



Bacterial WGS training : Exercise 1

Title	Linux command line.
Training dataset:	None
Questions:	<ul style="list-style-type: none">• How do I use the command line?• How do I navigate the file system?
Objectives:	<ul style="list-style-type: none">• Learn/Remember how to use the command line.• Learn/Remember how to navigate through the Linux file system.
Time estimation:	30 min
Key points:	<ul style="list-style-type: none">• Remeber the shell basic commands: <i>pwd cd ls mkdir rmdir mv nano cat less more head tail rm</i>

!! Important things to remnenber:

- Use Tab to automatically complete file names and paths, so it can be easiert to write in the terminal
- Use keyboard arrows ( ) to move through your terminal's history, so you don't have to write the commands again.
- Try not to use spaces, accents or special characters like "Ñ" letter, when writting directory of file names.
- Basic commands you should always remember: *pwd cd ls mkdir mv rm rmdir less nano*

Answering to main questions

How do I use the command line?

Open a terminal by clicking in the icon or typing **Ctrl+Alt+T**. Now you can type in the prompt.

How do I navigate the file system?

Let's remember the basics: *pwd cd ls mkdir mv rm rmdir less nano*. We are going to use those commands to:

Checking the working directory (pwd)

Check our working directory:

```
pwd
#Output: /home/alumno
```

Moving between directories (cd)

Move to our Desktop folder:

```
cd ~/Escritorio
pwd
#Output: /home/alumno/Escritorio
```

Move to the course folder:

```
cd
pwd
#Output: /home/alumno
cd wgs
pwd
#Output: /home/alumno/wgs
cd bacterial_wgs_training_dataset/RAW/FULL_DATA
pwd
#Output: /home/alumno/wgs/bacterial_wgs_training_dataset/RAW/FULL_DATA
cd .
pwd
#Output: /home/alumno/wgs/bacterial_wgs_training_dataset/RAW/FULL_DATA
cd ..
pwd
#Output: /home/alumno/wgs/bacterial_wgs_training_dataset/RAW/
cd ../../..
pwd
#Output: /home/alumno/
```

Questions:

- Which is the meaning of the "~" symbol?

It is the abbreviation of `/home/user/` path

- What does the `cd` command without arguments do?

It changes the current directory to `/home/user/` path.

- What does "." mean?
- What does ".." mean?

Listing directories (ls)

```
cd wgs
cd bacterial_wgs_training_dataset
```

```
ls
#Output: ANALYSIS RAW REFERENCES RESULTS
```

This is the folder structure we will use for this training. Now we are going to list the files in the **REFERENCE** folder:

```
ls REFERENCES
```

This command will output a big list of files, which are the files that we will use as REFERENCE through the different exercises of the training. Now we will run this other command:

```
ls /home/alumno/wgs/bacterial_wgs_training_dataset/REFERENCES/
```

Questions:

- ▶ What does "." mean?
- ▶ Which is the difference between this last command and the previous one?
- ▶ Do they display the same information?
- ▶ Which one is relative path?

```
ls REFERENCES
```

- ▶ Which one is absolute path?

```
ls /home/alumno/wgs/bacterial_wgs_training_dataset/REFERENCES/
```

Let's see different parameters for the **ls** command. Write:

```
ls REFERENCES
ls -l REFERENCES
ls -a REFERENCES
ls -la REFERENCES
```

Questions:

▶

What do the different arguments of **ls** do?

-l : Long listing format: Displays the permission information

-a : All files: Lists also hidden files

-la : Long format listing and hidden files together.

- What does the new file special?

⚠ **REMINDER:** ⚠ EVERY TIME YOU CHANGE DIRECTORY (cd) YOU HAVE TO CHECK YOUR PATH (pwd) AND LIST THE FILES INSIDE (ls) TO CHECK YOU DIDN'T MAKE MISTAKES

Creating and removing directories (mkdir & rmdir)

Now we are going to move to the ANALYSIS folder which is the folder were we will run all the exercises

```
cd ANALYSIS
pwd
#Output: /home/alumno/wgs/bacterial_wgs_training_dataset/ANALYSIS
ls
```

As you can see the folder is empty, so now we will fill this folder. Create a directory for this handson:

Remember: Linux is case sensitive and does not like white spaces in names

```
mkdir 01-handsonLinux
ls
#Output: 01-handsonLinux
```

Now type:

```
mkdir 01-handsonlinux 01-HandsOnLinux
ls
#Output: 01-handsonlinux 01-handsonLinux 01-HandsOnLinux
```

Questions:

- Is it possible to create more than one directory at the same time?
- If the names of the folders are the same, why it creates three different directories?

Now we will remove the extra directories:

```
rmdir 01-handsonLinux 01-HandsOnLinux
ls
#Output: 01-handsonlinux
```

Moving and renaming files (mv)

Move to the new folder

```
cd 01-handsonlinux
pwd
#Output: /home/alumno/wgs/bacterial_wgs_training_dataset/ANALYSIS/01-
handsonlinux
```

We are going to move the hidden file in REFERENCE folder to this directory and then rename it:

```
mv ../../REFERENCES/.ThisIsAHiddenFile .
ls
ls -a
mv .ThisIsAHiddenFile NowImNotHidden
ls -a
ls
```

Questions:

- ▶ Which is the difference between the two `mv` commands?
- ▶ Do you remember what "." mean from the first questions?
- ▶ And ".."?

⚠ **REMINDER:** ⚠ LINUX DOES NOT REQUIRE FILE EXTENSIONS

Editing files and displaying them (nano & cat)

We are going to read the file and edit it:

```
cat NowImNotHidden
#Output: I'm a hidden file.
```

This is not true, so we are going to edit it:

```
nano NowImNotHidden
```

Write: **I'm not a hidden file.**

And **save** it: *Ctrl + O + Intro*

Close the new file: *Ctrl + X*

Now read the new file:

```
cat NowImNotHidden
#Output: I'm not a hidden file.
ls
```

```
#Output: NowImNotHidden  
cd ../../
```

Read files other ways (less & more & head & tail)

And now we will read this file:

```
cat REFERENCES/bacterial_wgs_training_initial.tree  
less REFERENCES/bacterial_wgs_training_initial.tree
```

△ Remember: To close less press "q"

```
more REFERENCES/bacterial_wgs_training_initial.tree
```

△ Remember: To close more press "q"

```
head REFERENCES/bacterial_wgs_training_initial.tree  
tail REFERENCES/bacterial_wgs_training_initial.tree
```

```
head -n4 REFERENCES/bacterial_wgs_training_initial.tree  
tail -n3 REFERENCES/bacterial_wgs_training_initial.tree
```

Questions:



Which is the difference between **head** and **tail**?

Head displays first lines of a file.

Tail displays the last lines of a file



What does the argument **-nX** do to **tail** and **head**?

Displays de X numbers of lines from the begining (head) or end (tail) of a file.

Removing a file (rm)

Now we will learn how to remove files:

```
cd ANALYSIS/01-handsonlinux/  
pwd  
ls  
#Output: NowImNotHidden  
mv ../../REFERENCES/bacterial_wgs_training_initial.tree .  
ls  
#Output: bacterial_wgs_training_initial.tree NowImNotHidden  
rm NowImNotHidden  
ls  
#Output: bacterial_wgs_training_initial.tree
```