Day 1: Exercise walkthrough

All exercises in this course will be performed in the terminal. Detailed descriptions of every command can be found in the Handout. We have provided the commands for the first exercise to help you get started. As the course progresses, you will need to refer to the handout to find the appropriate commands and adapt them to your specific tasks.

Exercise 0: Setting the stage

The first step of any analysis performed in the terminal is to set up your workspace. Please follow the instructions below:

- Open the terminal and make sure you are in 'home' by typing the following command and pressing enter:

  ```
  cd ~
  ```

- Get a sense of which files and subdirectories are already inside your current working directory by listing its contents with the following command in the terminal:

  ```
  ls
  ```

- From this directory, we will download the data needed for the course. **wget** is a command-line utility for downloading files from the web. Copy and paste the following, then press enter:

  ```
  wget https://github.com/KasperThystrup/Unix4Beginners/raw/zippy/Unix4Beginners.tar.gz
  ```

- List the content of your working directory again and notice the new compressed file.

  ```
  ls
  ```

- Decompress the file (extracting the contents) by using the command below (this command will be further introduced later in the course):

  ```
  tar -zxvf Unix4Beginners.tar.gz
  ```

- List the content again and **notice the difference in color**. The file has changed from a zip file to a directory you can enter. To enter this directory through the terminal, run the following command:

  ```
  cd BacterialData
  ```

- The stage is now set, let's get on with the exercises.

Before starting Exercise 1, we highly encourage you to read and understand Chapter 1 in the Handout.

Exercise 1: Exploration of the data

Our supervisor has assigned us the task of running ResFinder on several bacterial assemblies to identify antimicrobial resistance genes. We have been given a disorganized directory from the supervisor, which we must organize before proceeding with the analysis. But first, let's investigate the directory's content. In these exercises, we will explore the BacteriaData directory to understand how the data is stored.

- Exercise 1A: How many subdirectories are present within the directory called "E_coli"?
- Exercise 1B: In which subdirectory can the file "E_coli_CP127297.fasta" be found?
- Exercise 1C: How many files are in the "Virus" subdirectory?
- Exercise 1D: What is the full path for the file named "R1_E_coli_02.fq"?

Before starting Exercise 2, Exercise 3, and Exercise 4, we highly encourage you to read and understand the Chapter 2 in the Handout.

Exercise 2: Cleaning it up!

Now, it is time to clean up the messy directory! In this process, we will learn how to create new directories, move files, and rename files.

- Exercise 2A: Move the assembly file "E_coli_WWcol315.fasta" into the "assemblies" sub-directory within the "E_coli" directory.
- Exercise 2B: Move the read files "R1_E_coli_01.fq" and "R2_E_coli_01.fq" into the "reads" sub-directory within the "E_coli" directory.

We will now create a directory named 'P_aeruginosa' to organize the reads and assemblies from the messy directory. Then, we will place all the assemblies in a subdirectory called 'assemblies' and all the reads in a subdirectory called 'reads'.

- Exercise 2C: In the "BacteriaData" directory, create a subdirectory named "P_aeruginosa".
- Exercise 2D: In the "P_aeruginosa" directory, make a subdirectory called "reads".
- Exercise 2E: In the "P_aeruginosa" directory, make another subdirectory called "assemblies".

All files containing reads start with the letter 'R' and have an .fq extension. All files containing assemblies have a .fasta extension.

- Exercise 2F: Move all read files into P_aeruginosa/reads using only a single command.

You may notice that one of the assemblies has a misspelling that needs to be corrected.

- Exercise 2G: Rename "P_oeruginosa_PPF1.fasta" to "P_aeruginosa_PPF1.fasta"
- Exercise 2H: Move all assembly files into P_aeruginosa/assemblies. Try to do it with a single command.

CHECKPOINT 1

Before starting Exercise 2, Exercise 3, and Exercise 4, we highly encourage you to read and understand the Chapter 2 in the Handout.

Exercise 3: Removing redundant content

Aside from moving files, you should delete any files that do not belong to this project. Let's get rid of the noise!

- Exercise 3A: Remove the "Credit_cards.txt" file from "BacteriaData" (nothing to see here, we promise, delete!).
- Exercise 3B: Remove the "Virus" directory and everything in it. In this lab, we are only working with bacteria.

Exercise 4: Prepare for manipulation

Your supervisor wants to run ResFinder on the assemblies. To do so, you need to copy the assembly files into a new directory called "Resfinder."

- Exercise 4A: Make a new subdirectory called "Resfinder" inside "BacteriaData".
- Exercise 4B: Copy the assembly file named "E_coli_ASM584v2_reference.fasta" from E_coli/assemblies and place the copy in the "Resfinder" directory.

Some programs require running on an entire directory, necessitating specialized directories for such analysis. Having multiple copies of the same files is inefficient and costly. Instead, use symbolic links to these files, which are space-efficient and avoid data duplication. More details can be found in the handout!

- Exercise 4C: Create a symbolic link to "E_coli_ASM584v2_reference.fasta" from the E_coli/assemblies directory in the "Resfinder" directory.
- Exercise 4D: Create symbolic links to all assembly files in the "E_coli" and "P_aeruginosa" directories within the "Resfinder" directory.

CHECKPOINT 2

Before starting Exercise 5, we highly encourage you to read and understand Chapter 3 in the Handout.

Exercise 5: Inspection

It is time to inspect the files inside the "Resfinder" directory and ensure everything is in order before we start the analysis.

- Exercise 5A: Inspect the assembly file "P_aeruginosa_TOprJ3-positive_part1.fasta" using the **cat** command.
- Exercise 5B: Inspect the assembly file "P_aeruginosa_TOprJ3-positive_part2.fasta" using the **less** command. Observe the difference between **cat** and **less**. Which would you use for a large text file? Note: press "q" to exit **less**.

Opening and navigating through an entire file is sometimes unnecessary when you are only interested in the beginning or end of a text file.

- Exercise 5C: Get the first three lines of "P_aeruginosa_TOprJ3-positive_part2.fasta" using the **head** command.
- Exercise 5D: Get the last five lines of "P_aeruginosa_TOprJ3-positive_part2.fasta" using the **tail** command.

You might have noticed that the first line of "P_aeruginosa_TOprJ3-positive_part2.fasta" is missing a ">". This ">" is important because it designates the line as a FASTA header.

- Exercise 5E: Open "P_aeruginosa_TOprJ3-positive_part2.fasta" using the **nano** command and add the missing ">" to the faster header.

*Now, we have organized the messy directory and inspected its content! **A job well done**! Tomorrow, we will further manipulate said files and prepare them for analysis.*

Extra exercise: Options to ls!

*You can customize the ls command by adding options. When dealing with many files, the default **ls** output can be cluttered. Try using the **-l** option for a long listing format and **-lh** to make file sizes human-readable. Observe the difference in output between these options and the default **ls**.*

- Extra 5A: What is the largest file in P_aeruginosa/assemblies?
- Extra 5B: Most commands have multiple options, and each command has a help page that can inform you more about said options. Using the "--help" option, check out the options from the "mv" command. Feel free to check out the --help option on other commands.