# Linux Exercises

# Moving and Looking Around

It is good to know where you are in the file system.

1) Let's start by checking which directory you are in. How can you print your working directory?
2) Have a look at what is inside it with the command ls. Now try it with the options –a and –l. What's the difference? Use the command ll instead. What happens?
3) Too many things on your screen now? Clear it with the command "clear".

We should also be able to create directories and move between them.

1) Create a new directory called *genome_analysis* and include a second directory called *linux_exercises* inside of it.
2) Go into the *linux_exercises* directory.
3) From inside the *linux_exercises* directory, check the content of your root directory using ls or ll. Can you see all the content of the root directory without having to be there?
4) Go back to *genome_analysis*. From here, create another directory called *test directory* (using spaces instead of underscores to separate the words!). Did it work? If not, how can you make it work?

# Reading files

We need a file with something in it to practice how to read files, so let's copy the manual of the program "ls" into a text file. Go to the directory linux_exercises and type:

```
$ man ls > manual.txt
```

Now:

1) Print the content of manual.txt to stdout
2) Can you see the content of manual.txt WITHOUT printing it to stdout?
3) Print only the first 10 lines of manual.txt. Can you also print only the first 30 lines?
4) Print only the last 10 lines of manual.txt. Can you also print only the last 15 lines?

The command "wc" can give you some extra information about text files. Use it and find out:

5) How many lines does the file manual.txt have?
6) How many words are in it?

Tip: If in doubt about how to use a command, check its manual!

# Writing to files and concatenating them

First, let's create an empty file. Go to the directory linux_exercises and type

```
$ touch file1.txt
```

Then let's write something inside it. The command "echo" prints something to stdout. As an example, try typing:

```
$ echo Biology is fun!
```

1) Using the command echo, can you write the sentence "Biology is fun" inside file1.txt?
2) Also using echo, can you write the sentence "Bioinformatics is also fun" inside file1.txt WITHOUT overwriting the sentence you had written there before?
3) Create another file called file2.txt and write the sentence "That's why I like Genome Analysis" in it. Concatenate the contents of file1 and file2 in a third file called file3.txt
4) Using the command "wc", can you find out how many characters there are in file1.txt? Was the number what you expected? If not, what is happening?

# Copying and moving files

For this part we will use the files 1, 2 and 3 which you just created.

1) While in the directory *linux_exercises*, create a copy of file1.txt called file1.copy inside the same directory.
2) Now create a copy of file1.txt called file2.txt. Check the content of file2.txt. What happened to the file2.txt that you had before? Be careful with this when using Linux! There is no "Are you sure you want to do this?" message when you overwrite or delete files!!!
3) Without going to another directory, create a copy of file2.txt called file2.copy inside *genome_analysis*. Now go to *genome_analysis* and see if your copy of file2 is there.
4) From *genome_analysis*, create a copy of file3.
5) Use the command "mv" to rename the copy of file3 to file4.txt
6) Now, from *genome_analysis*, move the files file2.copy and file4.txt, which should be in *genome_analysis*, to the directory *linux_exercises*
7) With a single command, create a copy of the directory *linux_exercises* and all files inside it. Call it *linux_exercises_2*

# Grep

Go to the directory *linux_exercises*. We are going to use the file manual.txt that you created before. Using the command grep:

1) Print only the lines of manual.txt which contain the word "file"
2) Print THE NUMBER of lines of manual.txt which contain the word "file"
3) Print the lines of manual.txt which contain the word "file", no matter if the word is written in low or high caps

4) Print THE NUMBER of lines of manual.txt which contain the word “file”, no matter if the word is written in low or high caps. Did you get the same number as before?
5) Now print only the lines which DO NOT contain the word “file”
6) Finally, print THE NUMBER of lines of manual.txt which contain the symbol >

WARNING: Note that grep can accept patterns that are quoted and not quoted. For example: *grep “hello”* and *grep hello* would work identically. However, having special characters in non-quoted patterns could produce some unexpected behaviors that could end up in some data being lost. For example if the symbol > is not quoted, Bash will interpret it as the sign of “redirection of the stdout into a file”. *grep >seq1* will create/overwrite a file called seq1. The correct way of grepping that pattern would be: *grep “>seq1”*. We advise you to **always use quoted patterns**.

# Pipes

You should be able to visualize the manual of the program “grep” by typing:

```
$ man grep
```

Now WITHOUT saving the content of the manual to any file, use multiple commands linked by pipes to:

1) Print only the first 10 lines of this manual
2) Count how many times the word “line” appears in the manual
3) Count how many times the word “line” appears only on the first ten lines of the manual
4) Count how many lines of the manual DO NOT contain the symbol >

# A bit more advanced

Let’s work now with a fasta file.

1) Download the genome of *Sulfolobus solfataricus* using the wget command:

```
$ wget
    ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/007/005/GCF_000007005.1_ASM700v1/GCF
    _000007005.1_ASM700v1_genomic.fna.gz
```

2) Check if the file is found in our directory. Note the suffix .gz. This means that the file is compressed. Try the command “cat” on this file. What do you see? Now, try “less”. Do you see the same?
3) Uncompress this file. Try the command “cat” again. Did anything change?
4) Let’s count the number of sequences in this file. Remember that in a fasta file each new sequence starts with the “>” symbol followed by the sequence name. How many sequences are there? What are the identifiers of those sequences?
5) Now, let’s count the number of nucleotides in the genome of *S. solfataricus.* How big is this genome? Remember to exclude the fasta headers when counting the characters!
6) One problem with “wc” is that it also counts newline-characters as characters, and therefore the calculated genome length won’t be accurate. However we can remove the

newline-characters using “tr” before counting the characters (see the manual page for more information). How big is the genome length now?

# Connecting to Uppmax and transferring files from remote servers

During our computer labs we’ll be working mostly on Uppmax, and sometimes we may have to transfer files from Uppmax to the local Linux server. Let’s try the commands now to make sure things will work later.

1) Connect to Uppmax using the command “ssh”
2) Go to the directory */proj/genomeanalysis2022/nobackup/linux* and check if there is a file called transferme.txt inside it. Now copy this file from Uppmax to the local Linux server using rsync.

Note: you need to be in the local server –the destination server – to do the transfer.

3) After you have done the transfer, exit Uppmax, if you are still connected to it. You can use the command “exit” for that.

# Deleting files

Make sure you are in the local server now, and NOT in Uppmax!

We are almost done with our Linux practice, so let’s clean up the directories and files we created, as we won’t be using them anymore.

1) Go to the directory *linux_exercises*. From here, delete all the files inside it, leaving the directory empty.
2) Go back to the directory *genome_analysis*. From here, delete the empty directory *linux_exercises*.
3) Now delete the directory *linux_exercises_2* and all files inside it by using a single command
4) Finally, delete the directory called *test directory*, which you created before.

If everything worked well up to here, you’re done with the Linux practice and ready for our computer labs! See you tomorrow!