Table of contents

Setting up a terminal

Terminology

The **Linux command-line interface (CLI)** is an alternative to a graphical user interface (GUI) with which you are likely more familiar. Both interfaces allow you to interact with an operating system. The key difference between the CLI and GUI is that the interaction with the CLI is based on issuing commands. In contrast, the interaction with a GUI involves visual elements, such as windows, buttons, etc. The CLI is also referred to as the shell, terminal, console or prompt.

Bash is a type of interpreter that processes shell commands. A shell interpreter takes commands in plain text format and calls the operating system to do something, for example changing a directory or modifying the content of some files. Bash itself stands for Bourne Again Shell and it is one of the popular command-line shells used to run other programs, many of which are useful for bioinformatic workflows. In this tutorial, we assume that you work with a Bash shell.

Installation guides

Linux

The default shell is usually Bash and there is no need to install anything to be able to follow this tutorial. On most versions of Linux, the shell accessible by running the Gnome Terminal or KDE Konsole or xterm, which can be found via the applications menu or the search bar. If your machine is set up to use something other than Bash, you should be able to switch the shell by opening a terminal and typing `bash`.

Mac

For Mac running macOS Mojave or earlier releases, the default Unix Shell is Bash. For a Mac computer running macOS Catalina or later releases, the default Unix Shell is Zsh. To open a terminal, try one or both of the following:

- In Finder, select the Go menu, then select Utilities. Locate Terminal in the Utilities folder and open it.
- Use the Mac 'Spotlight' search function. Search for: Terminal and press Return.

To ensure that you work with a consistent shell and to check if your machine is set up to use something other than Bash, type `echo $SHELL` in your terminal window. The name of the current shell should be printed to the terminal window.

If your machine is set up to use something other than Bash, you can try switching to Bash by opening a terminal and typing `bash`. To check if that worked type `echo $SHELL` again. If you have trouble switching from zsh to bash, do not worry, most steps in the tutorial work with either shell.

Windows

Operating systems, like macOS and Linux, come with a native command-line terminal, making it straightforward to run bash commands. However, Windows users need to install some software first to be able to use bash. Below you find three options:

1. **Mobaxterm** enables Windows users to execute basic Linux/Unix commands on their local machine, connect to an HPC with SSH and to transfer files with SCP/SFTP (more on that later). Installation instructions can be found here. This option should be the easiest to setup if you have little experience with the shell.
2. **Git Bash** is another option, for detailed installation instructions please have a look at the carpenties website.
3. The **Windows Subsystem for Linux (WSL2)** lets users install a Linux distribution (such as Ubuntu, which is the default Linux distribution, which we recommend to use) and use Linux applications, utilities, and Bash command-line tools directly on Windows. This option is the most flexible as you have a full Linux system at your command but since you basically install a separate system on your PC you need to have enough memory to run this. Installation instructions can be found here.

> **Note**
>
> I am myself familiar with Linux, Mac and WSL and the following tutorial is tailored towards the location of things when using WSL. If you use another system that is no problem, however, your folder structure might be slightly different when using Git Bash or Mobaxterm.
> Similarly, I am mainly familiar with the bash not the zsh shell. For Mac users that have trouble switching to bash this might create some issues when using wildcards but these users should be able to otherwise follow most parts of this tutorial.

> If parts of the tutorial do not work for you due to issues when working with different operating systems/shells, feel free to contact me and I can adjust the tutorial accordingly.

Sanity check

After you set everything up and opened a terminal you should see something like this and are good to go if you want to follow the tutorial: