The Unix command line

Dr. Umar Ahmad

September 3, 2024

#### Learning Objectives

* Recognise why the Unix command line is essential for bioinformatic analysis.
* Explain how the location of files and folders is specified from the command line.
* Memorise and apply key commands to navigate the filesystem and investigate the content of text files.
* Combine multiple commands to achieve more complex operations.

Learning the Unix command line is critical for bioinformatic analysis due to its widespread use in the field, particularly in the context of the Linux operating system. The Unix command line offers several key advantages:

* **Ubiquitous in computing**: used across various computing applications and is essential for working with remote servers like those in high-performance computing (HPC) environments.
* **Versatile command set**: provides a vast array of commands that enable intricate file manipulations, including tasks like locating and replacing text patterns. These capabilities are very useful in the field of bioinformatics.
* **Scripting for automation**: users can use, create and share script files to store and execute sequences of commands, facilitating automation and ensuring the reproducibility of analyses.

In summary, mastering the Unix command line enables bioinformaticians to efficiently handle data, automate workflows, and enhance the reliability of their research.

In this section we give a very brief overview of some of the key Unix commands needed to follow these materials. For a more thorough coverage of this topic, see our accompanying materials: [Introduction to the Unix Command Line](https://cambiotraining.github.io/unix-shell/).

## The command prompt

When you open a terminal you are presented with a command prompt, waiting for you to input a command. It will look something like this:

username@computer-name:~$ |

It gives you information about:

* Your username
* The name of your computer
* The location in your filesystem (~ indicates your home directory)
* A separator, usually $ symbol
* The prompt (often blinking) waiting for your command input

## Navigating the filesystem

The location of files in Unix is represented as a **file path**. For example:

/home/participant/Documents

Indicates the “Documents” folder of a user called “participant”. The first / at the beginning of the path indicates the *root* (or start) of the filesystem.

Paths can be specific in two ways:

* **Absolute path:** specify the full path starting from the *root*. These paths *always start with /*.
* **Relative path:** specify the path starting from your current location. For example, if you are located in /home/participant, the path Documents/resources would be equivalent to /home/participant/Documents/resources. Relative paths *never start with /*.

Here are some key commands to navigate the filesystem:

* pwd prints your current directory
* cd changes directory
* ls lists files and folders
* \* is known as a “wildcard” and can be used to match multiple files

For example:

pwd

/home/participant

Change to the “resources” folder, located within “Documents”:

cd Documents/resources

List the files within that folder

ls

CheckM2\_database bakta\_db mash\_db reference vibrio\_genomes

Commands have **options** that can change their behaviour, for example:

ls -l reference

-rwxr--r-- 1 ubuntu ubuntu 2269140 Sep 7 09:34 annotation.gff  
-rwxr--r-- 1 ubuntu ubuntu 139 Sep 7 09:34 count\_proteins.sh  
-rwxr--r-- 1 ubuntu ubuntu 4098588 Sep 7 09:34 genome.fasta  
-rwxr--r-- 1 ubuntu ubuntu 1413435 Sep 7 09:34 proteins.fasta

The -l option lists the files in a long format. We also specified that we wanted to list the files inside the reference folder (instead of the default, which lists files in the current directory).

You can see all the options available to a program by looking at its help/manual page: man ls or ls --help.

The **wildcard** \* can be used to match files that share part of their name. For example:

ls reference/\*.fasta

reference/genome.fasta reference/proteins.fasta

Only matches the files with .fasta extension.

## Files and folders

Here are some key commands to create directories and investigate the content of text files:

* mkdir creates a directory
* head prints the top lines of a file
* tail prints the bottom lines of a file
* less opens the file in a viewer
* wc counts lines, words and characters in a file
* grep prints lines that match a specified text pattern

To create a directory called “test” you can run:

mkdir test

To look at the top lines of a file you can use:

head genome.fasta

>NZ\_CP028827.1 Vibrio cholerae strain N16961 chromosome 1, complete sequence  
GTGTCATCTTCGCTATGGTTGCAATGTTTGCAACGGCTTCAGGAAGAGCTACCTGCCGCAGAATTCAGTATGTGGGTGCG  
TCCGCTTCAAGCGGAGCTCAATGACAATACTCTCACTTTATTCGCCCCGAACCGCTTTGTGTTGGATTGGGTACGCGATA  
AGTACCTCAATAACATCAATCGTCTGCTGATGGAATTCAGTGGCAATGATGTGCCTAATTTGCGCTTTGAAGTGGGGAGC  
CGCCCTGTGGTGGCGCCAAAACCCGCGCCTGTACGTACGGCTGCGGATGTCGCGGCGGAATCGTCGGCGCCTGCGCAATT  
GGCGCAGCGTAAACCTATCCATAAAACCTGGGATGATGACAGTGCTGCGGCTGATATTACTCACCGCTCAAATGTGAACC  
CGAAACACAAGTTCAACAACTTCGTGGAAGGTAAATCTAACCAGTTAGGTCTGGCCGCGGCTCGCCAAGTCTCTGATAAC  
CCAGGTGCGGCGTATAACCCCCTCTTTTTGTATGGCGGCACCGGTTTGGGTAAAACGCACTTGCTGCATGCGGTGGGTAA  
CGCGATTGTTGATAACAACCCGAACGCTAAAGTGGTGTACATGCACTCTGAGCGTTTCGTGCAAGACATGGTAAAAGCCC  
TGCAGAACAACGCGATTGAAGAATTCAAACGCTACTATCGCAGTGTAGATGCCTTGTTGATCGACGATATTCAATTCTTT

You can print only ‘N’ lines of the file using the following option:

head -n 2 genome.fasta

>NZ\_CP028827.1 Vibrio cholerae strain N16961 chromosome 1, complete sequence  
GTGTCATCTTCGCTATGGTTGCAATGTTTGCAACGGCTTCAGGAAGAGCTACCTGCCGCAGAATTCAGTATGTGGGTGCG

The tail command works similarly, but prints the bottom lines of a file.

To open the file in a viewer, you can use:

less genome.fasta

You can use ↑ and ↓ arrows on your keyboard to browse the file. When you want to exit you can press Q (quit).

To count the lines in a text file you can use:

wc -l genome.fasta

50601 genome.fasta

To print the lines that match a pattern in a file you can use:

grep ">" genome.fasta

>NZ\_CP028827.1 Vibrio cholerae strain N16961 chromosome 1, complete sequence  
>NZ\_CP028828.1 Vibrio cholerae strain N16961 chromosome 2, complete sequence

## Combining commands

You can chain multiple commands together using the **pipe** operator. For example:

grep ">" genome.fasta | wc -l

2

* First find and print the lines that match “>”
* And then count the number of lines from the output of the previous step

In this case, the wc command took its input from the pipe.

## Summary

#### Key Points

* The Unix command line is essential for bioinformatic analysis because it is widely used in the field and allows for efficient data manipulation, automation, and reproducibility.
* The location of files and folders from the command line using either absolute or relative paths.
  + Absolute paths always start with / (the root of the filesystem)
  + Subsequent directory names are separated by /.
* Key commands to navigate the filesystem include: cd and ls
* Key commands to investigate the content of files include: head, tail, less, grep and wc.
* The *wildcard* \* can be used to match multiple files sharing part of their name.

# For further reading: The GNU project

<https://www.gnu.org/gnu/thegnuproject.en.html>

# Homework:

Go to: <https://swcarpentry.github.io/shell-novice/>

Work through the tutorial on the Unix Shell from Software Carpentry and create a notes file. Turn in your notes by email before class on Monday. Note that the recommended time to teach this entire tutorial is 4.5 hours. I don’t know how long it will take each of you independently to work through it but I think more than 1 hour should be a minimum.