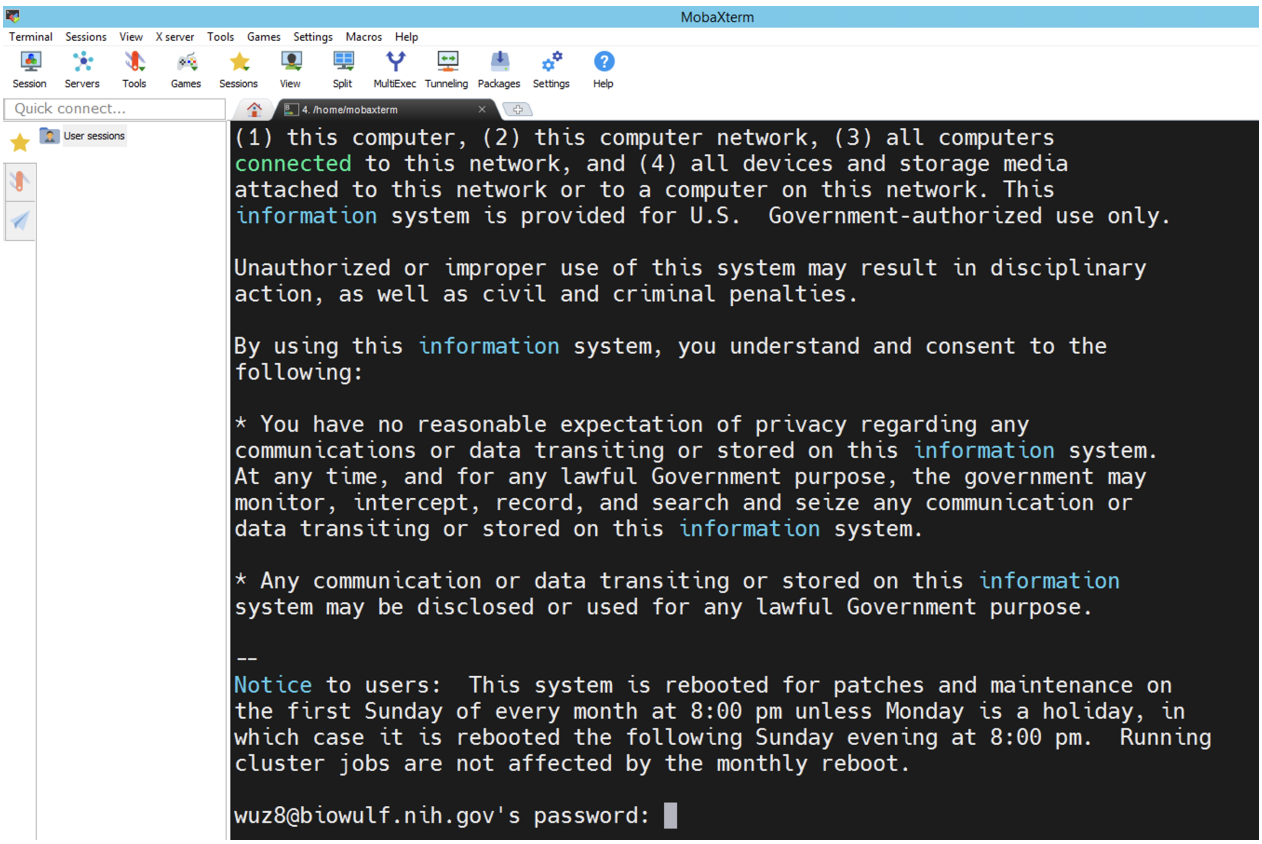


Figure 12

Hit No to not save the password (Figure 13) and we should land in our Biowulf terminal prompt.

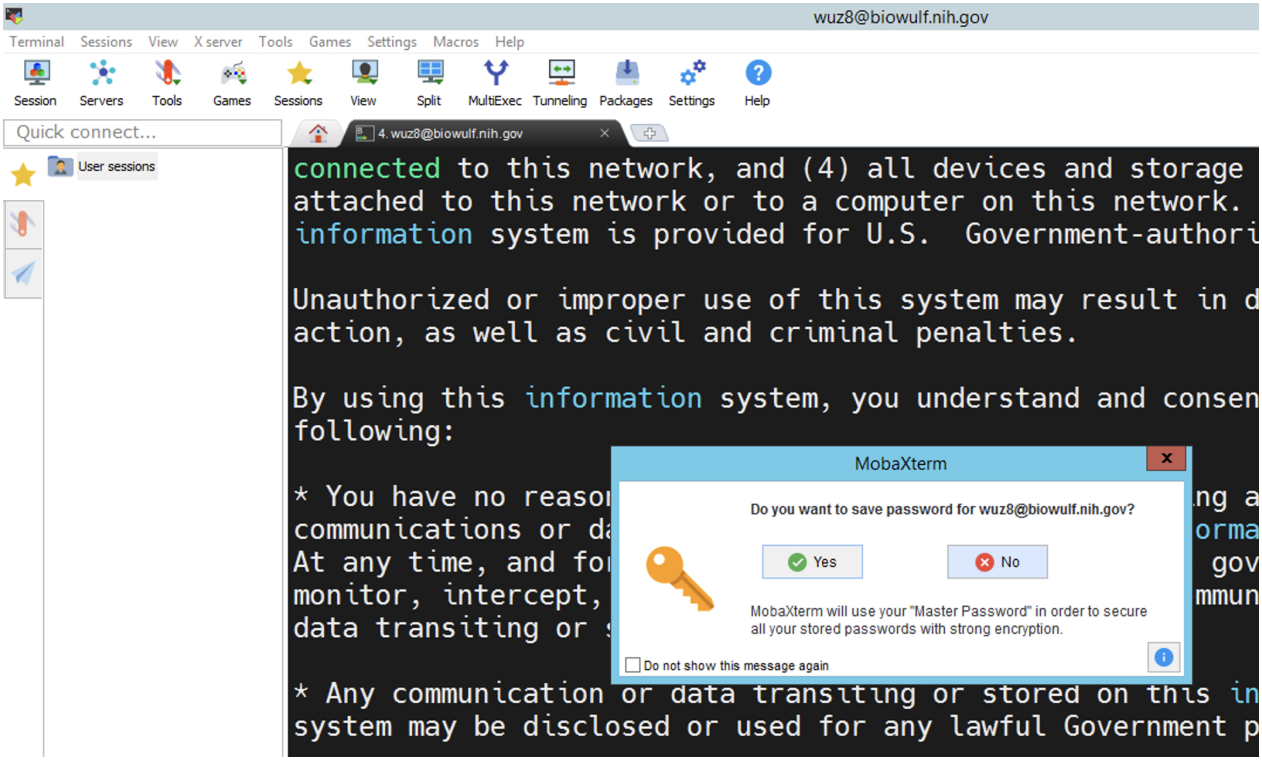
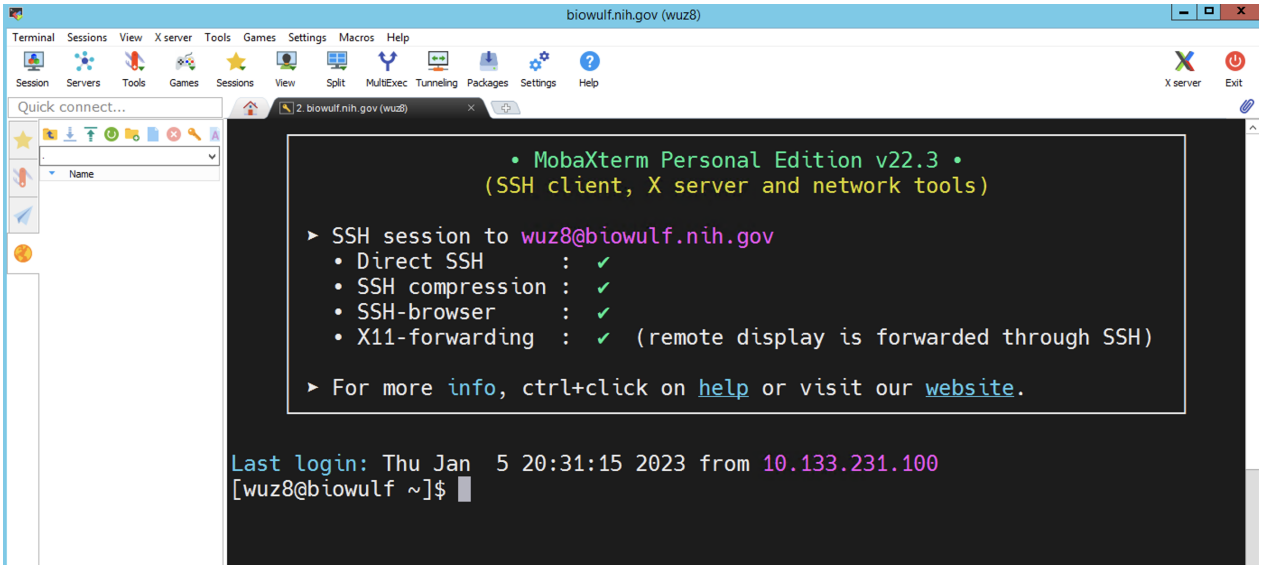


Figure 13



Mobaxterm - Biowulf connection successful

An alternative method for connecting to Biowulf is to select the Session tab (Figure 14) and choose to start a new SSH session (Figure 15).

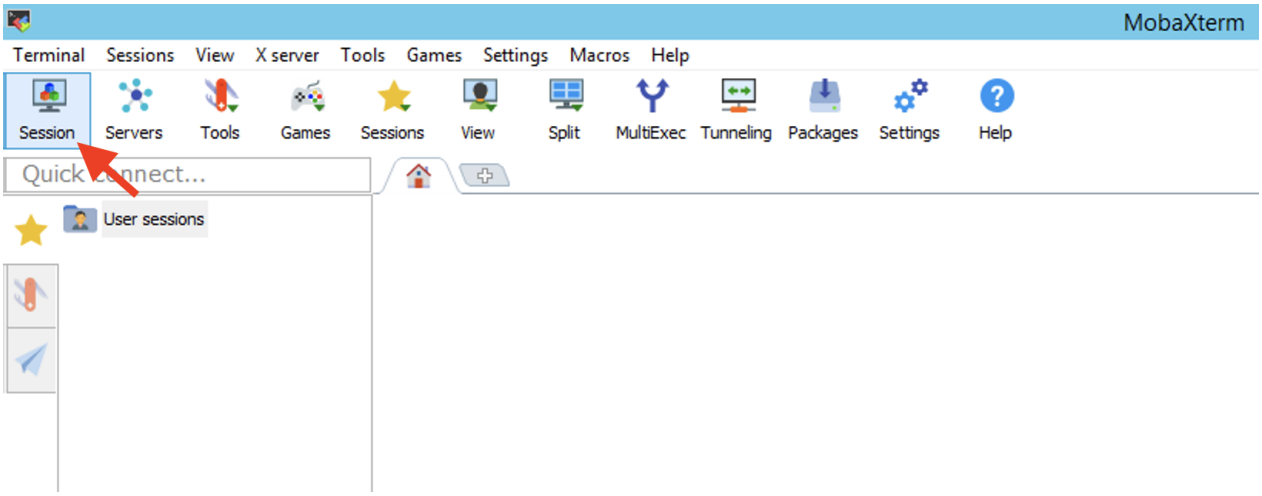


Figure 14

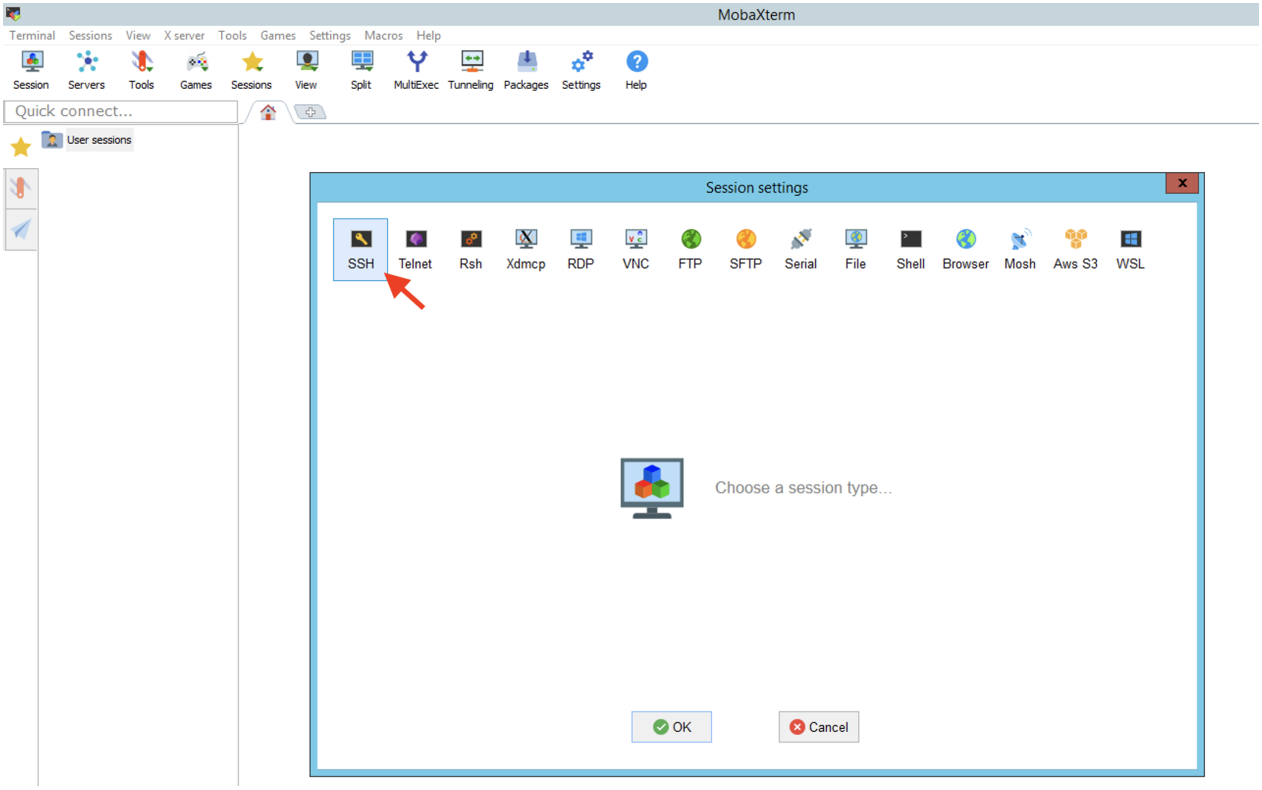


Figure 15

In the basic setting box, enter the Remote host (biowulf.nih.gov), followed by the username (remember to check the box Specify username), and again, stay on Port 22 (Figure 16).

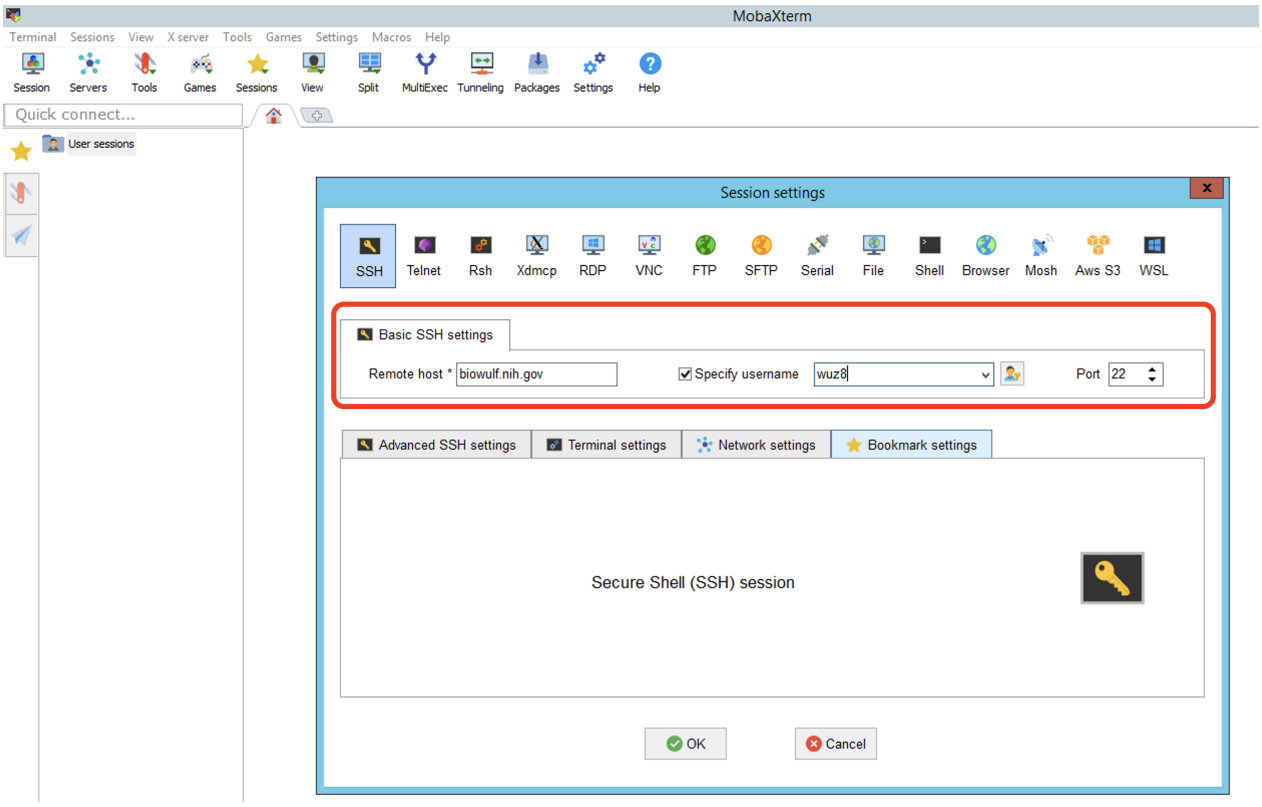


Figure 16

If this is the first time conecting to Biowulf, accept the certificate shown in Figure 17 and we will be taken to our Biowulf terminal prompt.

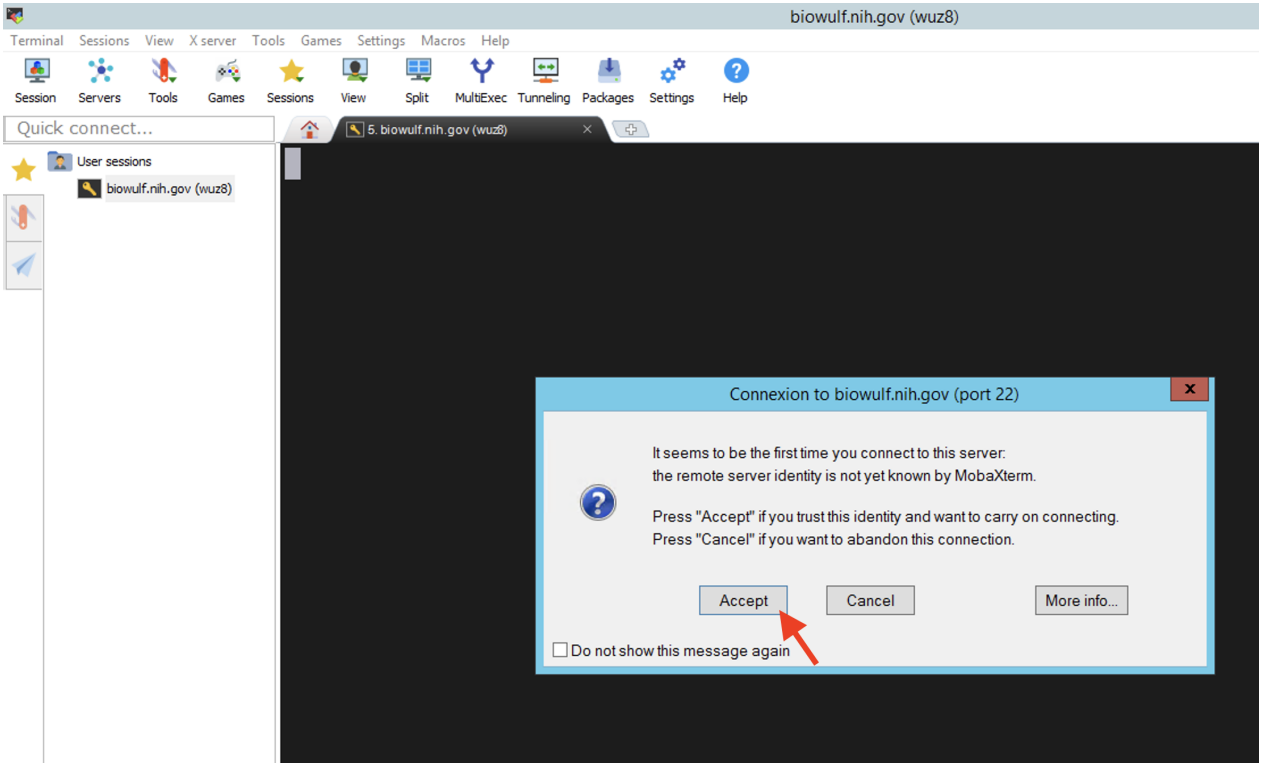
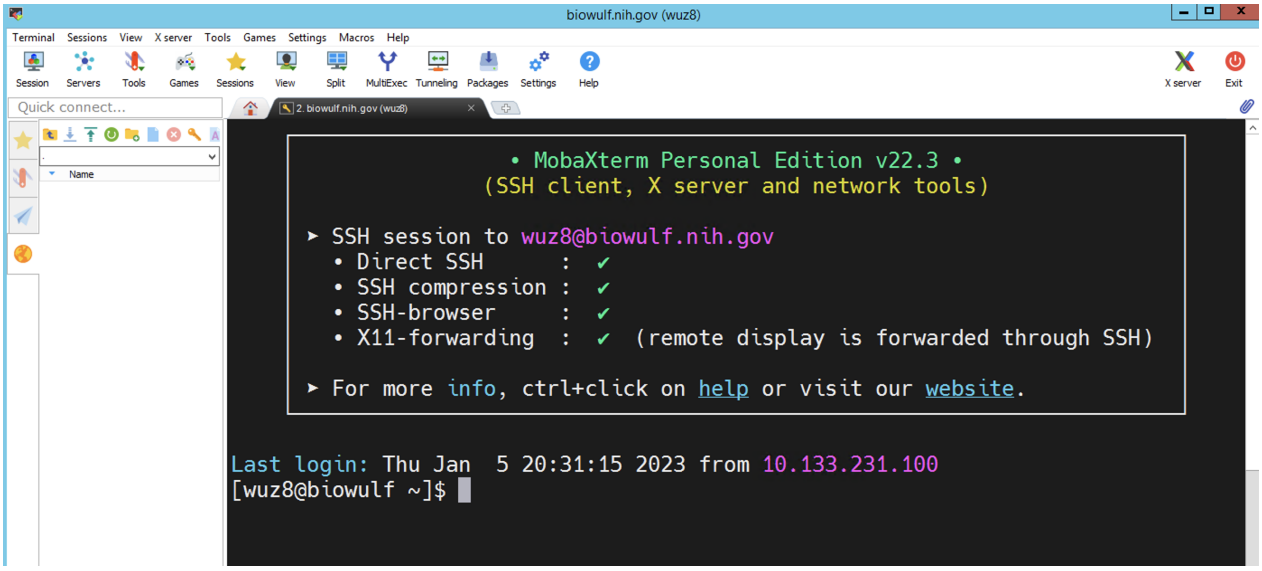


Figure 17



Mobaxterm - Biowulf connection successful

At the local terminal, we can use the `scp` command to transfer data from Biowulf to local and the other way around (Figure 18). See Figure 9 for opening a new local terminal.

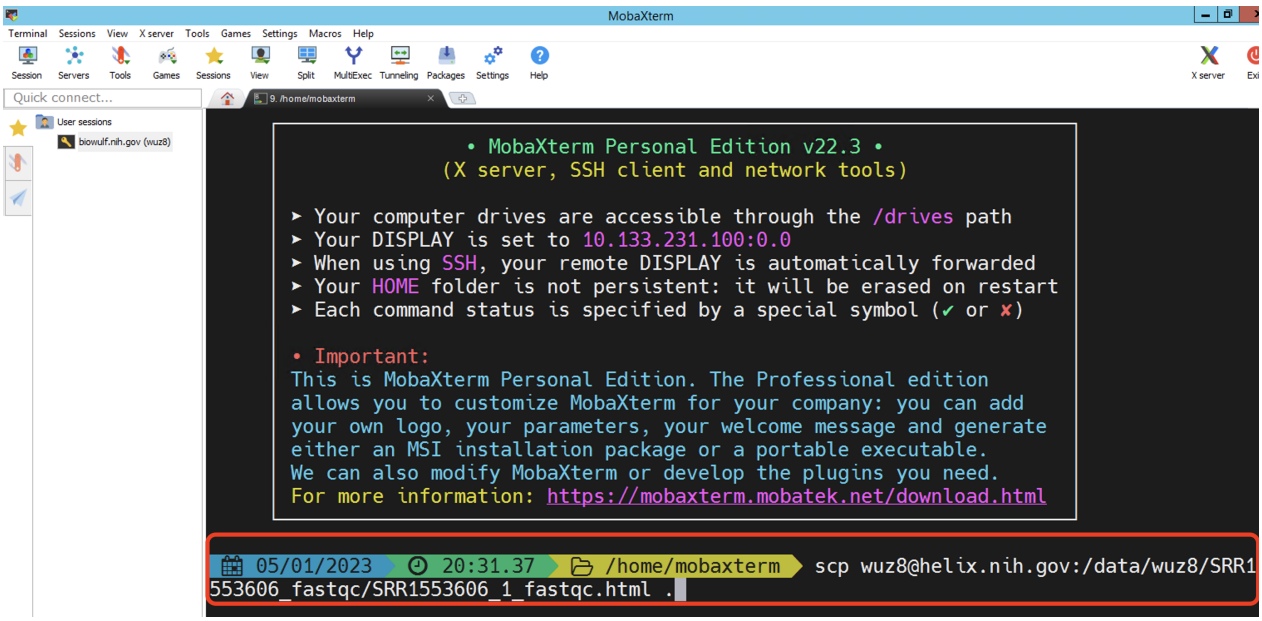


Figure 18

When clicking on the Session tab, we can also request a sftp session to help with data transfer (Figure 19).

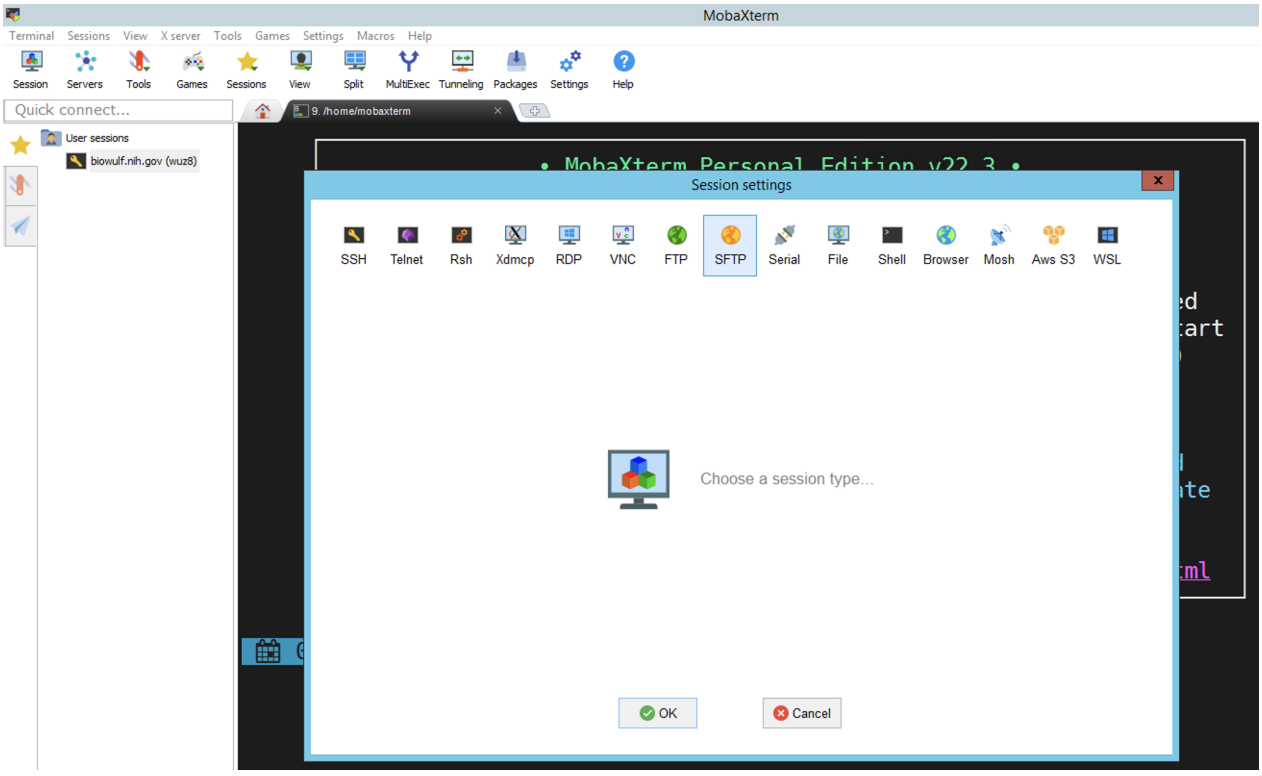


Figure 19

Again, provide the Remote host (helix.nih.gov for file transfer), followed by username, and remember to stay on Port 22 (Figure 20). Hit ok after the information has been entered.

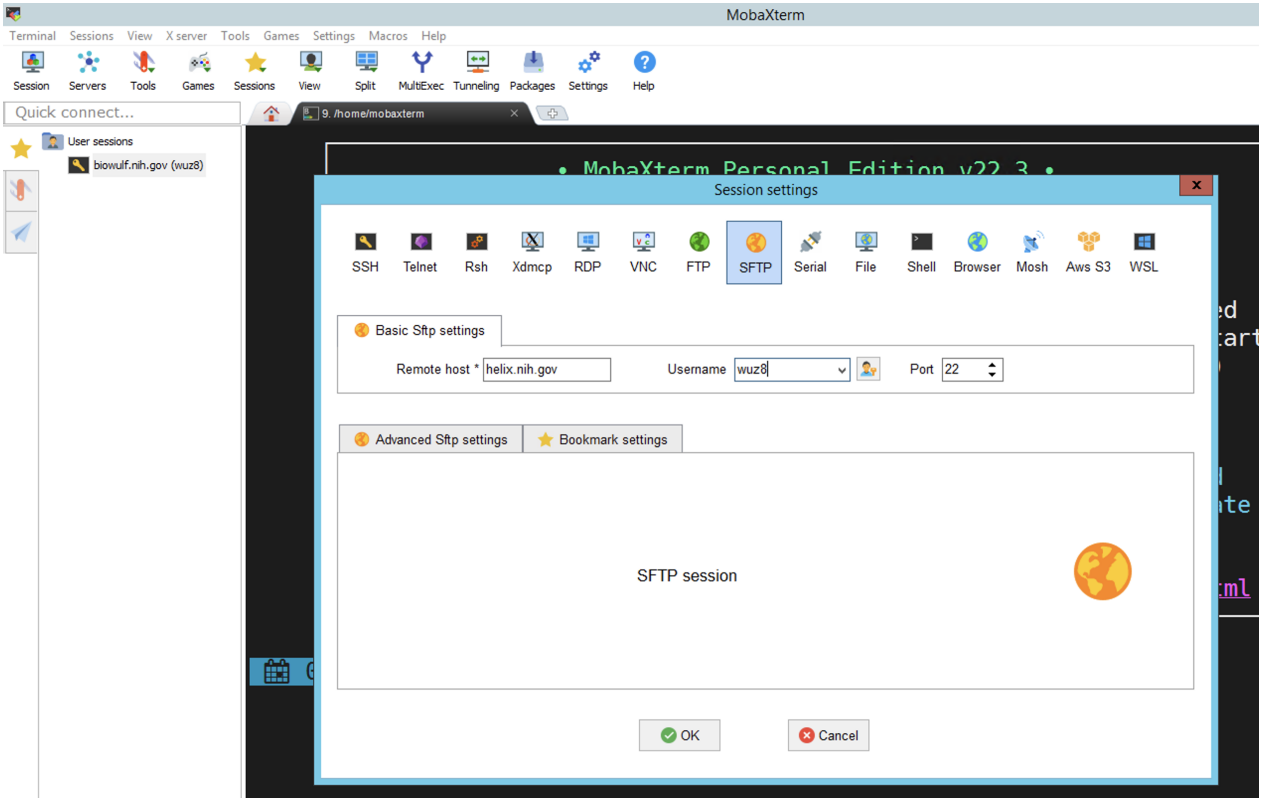


Figure 20

We will see the message to accept the certificate because its our first time logging in (go ahead and hit Accept) (Figure 21)

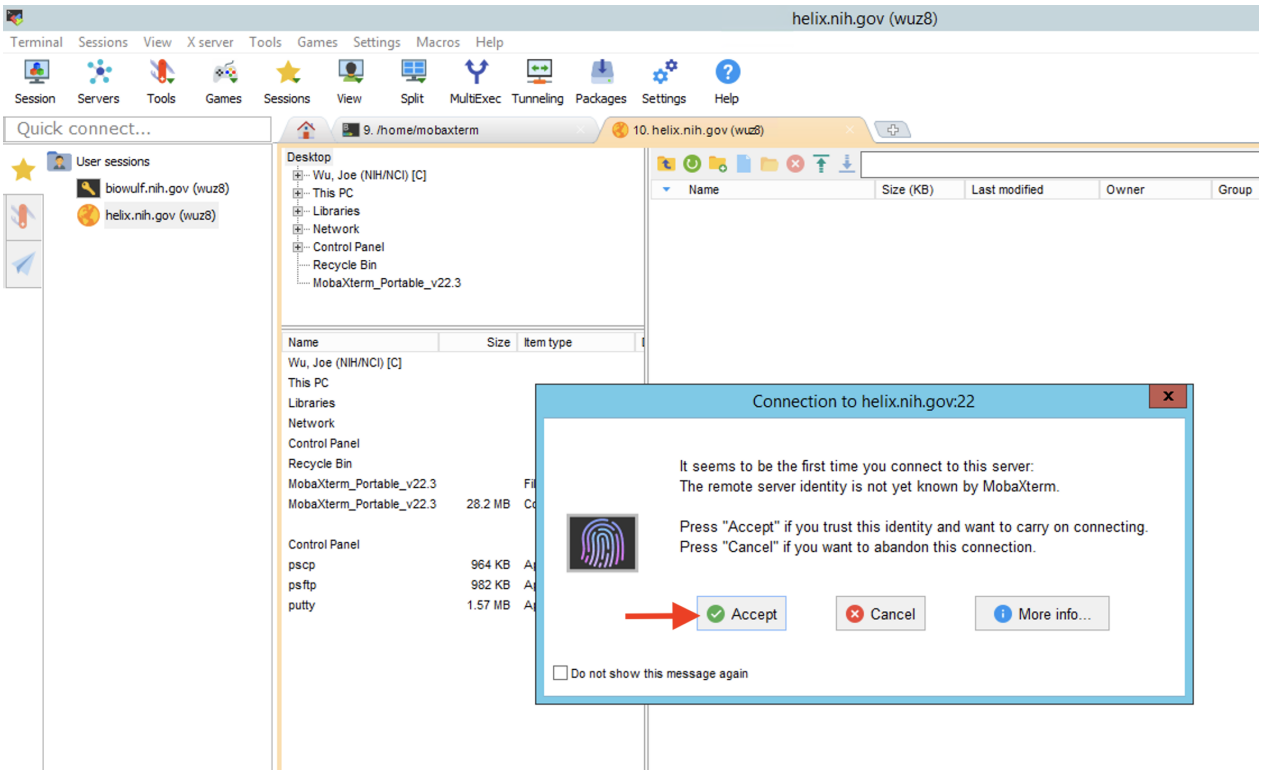


Figure 21

This will take us to an interface where on the right side we can navigate the directories in our Helix/Biowulf account and on the left, we have our local directories and files. We can then drag and drop files from local to Biowulf or from Biowulf to local (Figure 22).

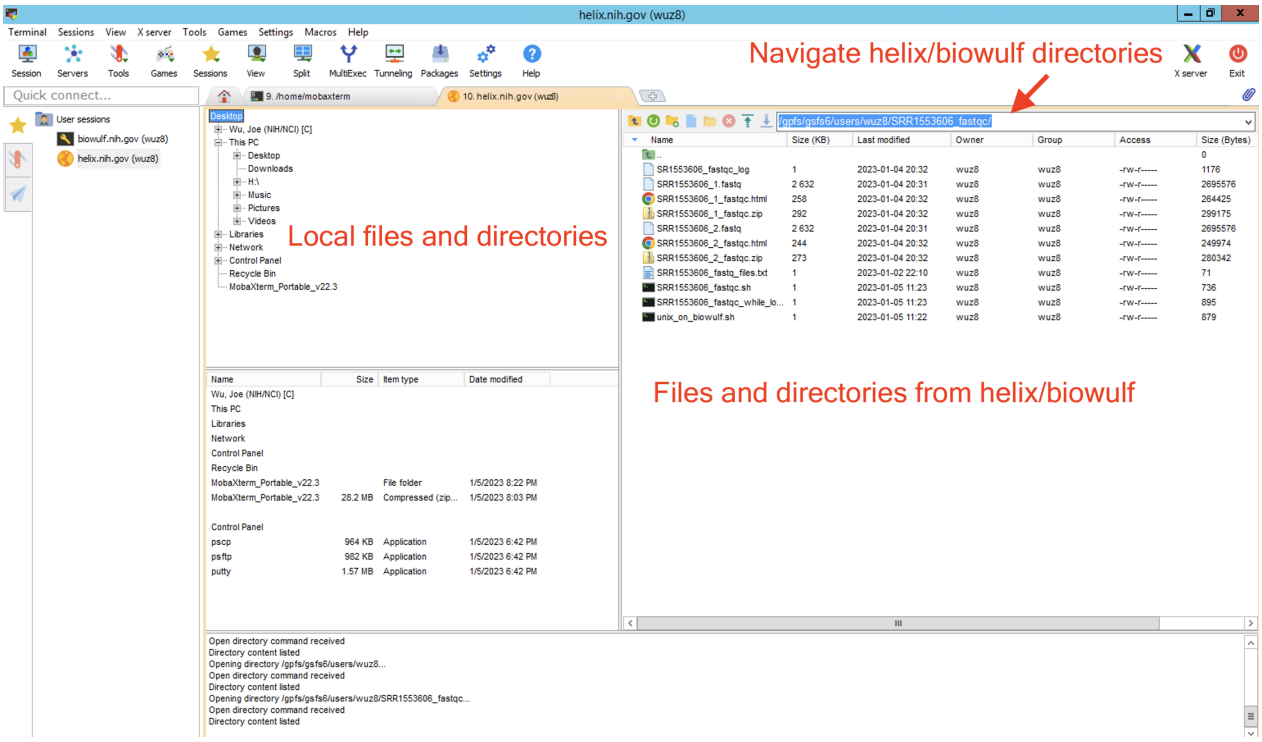


Figure 22

Interfacing with Biowulf using Fugu

Fugu is an open source and graphical based application for Mac users that can be used for data transfer between local and high performance compute systems such as Biowulf. To obtain Fugu, refer to the instruction from the Biowulf website for GUI file transfer applications *(https://hpc.nih.gov/docs/transfer.html#GUI)*.

> "Fugu is a graphical frontend to the commandline Secure File Transfer application (SFTP). SFTP is similar to FTP, but unlike FTP, the entire session is encrypted, meaning no passwords are sent in cleartext form, and is thus much less vulnerable to third-party interception. Fugu allows you to take advantage of SFTP's security without having to sacrifice the ease of use found in a GUI. Fugu also includes support for SCP file transfers, and the ability to create secure tunnels via SSH.

- Download Fugu from the U. Mich. Fugu website.
- For OSX 10.5 and above, download from cnet.com.
- Doubleclick on the downloaded Fugu_xxxx.dmg file to open. A small window with the Fugu icon will appear" -- Biowulf GUI file transfer applications *(https://hpc.nih.gov/docs/transfer.html#GUI)*.

Upon opening Fugu, we will see two panels. One allows us to navigate our local directories and files while the other allows us to connect to a remote host (ie. Helix/Biowulf) (Figure 1). Following Figure 1, do the following to connect to Helix

- Enter helix.nih.gov in the box that says Connect to
- Enter Helix/Biowulf username (mine is wuz8 so that is what we see)
- Make sure that Port is set to 22
- Specify the directory in Helix/Biowulf that we like to goto (ie. /data/wuz8, which is my data directory)
- Hit connect when done entering credentials

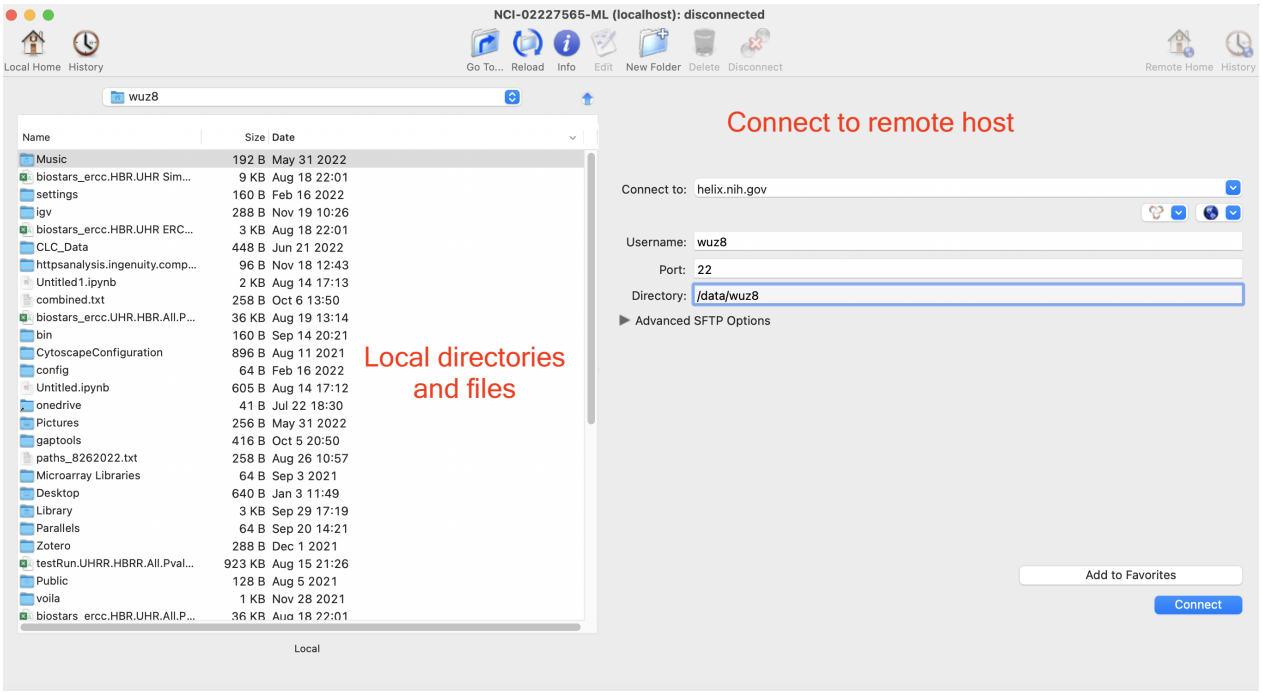


Figure 1

At the next screen, enter the password used to log into Helix/Biowulf and click the Authenticate button (Figure 2).

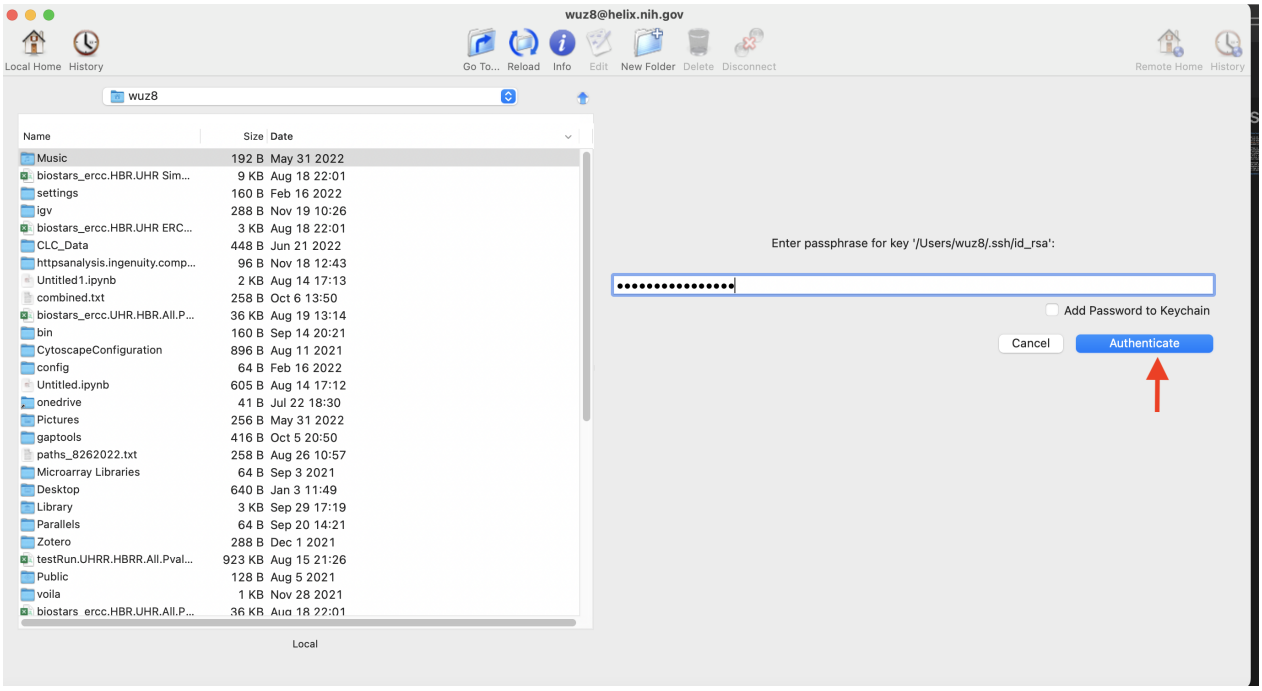


Figure 2

Once signed in to Helix, we will see our Helix/Biowulf directories and files in one panel and our local directories and files in another panel. From here we can select and then drag and drop either from Helix/Biowulf to local or from local to Helix/Biowulf (Figure 3).

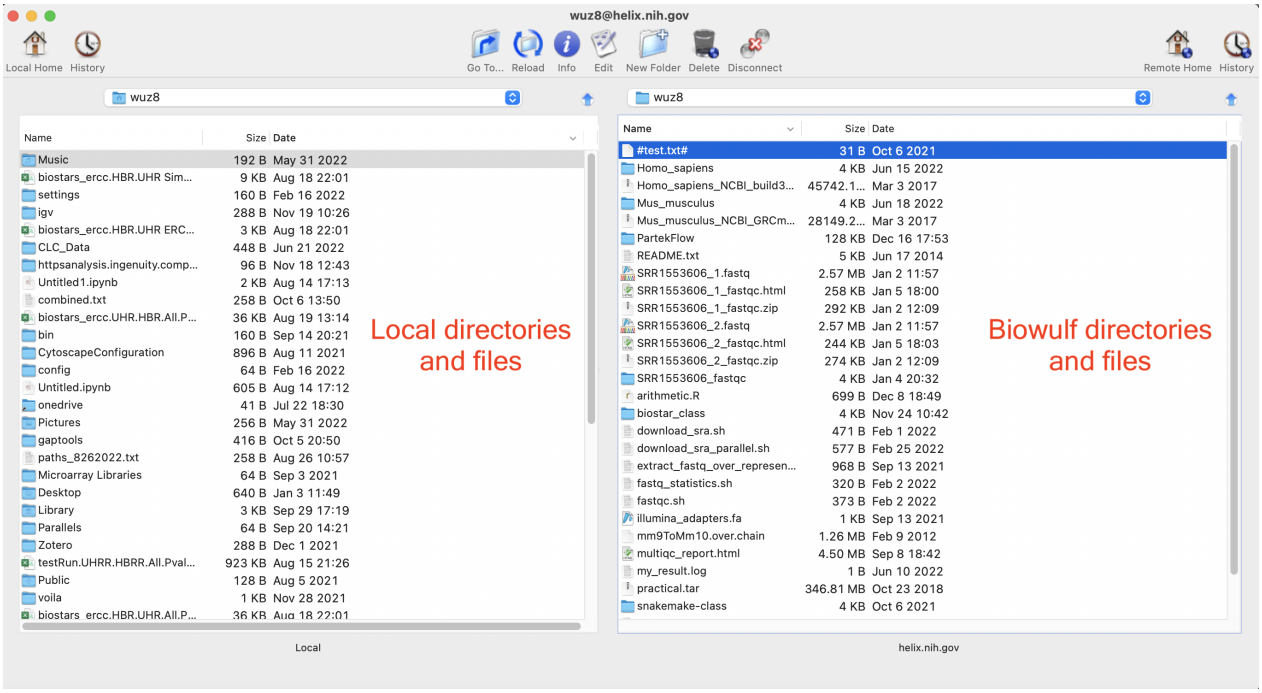

wuz8@helix.nih.gov
Local Home
History
Go To...
Reload
Info
Edit
New Folder
Delete
Disconnect
Remote Home
History
wuz8
wuz8
Name
Size
Date
Music 192 B May 31 2022
biostars_ercc.HBR.UHR Sim... 9 KB Aug 18 22:01
settings 160 B Feb 16 2022
igv 288 B Nov 19 10:26
biostars_ercc.HBR.UHR ERC... 3 KB Aug 18 22:01
CLC_Data 448 B Jun 21 2022
httpsanalysis.ingenuity.comp... 96 B Nov 18 12:43
Untitled1.ipynb 2 KB Aug 14 17:13
combined.txt 258 B Oct 6 13:50
biostars_ercc.UHR.HBR.All.P... 36 KB Aug 19 13:14
bin 160 B Sep 14 20:21
CytoscapeConfiguration 896 B Aug 11 2021
config 64 B Feb 16 2022
Untitled.ipynb 605 B Aug 14 17:12
onedrive 41 B Jul 22 18:30
Pictures 256 B May 31 2022
gaptools 416 B Oct 5 20:50
paths_8262022.txt 258 B Aug 26 10:57
Microarray Libraries 64 B Sep 3 2021
Desktop 640 B Jan 3 11:49
Library 3 KB Sep 29 17:19
Parallels 64 B Sep 20 14:21
Zotero 288 B Dec 1 2021
testRun.UHRR.HBRR.All.Pval... 923 KB Aug 15 21:26
Public 128 B Aug 5 2021
voila 1 KB Nov 28 2021
biostars_ercc.HBR.UHR.All.P... 36 KB Aug 18 22:01
Local directories and files
Local
Name
Size
Date
#test.txt# 31 B Oct 6 2021
Homo_sapiens 4 KB Jun 15 2022
Homo_sapiens_NCBI_build3... 45742.1... Mar 3 2017
Mus_musculus 4 KB Jun 18 2022
Mus_musculus_NCBI_GRCm... 28149.2... Mar 3 2017
PartekFlow 128 KB Dec 16 17:53
README.txt 5 KB Jun 17 2014
SRR1553606_1.fastq 2.57 MB Jan 2 11:57
SRR1553606_1_fastqc.html 258 KB Jan 5 18:00
SRR1553606_1_fastqc.zip 292 KB Jan 2 12:09
SRR1553606_2.fastq 2.57 MB Jan 2 11:57
SRR1553606_2_fastqc.html 244 KB Jan 5 18:03
SRR1553606_2_fastqc.zip 274 KB Jan 2 12:09
SRR1553606_fastqc 4 KB Jan 4 20:32
arithmetic.R 699 B Dec 8 18:49
biostar_class 4 KB Nov 24 10:42
download_sra.sh 471 B Feb 1 2022
download_sra_parallel.sh 577 B Feb 25 2022
extract_fastq_over_represen... 968 B Sep 13 2021
fastq_statistics.sh 320 B Feb 2 2022
fastqc.sh 373 B Feb 2 2022
illumina_adapters.fa 1 KB Sep 13 2021
mm9ToMm10.over.chain 1.26 MB Feb 9 2012
multiqc_report.html 4.50 MB Sep 8 18:42
my_result.log 1 B Jun 10 2022
practical.tar 346.81 MB Oct 23 2018
snakemake-class 4 KB Oct 6 2021
Biowulf directories and files
helix.nih.gov

Figure 3

Self learning resources

Introduction to Unix on Biowulf 2024: Self learning resources

Biowulf training and learning resources

For a list of Biowulf online classes, see https://hpc.nih.gov/training/intro_biowulf/ *(https://hpc.nih.gov/training/intro_biowulf/)*. These online classes are pre-recorded and are accompanied by exercise questions. These online classes cover topics that include Biowulf basics, swarm, and submission of batch jobs.

Biowulf also offers monthly Zoom consultations *(https://hpc.nih.gov/training/#upcoming)*.

You can always refer to the Biowulf website *(https://hpc.nih.gov/systems/)* as a good reference.

Dataquest

Below are two Unix courses offered by Dataquest. Learners are able to use an browser-integrated Unix terminal to gain hands-on experience in the two Dataquest classes below. You will need a license to access Dataquest courses. Please see https://btep.ccr.cancer.gov/licenses/ *(https://btep.ccr.cancer.gov/licenses/)* for instructions on obtaining a license.

Command Line for Data Science *(https://www.dataquest.io/course/command-line-elements/)*

Intermediate Command Line for Data Science *(https://www.dataquest.io/course/command-line-intermediate/)*

Useful Unix commands for Bioinformatics

See Stephen Tuner's Bioinformatics one-liners page *(https://github.com/stephenturner/oneliners)* for commands that can help your data wrangling tasks that are often needed when conducting bioinformatics analysis.