- Log into the docker that has been created in lesson 2

First of all, I have to move to the folder "docker", which is present on the Desktop. The name of my docker is r4:v.0.01 → to log the docker the command is docker run -i -t r4:v.0.01

In order for wget to work, I need to install it with apt-get. Then, I move to /bin, I go to the website at the link https://support.10xgenomics.com/single-cell-gene-expression/software/downloads/latest and I copy and paste, on the terminal, the wget command. I run it in /bin → it requires 7 minutes! At the end, a cellranger-6.1.2.tar.gz file will be created.

- I can, eventually, do ls to verify that the file cellranger-6.1.2.tar.gz is present in bin.
- Use gzip -d to unfold the compressed file (ls eventually); then, use tar xvf to unpack the folders.

! Remember to do the commit every time I do something on the docker, otherwise the data will not be saved. I exit from the docker and do: docker commit containerID nameofthedocker

The container ID is the one after the root@, while the name of the docker is always r4:v.0.01

It requires some time to run the command. The name of the containers is different from the 2 images because, when I firstly did the exercise, I did not do the commit, so I did not save anything. I re-did the procedure and committed.

In order to go on with the next steps, I have to mount the folder lesson4_data_analysis (present on the Desktop) on the docker: this can be done with the -v command. GSM4679532_P01 contains the 3 files of interest.

```
exit
PS C:\Users\cauci> docker run -i -t -v C:/Users/cauci/Desktop/lesson4_data_analysis/L4_GSM4679532_P01:/home r4:v.0.01
root@b65e1ddd1dc0:/# cd /home
root@b65e1ddd1dc0:/home# ls
command.sh  data  GSM4679532_P01  __MACOSX
```

Then I create another folder, called "data" (or scratch), with the command mkdir. In "data" I want to put the GSM4679532_P01.csv once it will be created. To do so, I have to write a bash command (command.sh) with blocco note of Windows or with nano.

Command.sh will be automatically added in /home of my docker, and it is rendered executable with chmod +x ./command.sh

In order to run cellranger, a file bashrc must be created in /home. At the end of the file, this string must be added:

```
root@40eef9b4879b:/home# nano ~/.bashrc
root@40eef9b4879b:/home#
```

```
#run cellranger
export PATH=/bin/cellranger-6.1.2:$PATH
```

If the command is run in an interactive way, the terminal is freezed → to run it in background, use the command nohup usr/bin/bash command.sh &

Finally, a file.csv will be created.

(However, I think that this step must be done only once). The command works well both in background and normally. Once I mount the folder (exercise4) in /home, the command to run mat2csv is cellranger mat2csv *nameofthefolder outputname.csv*

```
root@726b68719ae1:/home# ls
barcodes.tsv.gz  features.tsv.gz  matrix.mtx.gz
root@726b68719ae1:/home# cellranger mat2csv GSM4679532_P01 matrix.csv
Input file does not exist: /home/GSM4679532_P01
root@726b68719ae1:/home# exit
exit
PS C:\Users\cauci> docker run -i -t -v C:/Users/cauci/Desktop/exercise4:/home r4:v.0.01
root@4176a1f23563:/# cd /home
root@4176a1f23563:/home# ls
GSM4679532_P01  GSM4679532_P01.zip
root@4176a1f23563:/home# cellranger mat2csv GSM4679532_P01 matrix.csv

    WARNING: this matrix has 36601 x 585 (21411585 total) elements, 95.121487% of which are zero.
    Converting it to dense CSV format may be very slow and memory intensive.
    Moreover, other programs (e.g. Excel) may be unable to load it due to its size.
    To cancel this command, press <control key> + C.

    If you need to inspect the data, we recommend using Loupe Browser.

root@4176a1f23563:/home# ls
GSM4679532_P01  GSM4679532_P01.zip  matrix.csv
```

A few seconds are necessary to make this command run.

Otherwise, to run it in background, a file .sh must be created (in my case, it is called mat.sh) with nano; the string is the following:

Cellranger mat2csv /home/GSM4679532_P01 /home/background_matrix.csv

I changed the name of the final output file in order to not confuse it with the one I previously created.

I render the .sh file executable with chmod +x and then I type nohup ./mat.sh & → if the terminal freezes, type "INVIO" again → Done!

```
root@4176a1f23563:/home/GSM4679532_P01# nano mat.sh
root@4176a1f23563:/home/GSM4679532_P01# chmod +x mat.sh
root@4176a1f23563:/home/GSM4679532_P01# nohup ./mat.sh &
[1] 29
root@4176a1f23563:/home/GSM4679532_P01# nohup: ignoring input and appending output to 'nohup.out'

[1]+  Done                    nohup ./mat.sh
root@4176a1f23563:/home/GSM4679532_P01# ls
barcodes.tsv.gz  features.tsv.gz  matrix.mtx.gz  mat.sh  nohup.out
root@4176a1f23563:/home/GSM4679532_P01# cd ..
root@4176a1f23563:/home# ls
background_matrix.csv  GSM4679532_P01  GSM4679532_P01.zip  matrix.csv
root@4176a1f23563:/home#
```