Celine Al-Noubani

BIOL 7200 Exercise 1

1. Using documentation to explore functionality of ls

* List the files in your home directory
* Create two empty files in your home directory. One named "file1" and one named ".hidden\_file"  
  (note the dot in the second name)

A screen shot of a computer

Description automatically generated

* List all the files in your home directory sorted with oldest first (hint: "all" means the hidden file you made should be listed)



* What is the size of file1? Show your working
* What is the size of ".hidden\_file"?

A black background with white text

Description automatically generated

2. Creating and viewing file contents using the terminal

* Add two lines of text to "file1"
* View the contents of "file1" in your terminal

A computer screen with white text

Description automatically generated

3. Copying and removing files

* Use cp to copy "file1" to "file1\_copy.txt"



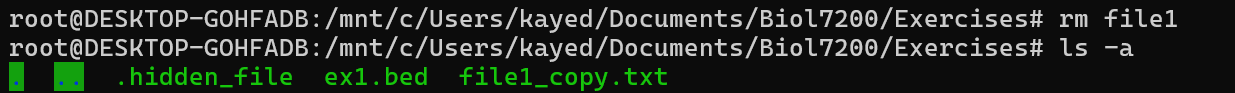
* Has the addition of ".txt" to the file name changed how the file contents look? Are file extensions significant in Unix systems?

A black screen with white text

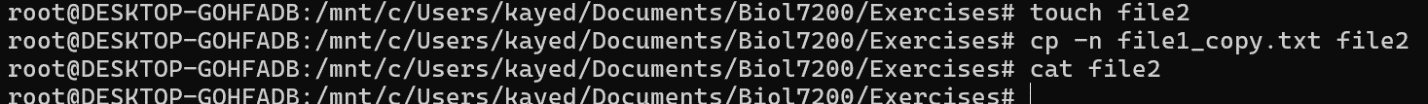
Description automatically generated

No, the file contents look the same. The file extension is not significant.

* Use rm to remove "file1"



* Create an empty file named "file2"
* Run the command cp -n file1\_copy.txt file2. Does "file2" now contain the same contents  
  as "file1\_copy.txt"? Explain why or why not



No, file2 does not contain any content, let alone the same contents as file1\_copy.txt. This is because file2 already existed before the cp - n file1\_copy.txt file2 command was run as an empty file and the -n flag prevents overwriting.

4. Using documentation to explore useful commands. State the command and options you could use to perform the following tasks:

* Create a directory structure "~/a/b/c" in a single command (i.e., create a directory and any  
  missing parent directories)

mkdir -p ~/a/b/c

* Check if a file has Windows- or Unix-line endings

file <filename>

* Copy files but only replace existing files if they are older than the source file

cp -u <source> <newplace>

* Check if whitespace characters in a file are tabs or spaces

To check for tabs: grep -P ‘\t’ <filename>

To check for spaces: grep ' ' <filename>

* View the last five commands you issued

history | tail -n 5

5. For each row of the below table, provide a glob or extended glob pattern that would match the set of filenames in the second column and not match the set of filenames in the third column. Give 1 pattern for each row of the table

A white text box with black text

Description automatically generated

#1. ls \*.txt \*.tsv \*.tiff

#2. ls @(SRR|ERR)[0-9]

#3. ls \*txt \*pdf

#4. ls \*@(reads\_[0-9].fastq |\_[0-9].fq)

#5. ls Samples/[ab]/assembly.fasta

6. Redirecting outputs

* Pick a command that produces stdout, run it, and direct its stdout to a file
* ls a path that does not exist in your current directory. Which output stream does the message you see come from?

It’s coming from the strerr (standard error).

* Rerun the command from step 2, but now direct the output to a file
* In a single command, ls both a path that does not exist (e.g., "~/not/a/real/path") and "./" (i.e., provide two positional inputs). Direct the stdout to one file and the stderr to another file (still within a single command - no use of ;).
* Use grep to find the help message entry for the "-l" option of ls (hint: "-" is a special character interpreted by bash so you need to get around that somehow)
* How many commands are there in your "/bin" dir?

A black screen with white text

Description automatically generated

A screen shot of a computer

Description automatically generated

7. Data cleaning. Bioinformaticians often have to work with data generated by others. Perform the following operations to tidy the data in file "ex1.bed" provided on Canvas

* Check if the file uses windows line endings instead of unix line endings



* Remove the windows line endings and output the new version to a new file, preserving the original file (always good practice)
* Remove the header lines starting with "#" and output the new version to a new file



8. Summarizing real data using bash commands. The following questions relate to the cleaned version of the "ex1.bed" file you generated above. BED format is a commonly used format for storing the location of features in an assembly. The provided file includes the three mandatory columns of a BED file: Sequence ID, start, and stop positions (further description of BED format can be found here). Using Bash commands answer the following questions about these data

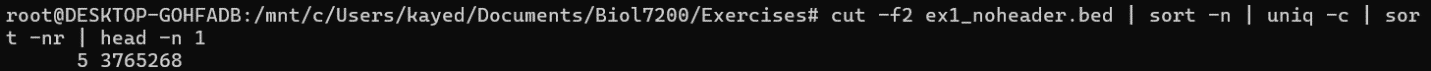
* How many unique sequence IDs are present in the file?



* How many different start positions are there?



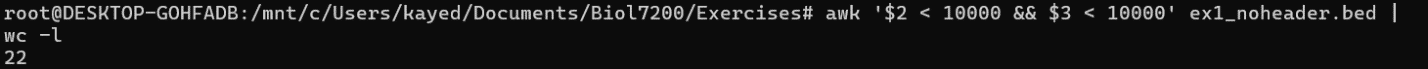
* What is the highest number of features starting at the same start position?



* How many features start in the first 10Kbp of the sequence?

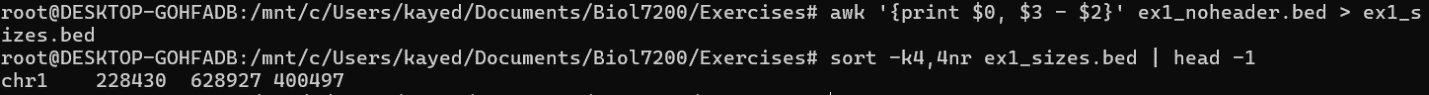


* How many features start and end in the first 10Kbp of the sequence?



EXTRA CREDIT (5 points)

* Which feature is the largest? Show your work



The awk command calculates the size of each feature, then the sizes are arranged in descending order using the sort command, and finally the head -1 command retrieves the first line which is in this case the largest feature. From the result, the largest feature is 400,497 base pairs.