Module C: More Command-line

You can choose from the following three tasks:

1. **Key Concepts in Unix** – These are additional exercises from yesterday's command-line session.
2. **Organizing the New Project** – Focuses on repeating key commands and navigating the terminal efficiently.
3. **A Deeper Dive** – Covers grep, wc, and piping for more advanced command-line operations.

Recommendation:

- If you did not complete the extra exercises from yesterday's session, we recommend reviewing **Key Concepts in Unix** first.
- If you're still getting comfortable with the command line, start with **Organizing the New Project**.
- If you're confident, skip ahead to **A Deeper Dive**.

Key concepts in Unix

Extra Exercises 1: More fun with grep!

When using grep, we have encouraged you to search using quotation marks " ". While this is more of a best practice than a strict requirement, it helps avoid potential issues. In this exercise, we will explore a scenario that can occur if it is not used. You will be working with the file "P_aeruginosa_TOprJ3-positive.fasta".

- Extra 1A: Use **grep** to save the header names in a file called "header_names.txt". View the file using **less**.
- Extra 1B: Let's discover what happens when you don't use proper quotations with **grep**. Display the content of header_names.txt using less. Run the following command:

```
grep > header_names.txt
```

Then display the content of header_names.txt again.

What happened and why? Discuss with your partner.

Extra exercise 2: 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 2A: What is the largest file in P_aeruginosa/assemblies?

Extra Exercises 3: Save space – use symbolic links

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

- Extra 3A: Make a new subdirectory called "Resfinder" inside "BacteriaData".
- Extra 3B: Copy the "E_coli_ASM584v2_reference.fasta" assembly file 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!

- Extra 3C: Create a symbolic link to "E_coli_ASM584v2_reference.fasta" from the E_coli/assemblies directory in the "Resfinder" directory.
- Extra 3D: Create symbolic links to all assembly files in the "E_coli" and "P_aeruginosa" directories within the "Resfinder" directory.
- Extra 3E: List the contents of the "Resfinder" directory and observe the visual differences between symbolic links and actual file copies.

Extra Exercises 4: Files for a Colleague – Compressing files

A colleague also needs to run ResFinder and would like a copy of your files. Let's be helpful and provide her with that.

- Extra 4A: Compress the "Resfinder" directory using the **tar** command and name the resulting gzipped file "ResFinder.tar.gz". Using the tar command, ensure that tar zips the real files through the symbolic links, not just the symbolic links themselves. Make sure you're not inside the "Resfinder" directory when you do this.
- Extra 4B: Retrieve the path for the "ResFinder.tar.gz" file so you can share it with your colleague.

Organizing the New Project – repetition

Exercise: The "other" project

Your supervisor realized they forgot to include some assemblies for the ResFinder analysis. Even if it's a bit late in the process, you know what to do, so let's get started!

- Exercise 1A: Unzip the new assemblies from SorryForAddingTheseLate.tar.gz from the Day2 folder using **tar**:

These files are a mixed collection that doesn't fit our previous categories, but we'll still organize them using the same directory structure.

- Exercise 2B: Create a directory named "other" in BacterialData, and inside it, create another directory called "assemblies". Then, move all the new FASTA files into the "assemblies" directory.

For some reason, the file "our_first_nanorepore.fasta" is split into three.

- Exercise 3C: There is no reason why the file needed to be split up for the ResFinder analysis. Combine the different parts of "our_first_nanopore_part[1-3].fasta" into one file called "our_first_nanopore.fasta".
- Exercise 4D: Create symbolic links for all the assemblies from the "other" directory in the "Resfinder" directory.
  (If you haven't done the Extra exercises: Make a new subdirectory called "Resfinder" inside "BacteriaData")

A Deeper Dive

More grep: Finding elements in files

Tellurite resistance genes are crucial for bacterial survival in toxic environments, especially in pathogenic bacteria, as they help resist host defenses and antimicrobial treatments. Identifying and studying these genes is essential for understanding resistance mechanisms and developing strategies to combat resistant strains.

A colleague afraid of computers is concerned that the known tellurite resistance gene, terA, might have been accidentally omitted from one of the assemblies and has expressed interest in confirming its presence. This is a good opportunity to explore one of the most powerful command-line search tools. For this exercise, you will work specifically with the file ***P_aeruginosa_TOprJ3-positive.fasta***.

- Exercise 1A: Use **grep** to extract any line containing "tera" (ignoring case).
- Exercise 1B: Use **grep** to extract any line containing "terA" (ignoring case) and the subsequent five lines to verify if the sequence is present, not just the header.

Once you've confirmed that the sequence is present, you will need to determine its location in the file so your supervisor can further investigate.

- Exercise 1C: Determine the line number where "terA" appears in the file.

Your colleague also wants to know what other Tellurite Resistance genes are in the file.

- Exercise 1D: Search for all sequences that contain "ter". What changes do you notice when you add the -o option?

Counting occurrences: grep and wc through piping

Your colleague is interested in determining how often the sequence "TTTTTT" appears in one of the FASTA files. You'll combine the grep and wc commands using a pipe to accomplish this. We're using this approach instead of simply adding the -c option to grep because we want to count the total occurrences of "TTTTTT", not just the number of lines it appears on, as there could be multiple instances of "TTTTTT" on a single line.

- Exercise 2A: Determine how often the sequence "TTTTTT" appears in the "P_aeruginosa_TOprJ3-positive.fasta" file.

If you want to see how grep counts occurrences, try searching for "TTTTTT" with the --color option. This will allow you to visualize how it highlights each match.