Linux for Biologists

Exercise Part 1

Robert Bukowski
Institute of Biotechnology
Bioinformatics Facility
(aka Computational Biology Service Unit - **CBSU**)

Slides: http://biohpc.cornell.edu/lab/doc/Linux_workshop.pdf

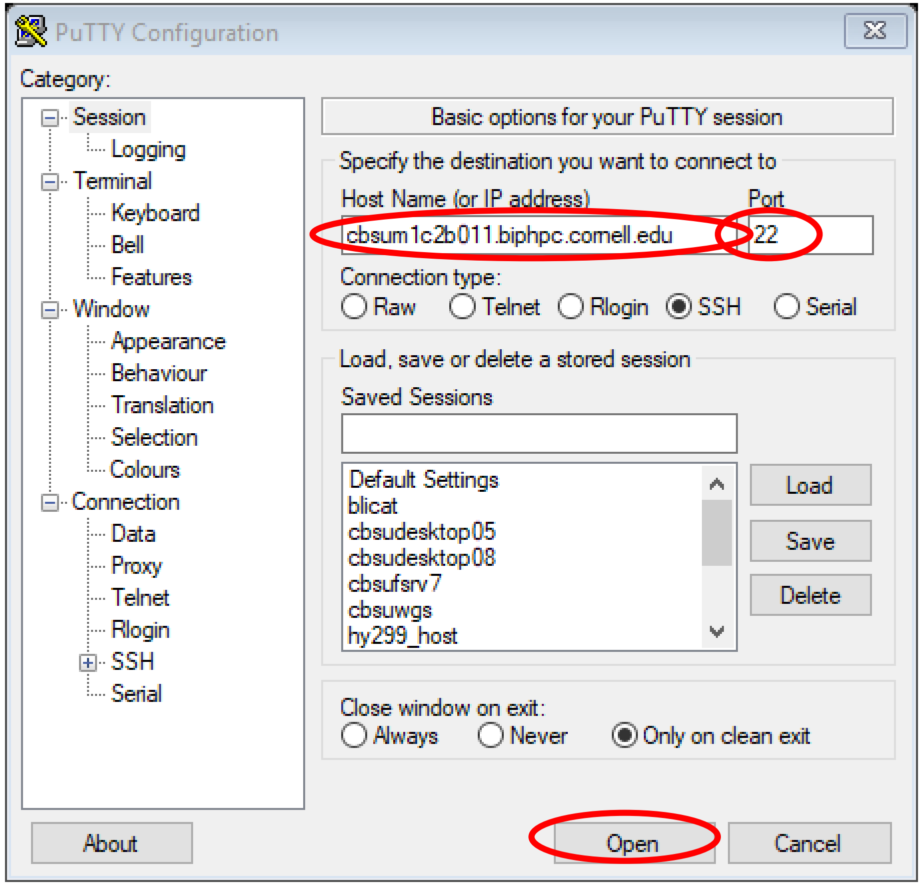
Exercise: http://biohpc.cornell.edu/lab/doc/Linux_exercise_part1.pdf

Contact: brc_bioinformatics@cornell.edu

Exercise 0: Log in to your workshop machine via ssh

Machine allocations: https://biohpc.cornell.edu/ww/machines.aspx?i=115

Windows: double-click on PuTTy icon
Provide machine name and click **Open**
Provide user name and password (when asked)



PuTTY Configuration

Category:

- Session
  - Logging
- Terminal
  - Keyboard
  - Bell
  - Features
- Window
  - Appearance
  - Behaviour
  - Translation
  - Selection
  - Colours
- Connection
  - Data
  - Proxy
  - Telnet
  - Rlogin
  - SSH
  - Serial

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): cbsum1c2b011.biphpc.cornell.edu
Port: 22

Connection type:
○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings
blicat
cbsudesktop05
cbsudesktop08
cbsufsrv7
cbsuwgs
hy299_host

Load
Save
Delete

Close window on exit:
○ Always ○ Never ● Only on clean exit

About
Open
Cancel

Mac, Linux:

Open the **Terminal app** (in Utilities/Applications)

In the **Terminal**, type

```
ssh –Y myID@cbsum1c2b011.biohpc.cornell.edu
```

(replace **`myID`** and **`cbsum1c2b011`** with your user ID and allocated machine name)

Exercise 1: conversation with Linux – simple command examples

Find the name of the machine you are logged in to (**`uname –a`**)

Who else is logged in to this machine? (hint: use **`who`** or **`w`**)

What is the directory you “are” currently in? (hint: use **`pwd`**)

List the contents of the directory (use **`ls -al`**)

How much disk space does my directory take? (**`du –hs .`** or **`du –h –max-depth=1`**)

Find summary information about the storage available on the machine (**`df –h`**)

Find summary information about RAM memory available on the machine (**`free`**)

Fid more information about the du command (**`man du`**)

Repeat one of the previous commands without re-typing it (use mouse-copy and paste, **`history`** command)

Exercise 2: basic operations on directories

1. Create your temporary directory in the scratch file system **`/workdir`**
2. create a subdirectory (of that new directory), called **`mytmp`**.
3. Verify the subdirectory **`mytmp`** has been created
4. list contents of **`mytmp`**
5. remove **`mytmp`**

Exercise 2: solution

```
cd /workdir

pwd

mkdir my_id                (replace my_id  with your own userID)

ls -al

mkdir my_id/mytmp

ls  -al

ls -al mytmp

rmdir mytmp
```

Exercise 3: basic operations on files

1. If not yet present, create directory `/workdir/your_id` (replace `your_id` by your real userID).
2. Copy the file `examples.tgz` located in `/shared_data/Linux_workshop` to your temporary directory
3. Unpack the file `examples.tgz` and list the resulting files and directories
4. Check the type of each file (hint use the `file` command)
5. Create a new directory in `/workdir/your_id`, called `sequences`
6. Move the files `flygenome.fa` and `short_reads.fastq` to directory sequences
7. Create a new directory in `/workdir/your_id`, called `shellscripts`
8. Move all shell scripts (i.e., all files with names ending with "`.sh`") from directory `scripts` to the newly created directory `shellscripts`
9. Remove the directory `scripts`

Exercise 3: solution

```
cd /workdir

mkdir bukowski

cd bukowski

cp /shared_data/Linux_workshop/examples.tgz .

tar –xzvf examples.tgz

ls –al

ls –al scripts

file * scripts/*

mkdir sequences

mv flygenome.fa short_reads.fastq sequences

mkdir shellscripts

mv scripts/*.sh shellscripts

ls –al shellscripts

rm –Rf scripts
```

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/userID/shellscripts/bwascript2.sh**

Page through a file using **less**

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

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

```
    cd /workdir/userID/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
```

Note the size of the file (use **ls –al**). The compress the file using **gzip**. What is the gain from compression?

```
ls -al short_reads.fastq
gzip short_reads.fastq
ls -al short_reads.fastq.gz
```

Exercise 5: advanced processing of text files

Among the files used in Exercise 2, there is a file `ZmB73_5b_FGS.gff`, describing gene annotations in maize. The file is TAB-delimited (check this!) with following columns:

1. Chromosome
2. Source
3. Feature
4. Start position
5. End position
6. Score
7. Strand
8. Frame
9. Attribute

Tasks:

Look into the file to examine its structure (use **`more,cat`** or a text editor)

Create a new file, containing only **gene** features, with columns 9, 1, 4, and 5 (in this order)

Sort this new file over **Chromosome** and **End position**

Examine the sorted file in a text editor

Exercise 5: solution

Extract the genic lines to a temporary file

```
grep -P "\tgene\t"  ZmB73_5b_FGS.gff > tmp_gene
```

Extract the last column to another temporary file

```
cut –f 9 tmp_gene > tmp_gene_attr
```

Get columns 1,4,5 and paste them to the right or column 9

```
cut -f 1,4,5 tmp_gene | paste tmp_gene_attr - > final_file
```

Sort the file obtained above

```
sort –k 2,2 –k 4,4n final_file > final_file_sorted
```

Remove the temporary files

```
rm  tmp_gene  tmp_gene_attr  final_file
```

Examine the final sorted file in a text editor

```
vi final_file_sorted
nano final_file_sorted
```

…