# Section 1: Connecting to the server

## Installing the following software on your laptop

**Windows user**

**Putty**
**Filezilla client**
**VNC viewer**

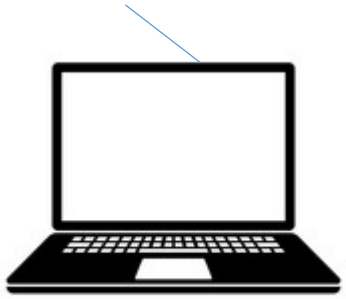
**Mac user**

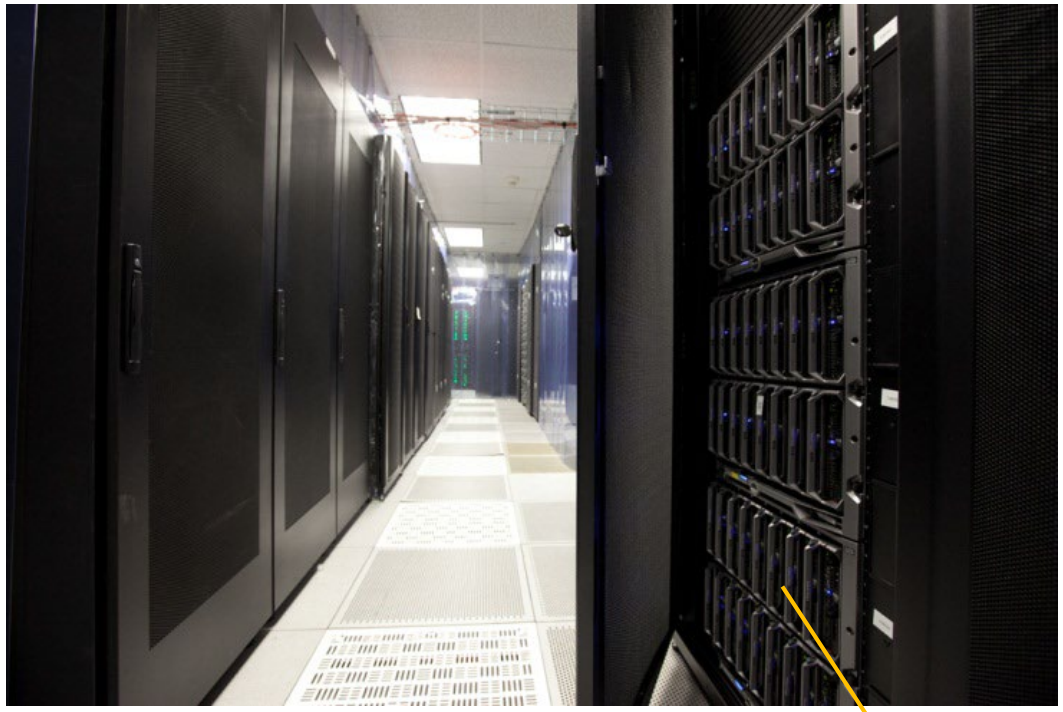
**Filezilla client**
**VNC viewer**

- Putty: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
- Filezilla client: https://filezilla-project.org/download.php?show_all=1
- VNC viwer: https://www.realvnc.com/en/connect/download/viewer/

# Running software on the BioHPC server

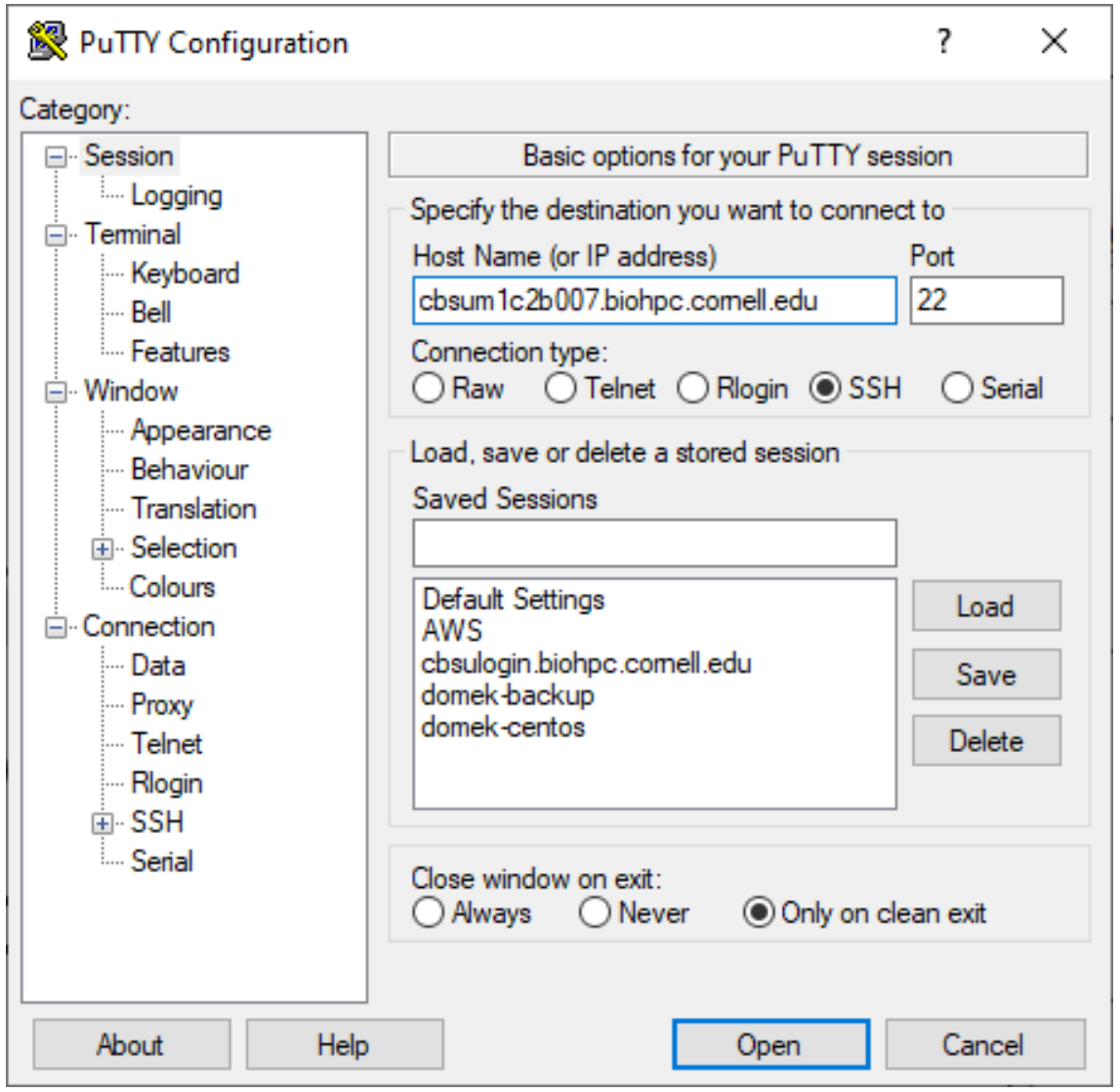
You run Putty (or other client software) on your laptop to access the server





Data analysis happens on one of the BioHPC cloud servers

# From Windows Laptop, using Putty

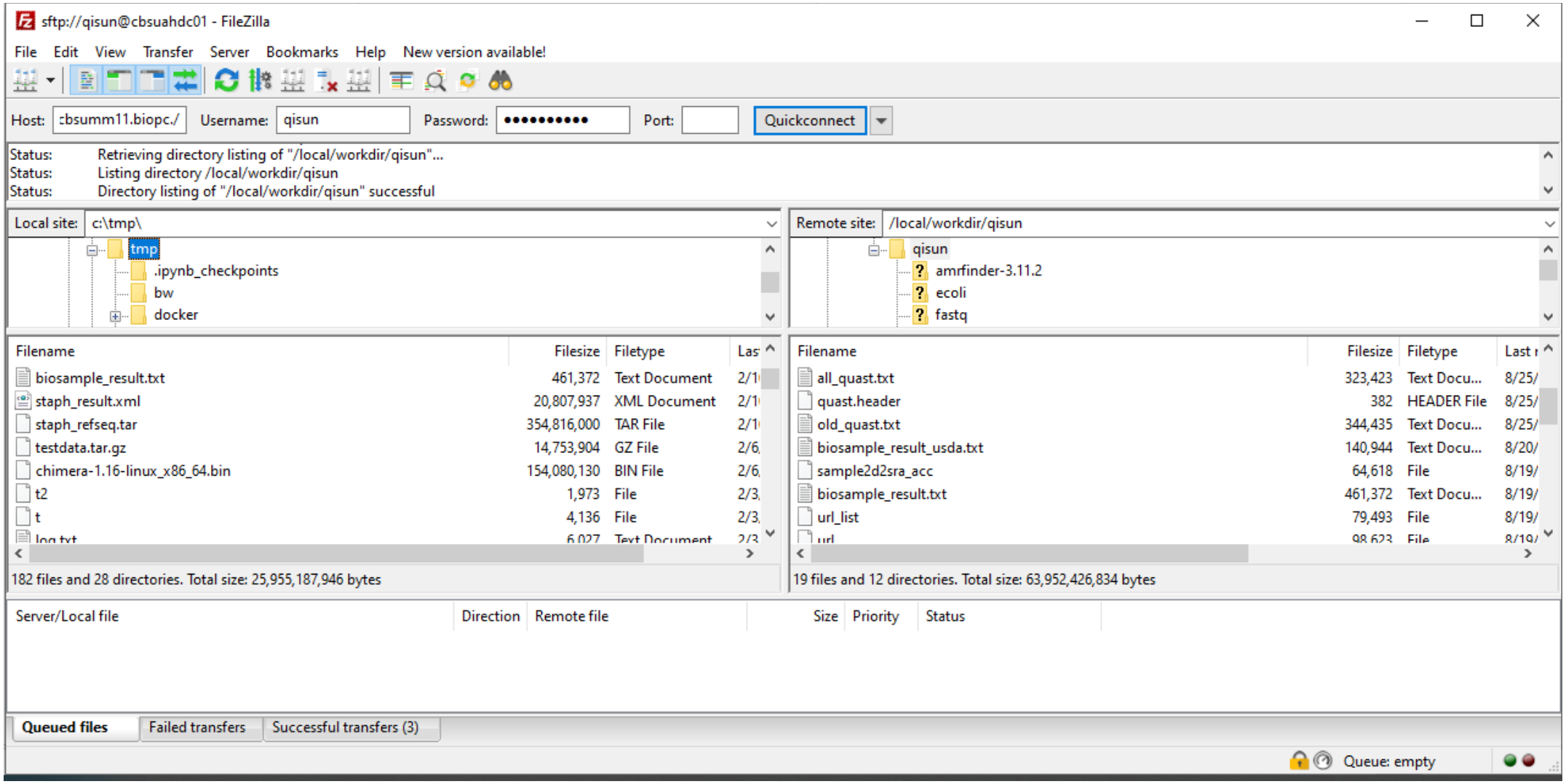


# From Mac, use Mac Terminal (no need to install)

Type the command:

```
ssh qs24@cbsumm11.biohpc.cornell.edu
```

# Upload and download files, using FileZilla



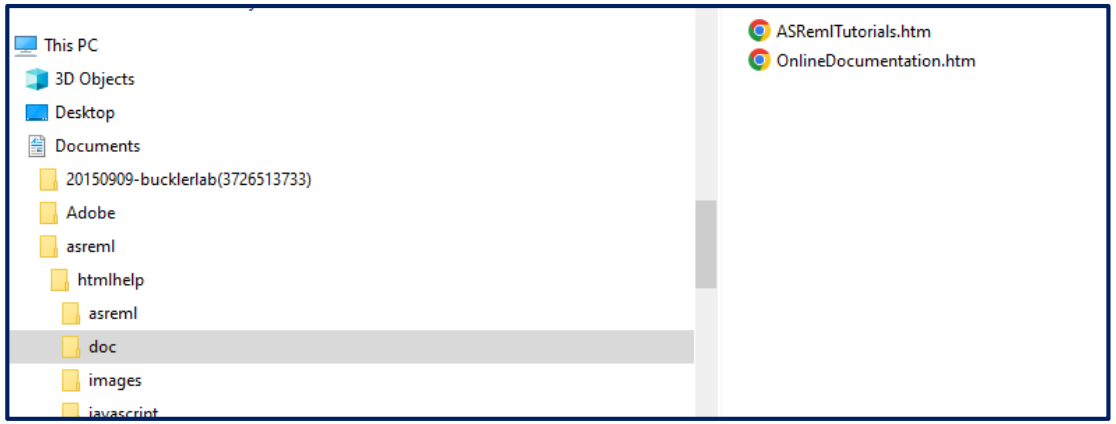
**If you work from home, set up CU VPN before you can connect to BioHPC.**

https://it.cornell.edu/cuvpn

# Section 2: Files and Directories

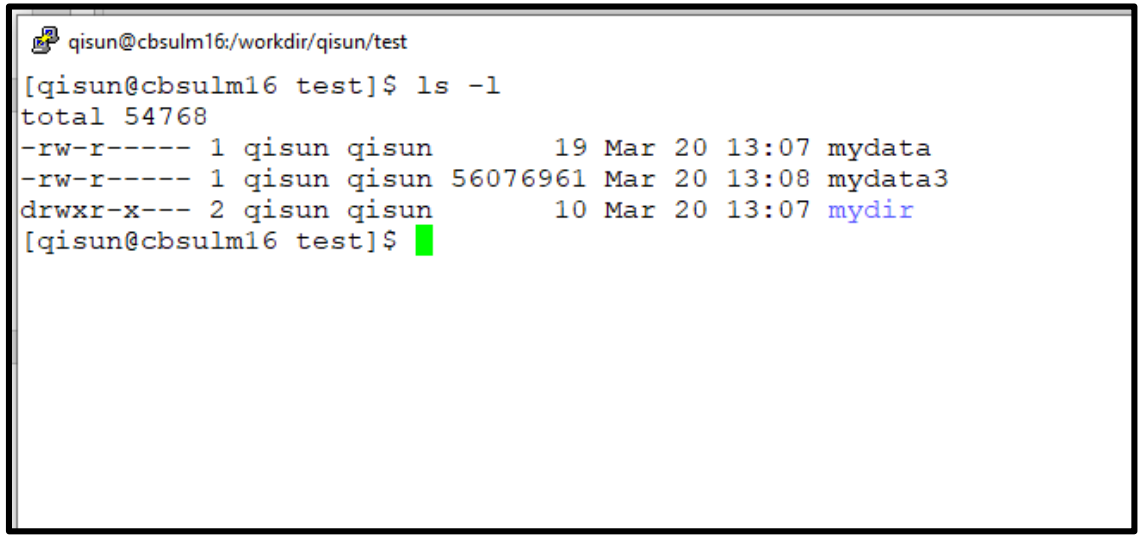
## What is **Path**?

**Path** of a file or a directory on Windows computer




C:\Users\qs24\Documents\asreml\html help\doc\ASRemlTutorials.htm

**Path** of a file or a directory on Linux computer




/workdir/qisun/test/mydata

# **Absolute Path** vs **Relative Path** of a file or a directory

Absolute path:

/workdir/qisun/mydata

Start with “/”. The first “/” refers to ROOT of the file system.

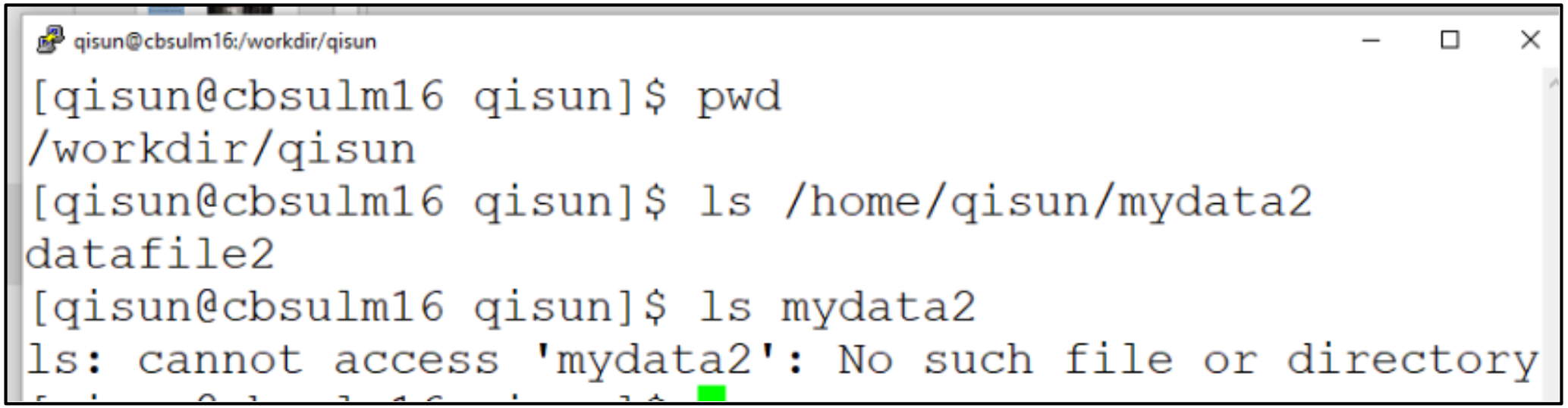
Relative path:

qisun/mydata

#relative to /workdir

# Distinguish between absolute and relative **Path?**

/home/qisun/mydata2 vs mydata2



# Section 3: Basic Linux commands

**ls**: List contents of a directory

```
ls

ls -l

ls -al

ls -l /workdir/
```

# Commands to navigate around the Linux file system

```
cd /home/qisun

pwd

cd mydata

cd ..
```

#change current directory to /home/qisun

#show current directory

#change current directory (using relative path)

#change to parent directory  (short cuts in Linux: . Current  .. Parent)

# Commonly used commands

## Create and delete directory

mkdir

rmdir

## Copy files

cp mydatafile /workdir/qisun/tmp/mydatafile2

## Delete files

rm mydatafile

rm -fr /workdir/qisun/tmp

#delete a directory including all contents

# Commonly used commands

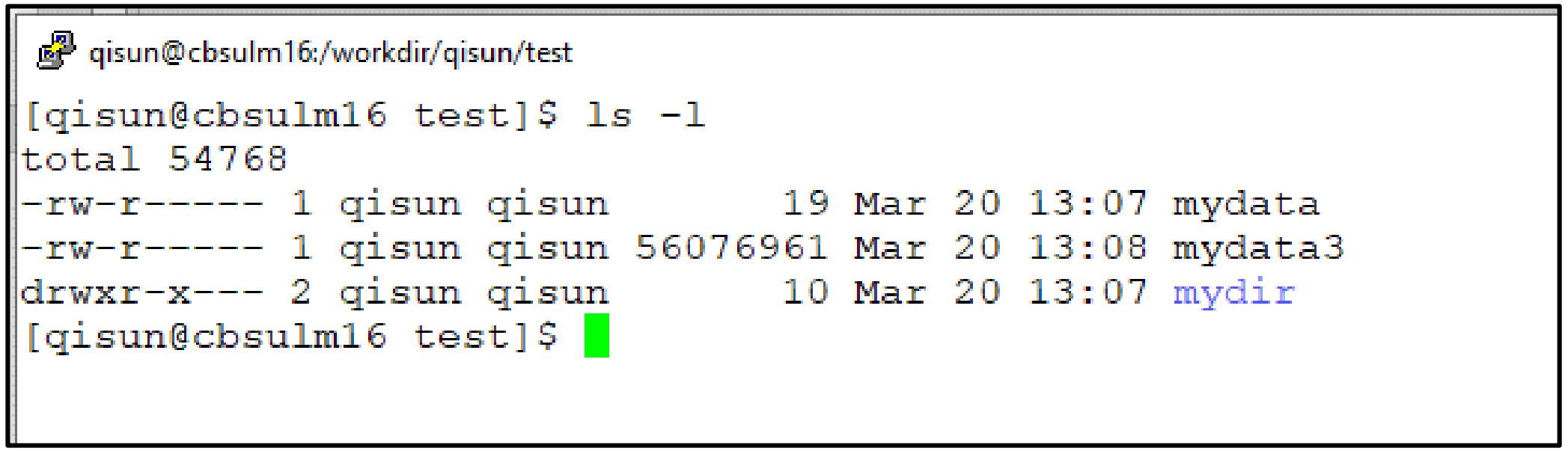
## Move a file or a directroy

mv ./mydata1  tmp/mydata1

## Rename a file or a directory

mv mydata1 mydata3

# File permissions:




rwxrwxrwx (use chmod command to change)

Owner Group Others

# **chmod**: change file permissions

```
chmod g+rw myfile

chmod a-rwx myfile

chmod -R g-w mydir

Chmod -R o+rX mydir
```

-R: recursively to the directory;

+: add

-: remove

u g a:  user, group, all

r w x: readable, writable, executable

Capital X:  executable if applicable

# Run software on Linux

- **A software is a file**

```
/programs/STAR-2.7.10b/STAR --genomeDir genome --readFilesIn
ERR458493.fastq.gz   --outFileNamePrefix wt1_
```

or

```
export PATH=/programs/STAR-2.7.10b:$PATH

STAR --genomeDir genome --readFilesIn ERR458493.fastq.gz   --
outFileNamePrefix wt1_
```

Check whether a software is still running:

```
top

top -u qs24

ps -u qs24
```

**Stop a running software:**

- Press Ctrl-C

- kill pid-number

(use ps -u qs24 command to get the PID number)

# Some short cuts

**Tab key:** auto-finish a command

**Copy paste (on Putty):** right click to copy and paste

**Copy paste (on Mac Terminal)** : standard Mac operation

# Section 4. A little more advanced

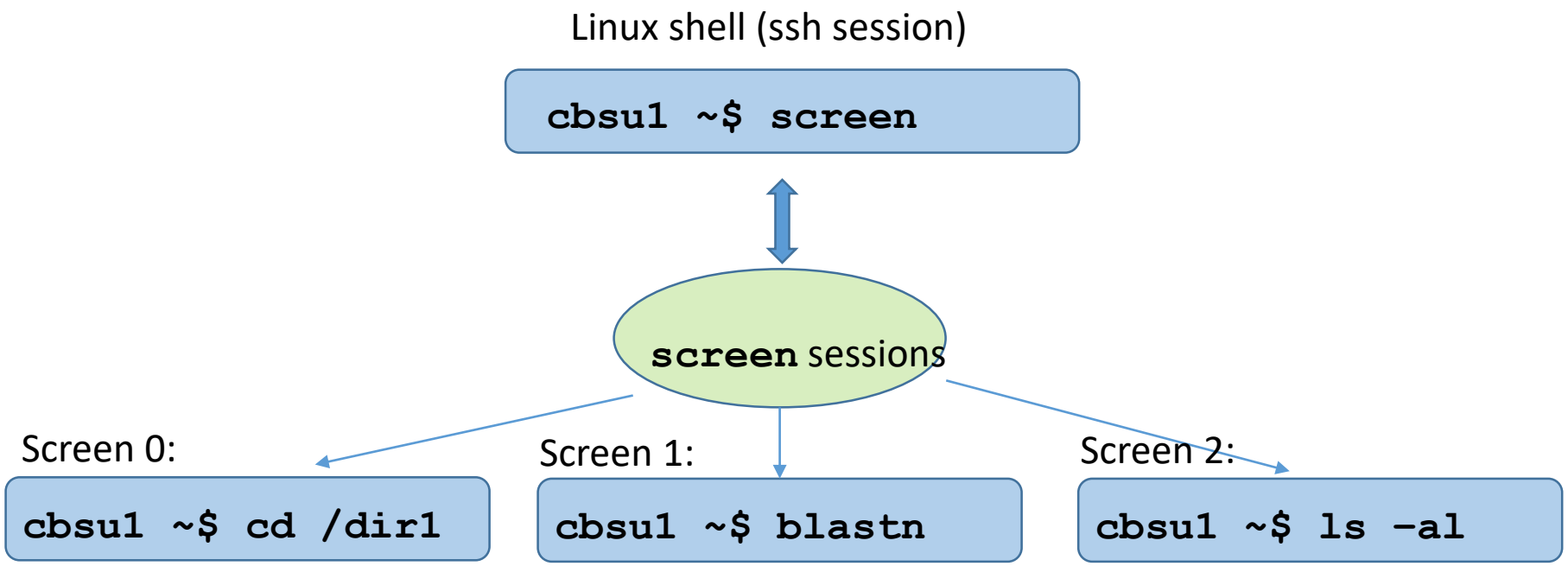
## 1. “Screen” persistent session, why do we need it?

- When you connect to a Linux server and run software, you are in a connection “session”, and all software running processes are associated with this “session”;
- If you close the laptop, or your internet connection is interrupted, the “session” is closed. All the jobs running associated with the session would be terminiated.
- To avoid this problem, you can create “screen” persistent sessions, and running software in these persistent sessions. These sessions are not killed even if you close the laptop.

**A tutorial of “screen”**

https://biohpc.cornell.edu/lab/doc/Linux_exercise_part2.pdf

# Using `screen`




Ctrl-a + d: detach from “screen”;

Ctrl-a + c:  create new session;

Ctrl-a + n:  switch between sessions

# screen: cheapsheet

After logging in, type **`screen`**

Most useful **`screen`** commands:

| Screen command | What it does |
|---|---|
| **screen** | **Start a new session** |
| **screen -list** | **List all your screen sessions** |
| **screen -d -r**<br>**screen -d -r [sessionID]** | **Re-attach previously detached (or unintentionally disconnected) session – can be done upon next login** |
| **Ctrl-a c** | **Create a new window (shell) in a session; can be repeated multiple times** |
| **Ctrl-a n  Ctrl-a p** | **Switch to next (n), previous (p) window within a session** |
| **Ctrl-a “** | **List all windows in a session, switch to one** |
| **Ctrl-a d** | **Detach a session (all windows will continue running)** |
| **Ctrl-d** | **Exit form current window (or from whole session, if in last window)** |
| **screen -X -S [name] quit** | **Kill session “name” (obtained from screen -list)** |

For more features/functionality – type screen –h or Ctrl-a ? (within session)

**Sessions are persistent – will survive connection problems, turning off laptop, etc.**

# 2. Run commands in batch:

## Create a text file, give it a name, e.g. “myscript.sh”

```
STAR --genomeDir genome --readFilesIn ERR458491.fastq.gz   --outFileNamePrefix wt1_
STAR --genomeDir genome --readFilesIn ERR458492.fastq.gz   --outFileNamePrefix wt2_
STAR --genomeDir genome --readFilesIn ERR458493.fastq.gz   --outFileNamePrefix wt3_
STAR --genomeDir genome --readFilesIn ERR458494.fastq.gz   --outFileNamePrefix wt4_
```

- use Notepad++ on Windows, BBEdit on Mac;
- Advanced users can use vi or nano;
- More advanced users can use xargs

## Run batch script

#run command one by one

```
sh myscript.sh
```

#run command in parallel

```
parallel -j 4 < myscript.sh
```

# process 4 commands at a time

## 3. Check file contents

```
head -n 20 myfile.txt
```
#show first 20 lines

```
tail -n 20 myfile.txt
```
#show last 20 lines

```
more myfile.txt
```
#show file content page-by-page, press “space bar” to continue, “q” to quit

```
wc -l myfile.txt
```
#”wc -l” to count number of lines in a file

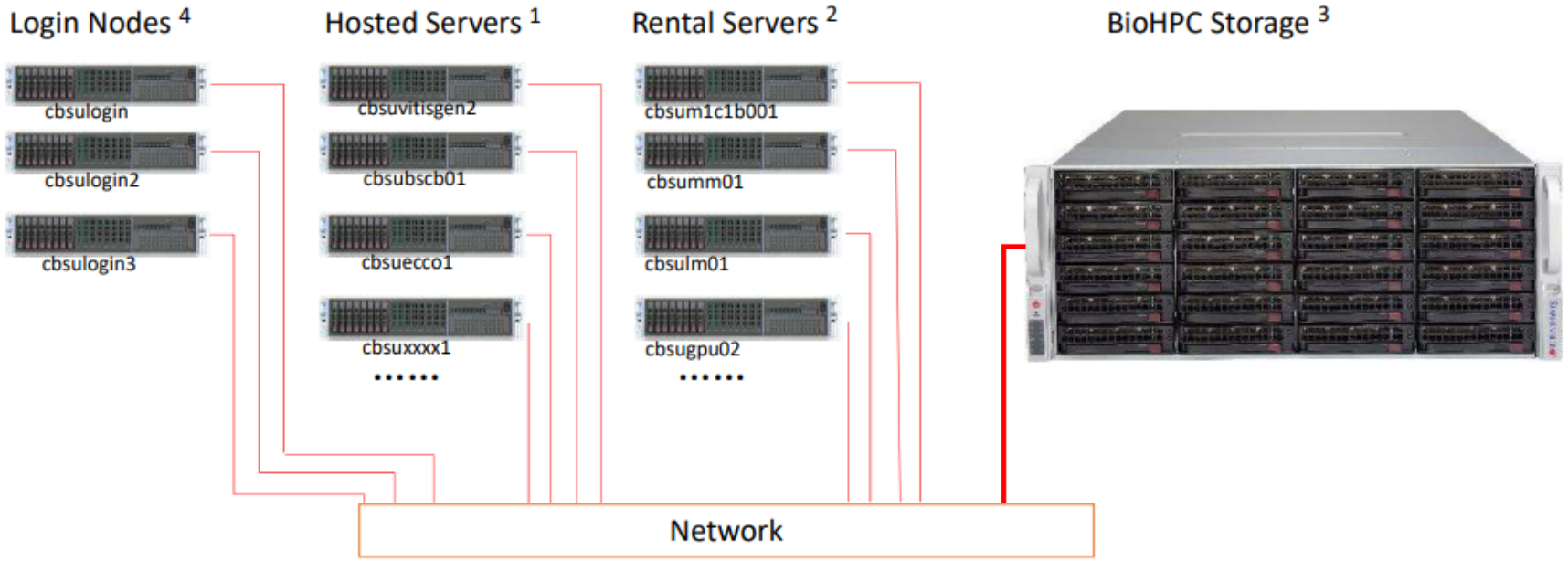
```
zcat myfile.gz | more
```
#show compressed by, pipe into more

```
zcat myfile.gz | grep ACGGGAT
```
#filter to lines containing “ACGGGAT”

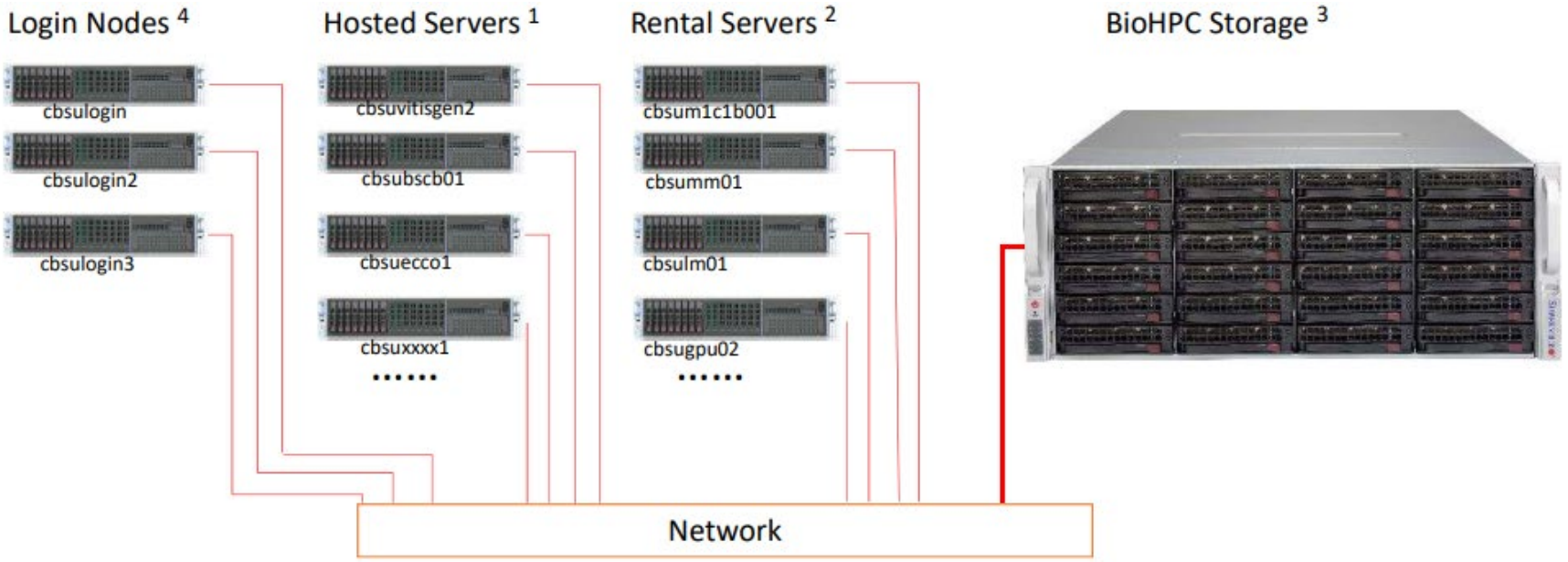
# Best Practice of using BioHPC

## The BioHPC Cloud has 4 types of computers

1. **Hosted Servers:** Computers owned by individual labs but managed by the BioHPC team;
2. **Rental Servers**: Computers available for hourly rent by BioHPC users;
3. **BioHPC Network Storage**: Central storage accessible from each BioHPC server;
4. **Login Nodes**: Computers used for accessing BioHPC data storage without reservation, and for BSCB users to access their cluster. The login nodes can also be used for accessing compute servers from outside campus without VPN. No computing is allowed on these machines.



# Best Practice of using BioHPC




**/home/$USER:** home directory, network drive, do not compute on large files in the directory

**/home2/$USER:** network drive, ok to compute large files in the directory

**/workdir/:** scratch disk on local computing node, you can compute large files in the directory

# Using BioHPC

## 1.Upload data files to BioHPC

Filezilla to login nodes (cbsulogin, cbsulogin2, cbsulogin3)

## 2. Reserve a BioHPC server

- Make reservation through https://biohpc.cornell.edu
- You can share the reservation with other lab users

## 3. Run software on the computing nodes

- Make sure to copy large data files to /workdir, and compute on files in /workdir
- After done, copy data files back to /home/$USER, and cancel reservation

**For support**

- Email support@biohpc.cornell.edu

- Book an office hour https://biohpc.cornell.edu/lab/office1.aspx