

BioHPC workshop Linux for Biologists: Exercises

Part 1

Exercise 0: Log in to a Linux workstation using an ssh client

If you are on Ithaca campus or have the Ithaca NetID

If you have the Cornell-Ithaca NetID but are currently not on campus, launch the VPN connection on your local machine (laptop) using the CIT-provided **Cisco AnyConnect Secure Mobility Client**. This will make your laptop effectively a part of Ithaca campus network.

If you have a Windows laptop

If not yet done, download the PuTTY ssh client: <https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe>. Save the exe file anywhere on your laptop (e.g., on the Desktop for access).

Double-click on the PuTTY icon. In the '**Host Name**' field, enter the full name of your assigned machine (e.g., `cbsum1c1b002.biohpc.cornell.edu`). Make sure that '**Port**' is set to '**22**' and '**Connection type**' to '**ssh**'. Click '**Open**'. A terminal window will open with the login prompt. At the prompt, type your BioHPC user ID and hit ENTER. Then enter your BioHPC password and hit ENTER (NOTE: as you type the password - nothing will be happening on the screen - this is on purpose).

Since you will be accessing your assigned machine often during the workshop, it makes sense to create and save a customized profile for it in PuTTY. To do this, open the PuTTY client and enter the full name of the workstation in the '**Host Name**' field and make sure the '**Connection Type**' is '**ssh**' and '**Port**' is '**22**'. Then under 'Saved Session', enter a short nickname for the machine (e.g., the first part of the name, like `cbsum1c1b002`). Expand the '**SSH**' tab in the left panel and click '**X11**' in the left panel, check the box '**Enable X11 forwarding**'. If you prefer the black text on white background, you can change the color settings. Click '**Colours**' in the left panel, set '**Default Foreground**' and '**Default Bold Foreground**' to '**0 0 0**', '**Default Background**' and '**Default Bold Background**' to '**255 255 255**'. Once the customization is complete, click '**Session**' in the left panel, and then click '**Save**'. This will save the machine's profile under a nickname you specified, and it will appear on the list of saved profiles. To connect to a machine with the saved profile, just double-click on the nickname displayed in the '**Saved Sessions**' section.

If you have a Mac (or Linux) laptop

Launch the terminal window. Type (replacing `cbsum1c1b002` with the name of your assigned machine and `your_id` with your own BioHPC user ID)

```
ssh -Y your_id@cbsum1c1b002.biohpc.cornell.edu
```

Enter user your BioHPC password when prompted.

If you are outside of Ithaca campus and do not have the Cornell-Ithaca NetID

First, you will need to ssh to one of our [login nodes](#), and from there - `ssh` further to your assigned machine. To do this, follow the instructions above for your type of laptop, replacing the name of your assigned machine with either of the [login nodes](#): `cbsulogin`, `cbsulogin2`, or `cbsulogin3` (all with the `.biohpc.cornell.edu` suffix). In the terminal which opens on the login

node (you will notice the name of that node at the prompt), `ssh` further to you assigned machine, e.g.,:

```
ssh cbsum1c1b002
```

Notice that the part `your_id@` and the domain `.biohpc.cornell.edu` have been omitted from the `ssh` command above. This is possible because your user ID on the login node is the same as on the assigned machine, and all BioHPC machines share the same domain.

Exercise 1: conversation with Linux – simple command examples

Now that you are logged in to your machine, let's try running some simple commands.

Find the name and other information about the machine you are logged in to

```
uname -a
```

Check who else is logged in to this machine?

```
who
```

What is your current working directory?

```
pwd
```

List the contents of the directory

```
ls -al
```

How much disk space does my directory take? How about breakdown into subdirectories? Save the output to a file on disk.

```
du -hs
du -h --max-depth=1 >& disk_occupancy.log
```

Display the created file on the screen

```
cat disk_occupancy.log
```

Look into the file using the `less` paginator (hit `q` to exit when done looking)

```
less disk_occupancy.log
```

Open the file in the nano text editor. Try to change the content of the file and save the changes:

```
nano disk_occupancy.log
```

Find summary information about the storage available on the machine

```
df -h
```

Find summary information about RAM memory available on the machine (the most important fields are `total` - all the machine has, and `available` - this is what is left for you to use)

```
free
```

Find more information about the `du` command (when done reading - press `q`)

```
man du
```

Find and display on the screen the recent commands containing the string `occupancy`. Note the use of the **pipe** construct: the vertical bar `|` means that the output of the command on the left-hand side (here: `history`) is passed on as input to the command on the right-hand side (here: `grep`)

```
history | grep occupancy
```

Using the mouse, copy one of these commands to the clipboard, then paste it into the command line and hit ENTER to execute again.

Exercise 2: basic operations on directories

1. Create your temporary directory in the scratch file system `/workdir` (in the commands below, replace `your_id` with your own BioHPC user ID) and verify this new directory exists

```
cd /workdir
mkdir your_id
ls -al
```

or

```
mkdir /workdir/your_id
ls -al /workdir
```

2. Now create a subdirectory (of that new directory), called `mytmp` and verify it has indeed been created

```
cd /workdir/your_id
mkdir mytmp
ls -la
```

3. List contents of `mytmp`
`ls -al mytmp` (if already in `/workdir/your_id`)
4. Delete `mytmp` (and verify it is no longer there)

```
rmdir mytmp
ls -al
```

Exercise 3: basic operations on files

If not yet present, create directory `/workdir/your_id` (replace `your_id` by your real user ID)

```
mkdir /workdir/your_id
```

Copy the file `examples.tgz` located in `/shared_data/Linux_workshop` to your temporary directory

```
cd /workdir/your_id
cp /shared_data/Linux_workshop/examples.tgz .
```

With `/workdir/your_id` still as your current directory, unpack the file `examples.tgz` and list the resulting files and directories, paying attention to file sizes:

```
tar -xzf examples.tgz
ls -al
```

Check the type of a few files (and directories)

```
file flygenome.fa
file shorty_reads.fastq
file examples.tgz
file scripts
```

Compress the file `flygenome.fa` using `gzip`, then check the size of the resulting file `flygenome.fa.gz`. How much disk space was saved by compressing the file?

```
gzip flygenome.fa
ls -al *.gz
```

Un-compress the file back to its original form, verify that the file has been recovered

```
gunzip flygenome.fa.gz (or gzip -d flygenome.fa.gz )
ls -al flygenome*
```

Create a new directory in `/workdir/your_id`, called `sequences`

```
cd /workdir/your_id
mkdir sequences
```

Move the files `flygenome.fa` and `short_reads.fastq` to directory `sequences`

```
mv flygenome.fa short_reads.fastq sequences
```

(note: the last argument of `mv` is the target directory). Alternative method: move each file separately

```
mv flygenome.fa sequences
mv short_reads.fastq sequences
```

Create a new directory in `/workdir/your_id`, called `shellscripts`

```
mkdir shellscripts
```

Move all shell scripts (*i.e.*, all files with names ending with `.sh`) from directory `scripts` to the newly created directory `shellscripts`

```
mv scripts/*.sh shellscripts
```

Remove the directory `scripts`

`rmdir scripts` (What is the error and why?)

To remove a non-empty directory, we need to use `rm` instead:

```
rm -rf scripts
```

Create a `tgz` archive of the directory `shellscripts`, (call it `my_shellscripts.tgz`), verify it was created

```
tar -czvf my_shellscripts.tgz ./shellscripts
ls -al *.tgz
```

Exercise 4: basic operations on text files

Open the file `/workdir/userID/ZmB73_5b_FGS.gff` in text editor `nano` and/or `vim`, navigate through the file, edit it, save. Repeat with file `/workdir/your_id/shellscripts/bwascript2.sh`

Page through a file using `less`

```
cd /workdir/your_id
less ZmB73_5b_FGS.gff
```

Display the first 10 and the last 10 lines of the `fastq` file

```
cd /workdir/your_id/sequences
head -10 short_reads.fastq
tail -10 short_reads.fastq
```

Save lines 1000 through 2000 of the `fastq` file above into another file

```
head -2000 short_reads.fastq | tail -1000 > middle_lines.fastq
```

Count the lines/words/characters in a `fastq` file. How many reads does this file contain?

```
wc short_reads.fastq
```

Look for a string in a file and number of lines the string occurs in

```
grep AATTCGT short_reads.fastq
grep AATTCGT short_reads.fastq | wc -l
```

Exercise 5: using `screen` to create a persistent session

If not already done so, connect to your assigned workstation via `ssh` (using PuTTY or other `ssh` client)

In the terminal window, type `screen` and hit ENTER. You just opened the first window in your `screen` session.

Type `Ctrl-a c` (i.e, press `Ctrl` key and while holding it press `a`, then let go of both keys and press `c`). Then do it one more time. You just opened two more `screen` windows within your session. Each of these is a separate Linux shell awaiting your commands.

Now let's do something different in each of the windows (shells) you just created within your `screen` session. Execute the `ls -al` command in the current window. Then switch to the next window pressing `Ctrl-a n` and run the `pwd` command there. Switch to the next window hitting `Ctrl-a n` again. Switch to previous window using `Ctrl-a p`. As you cycle through the windows this way, you will see them as you last left them.

Simulate a network or power problem by closing the PuTTY terminal window (it "X" in the upper right corner). This will close your terminal window and disconnect you from the machine. However, the `screen` session you created before with all windows you opened in it will continue running so that you can re-connect to it later.

To do this, used PuTTY to log in to your assigned machine again. In the terminal window, type `screen -list`. You should see the `screen` session you left behind (in this case, it will be just one such session)

Type `screen -d -r`. This will re-connect you to your `screen` session. Cycle through the windows using `Ctrl-a p`, `Ctrl-a n`, or `Ctrl-a "` (this last command will list all your windows and allow you to select one of them). Do you see your windows as you left them?

Gracefully detach your `screen` session from the terminal using `Ctrl-a d` (you won't see your windows any more, but they will keep running 'behind the scenes'). Then re-attach again using `screen -d -r`.

Terminate your `screen` session by hitting `Ctrl-d` in each window (this will terminate the current window). Doing it in the last window will terminate the whole `screen` session (a relevant message will be displayed). Your main PuTTY terminal will keep running (until you close it). After a `screen` session is terminated, you cannot re-connect to it (since there is nothing to re-connect to any more). You can open a fresh `screen` session if you wish.

Exercise 6: connect to your assigned workstation using VNC

Go to "My Reservations" page <http://biohpc.cornell.edu/lab/lab.aspx>, log in, click on "My Reservations" menu link.

Choose resolution from the resolution dropdown (depends on your monitor).

In the table listing your reserved machines, find the column "VNC port" in the row corresponding to the machine you want to connect to. If the value in this column is empty, click on "Connect VNC". This will start the VNC server program on the Linux machine which will be waiting for your connection attempt. If the value of the **VNC port** is not empty, it means that the VNC server was already been started on your behalf in the past and it may be running. In such a case, use your VNC viewer to attempt a connection. If it does not work (even though you are sure you have established a VPN connection or port tunneling - see below), the VNC server may have been terminated or may be hung up, in which case clicking on the "Reset VNC" link will restart it (killing the old, hung-up instance).

Make sure you are not restricted by Cornell firewall

If you have not already done so, launch the VPN connection to Cornell network, then proceed to sub-section 'Connect via VNC'.

If you cannot use Cornell VPN, you will need to **tunnel** the VNC port assigned to you through ssh to one of the login nodes:

On a Mac: launch the **terminal** application and enter (substituting your user ID, workshop machine name, and the assigned VNC port)

```
ssh -N -L 5901:cbsum1c2b007:5901 your_id@cbsulogin.biohpc.cornell.edu
```

(instead of `cbsulogin` you can also use `cbsulogin2` or `cbsulogin3`). Provide your BioHPC password when prompted. Keep this ssh connection running (you can minimize the window).

On Windows: launch the PuTTY ssh client. In the **Host Name** textbox, enter `cbsulogin.biohpc.cornell.edu` (`cbsulogin2` and `cbsulogin3` may also be used instead). In the left panel click **SSH** and then **Tunnels**. Enter your assigned VNC port (e.g., `5901`) as **Source port**. As **Destination**, enter the name of your workshop machine name followed by the colon ':' and the VNC port (e.g., `cbsum1c2b007:5901`). Click the **Add** button. Click **Open** and log in using your BioHPC user ID and password. Keep this ssh session open (you can minimize the terminal window).

Connect via VNC

Open your VNC viewer and enter the name of the machine followed after the colon ":" by the port number shown by the website (or - if none of the links was clicked - taken from the **VNC port** column), for example, `cbsum1c2b007.biohpc.cornell.edu:5901` (if you are using port tunneling rather than VPN, the machine name to enter will be `localhost`). When prompted, enter your BioHPC password in the VNC viewer. When the splash screen appears, you may need to position your mouse pointer on it and hit ENTER to access the Linux desktop login screen. Enter your BioHPC password again on that screen to log into the machine.

Open terminal window in the VNC desktop by right-click on the desktop background and choosing "Open Terminal". You may also open other applications. For example, to open a web browser, type `firefox` in the terminal window.

To disconnect from the machine but keep your VNC session running, close the VNC viewer using the "X" in the upper-right corner of the viewer's window. You can re-connect by opening the VNC viewer again and entering the machine name and port. You will find your session alive and well, the way you left it, with all opened applications still running.

To permanently close your VNC session, click on the "power button" icon in the upper-right corner of the Linux desktop (not of the VNC viewer!). You will find "Logout" as one of the options. Use it to close your session (killing all application within it), which will also terminate the VNC server on the machine. You may see the empty VNC viewer window trying to re-connect to the non-longer running VNC server and displaying an error message - just ignore it and close the VNC viewer. Once the VNC session is closed this way, it no longer exists and so you cannot re-connect to it. You can start a fresh VNC session by visiting the "My Reservations" page and using the "**Reset VNC**" link to start the VNC server.