
Unix introduction

Denis Seyres < *ds777@medschl.cam.ac.uk* > Sandra Cortijo
< *sandra.cortijo@slcu.cam.ac.uk* >

ADVANCED CHIP-SEQ DATA ANALYSIS
FEBRUARY, 2018

General information

The following standard icons are used in the hands-on exercises to help you locating:



Important Information



General information / notes



Follow the following steps



Questions to be answered



Warning – PLEASE take care and read carefully



Optional Bonus exercise



Optional Bonus exercise for a champion

Starting with unix? Find here a quick overview of the most used commands

The Directory Structure



All the files are grouped together in the directory structure. The file-system is arranged in a hierarchical structure, like an inverted tree. The top of the hierarchy is traditionally called root (written as a slash /).

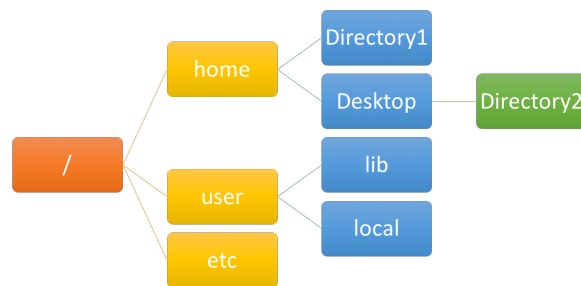


Figure 1: Typical unix tree

Open the Terminal.

You can easily see how is organized your computer,

```
tree
```

and see the full path of your current directory.

```
pwd
```

When you first login, your current working directory is your **home directory**.

Your **home directory** has usually the same name as your user-name and it is where your personal files and subdirectories are saved.



To find out what is in your **home directory**, type

```
ls
```

The **ls** command lists the contents of your current working directory. Use **-l** option to display files and folders permissions, sizes or last changes.

```
ls -l
```

You can add **h** option which means *human readable* to have more meaningful infos.

```
ls -lh
```

The `*` is the wildcard. It replaces any character/digit of any length. Imagine you want to display only pdf files in your current working directory.

```
ls *.pdf
```

We will now make a subdirectory in your home directory. To make a subdirectory called *Directory1* in your current working directory type.



```
mkdir Directory1
```

You can double-check if a new directory has been created,

```
ls
```

and its content (it is still empty).

```
ls Directory1
```

Now, let's walk into the tree. We want to change directory and go to *Directory1*. The command is

```
cd Directory1
```

In fact, directories are never really empty. If you add the option `-a` you will see all files, including the hidden files (starting with a dot).

```
ls -a
```

As you can see, in the current directory (and in all other directories), there are two special directories called `(.)` and `(..)`. In UNIX, `(.)` means the current directory, so typing

```
cd .
```

means 'stay where you are'. `(..)` means the parent of the current directory, so typing

```
cd ..
```

will take you one directory up the hierarchy (back to your home directory).

File operations

Copying a file is achieved by the command 'cp' followed by the file to copy and the name of the new file.



```
cp ./unix_commands/file1.txt ./Directory1/file1.txt
cp ./unix_commands/file1.txt ./Directory1/file2.txt
```

You can remove a file with the command **rm**. Use this command sparingly because its action is irreversible.

```
rm ./Directory1/file1.txt
```



You are working with NGS data. Files are generally too big to be opened and you just want to see what is inside your files. One option is to print the first lines of the file.

```
head ./unix_commands/file1.txt
```

head command will display the first 10 lines by default.

You can choose to print more lines:

```
head -n 40 ./unix_commands/file1.txt
```

Now you want to see how your file ends. Use

```
tail ./unix_commands/file1.txt
```

Your file is not so big and you decide to open it entirely. The command **less** writes the contents of a file onto the screen a page at a time. Use [space] key to change pages. Press 'q' key to quit.



```
less ./unix_commands/file1.txt
```

When you launch a job, you know it will take several minutes or hours and you want to stop it because you have just identified a mistake in your code. You can stop it by associating the two keys, 'ctrl' + 'c' of your keyboard. Start the script runningScript.sh. This bit of code will run for several minutes. Try to stop it.

```
cd ../unix_commands/
bash runningScript.sh
```

grep is one of many standard UNIX utilities. It searches files for specified words or patterns. You must type **grep** command followed by the word/pattern of interest and at the end the file to parse.

```
grep "0.8951747" ./unix_commands/file1.txt
```

Each line matching the expression *0.8951747* will be displayed.

Another VERY useful command is **wc** for *word count*. It will return either the number of words or of lines of your file.

Number of words:

```
wc -w ./unix_commands/file1.txt
```

Number of lines:

```
wc -l ./unix_commands/file1.txt
```



Redirecting the output

You may need to save the resulting output of a command. You can simply use `>` to redirect that output to an other file.

```
head ./unix_commands/file1.txt > ./Directory1/file1_head.txt  
less ./Directory1/file1_head.txt
```

A good practice is to limit the number of intermediate files. It will save you a lot of time and space on the disk.

For this, `|` is used to **pipe** different commands, and is very useful to save time (intermediate files are not written on disk) and space. Let's print the lines number 6 and 7.

```
head -n 7 ./unix_commands/file1.txt | tail -n 2 >  
./Directory1/file1_lines6-7.txt
```

Concatenate two files, or more using **cat** command.

```
cd ./Directory1  
cat file1_lines6-7.txt file1_head.txt > file3.txt  
cat file1_lines6-7.txt file1_head.txt file3.txt > file4.txt  
cat file1_lines6-7.txt file2.txt file3.txt file4.txt > file5.txt
```

cat command can also be used to open and visualize a file.

```
cat file5.txt
```

In bioinformatics, algorithms often need to work with sorted files. To do so, you can use **sort** command.

```
sort file5.txt | head -n 20  
sort -k1,1 file5.txt | head -n 20  
sort -k1,1r file5.txt | head -n 20  
sort -k1,1 -k2,2n file5.txt | head -n 20
```

-k1,1 is used to sort by specific column, here the first column. It is by default a lexicographical sort. If you want a reverse sort, use **r** and if you want a numeric sort use **n**.

Option	Action
-n	Numerical sorting.
-f	No differentiation is made between lowercase and uppercase.
-b	Ignore spaces
-r	Reverse sorting
-M	Sort by months

Figure 2: Exemple of sorting options

You will often need to select particular columns from your files. Use the command ‘cut’.

```
cut -f1-3 -d$'\t' file5.txt | head
```

Here we extract the first 3 columns (**-f**) from file1 where the column’s separator is tab (‘\t’) and print the first 10 lines (**head**).



Using R

Sometimes, you might want to use *R* on the terminal. We will see more about *R* tomorrow, but from now just try opening *R* by typing:

```
R
```

To close it, press Ctrl+D.

How to get help

You have seen some of the UNIX commands that every bioinformaticians use routinely. Don’t hesitate to come back to this tutorial.

A last tip: all tools and commands will give you details about their usage and options. Always try these command to display help.

```
sort --help
cut --help
grep --help
bedtools --help
```