Data analysis – Prof. Calogero

Lesson 1

REPRODUCIBILITY

During an experiment, the results obtained require the experiment to be performed many times. This is the concept of repeatability = to perform the same experiment with the same reagents and instruments to see if the obtained results are reliable or not.

Reproducibility = some else repeats what I have done with different instruments. This is what I do when I read a paper and I try to do an experiment with my materials and instruments. In biology, there is variability (the obtained results are never exactly the same, but the procedure I use is the same). Reproducibility is a critical point not only in biology, but also in psychology and artificial intelligence field.

When, in the lab, I perform a HT analysis to be published, I have to put also the row data. There is no guarantee that somebody, taking my data, will obtain the same row data. Every time a different instrument is used, also the statistics of my final analysis will change. Same experiment in different places with different reagents and methodological approaches → results represent a biological phenomenon, even though they are opposed to each other.

Example: differentially expressed gene analysis → some of them will not be considered. Even if I try to take all the data with the recommended tools, the data that I will obtain will not be the same, because informatic tools constantly evolve. Sometimes, reproducing results from one paper is important, because a paper will never give me ALL the information (example: it will give me the p-value, but not the full list of differentially expressed genes).

Differential expression analyses are easier to be done than single-cell analyses. Goal: to provide a sufficiently high amount of reproducibility, so that I can be able to trust to what is published and to extract data in which I am really interested (normally, before starting a new experiment, it is important to chose data that can address my question).

Criticality of reproducibility

Researchers tried to publish a paper in which they thought: if I have a certain number of cell lines, on which I do a drug treatment analysis to calculate the IC50, I can identify sensitive and resistant cell lines. Given these 2 extremes and the untreated cell line, I can detect the resistance-related genes by differential expression analysis. Data were published, and Calogero's group tried to replicate the experiment: they took the same cell lines and performed a more evolved differential expression analysis (in comparison with the previous scientists, who performed a t-test); researchers published a heat map using 5-fluorouracil nucleotides. However, Calogero's group obtained different results related to the differentially expressed genes: what happened? Another group tried to understand it. They discovered that, using Excel, a researcher of the paper dis-aligned the data, so he was asked to retract the data. Scripting is better than "manually" manipulating data because errors can be tracked from the script.

To do list:

- Install a github repository: it is a place where things can be deposited to be seen also by other people.
- Create a repository: new → the name of the creator is mine → repository name: hello world (example). Next, there is a description of what is my repository doing: github is a mixture between a commercial and a non-commercial platform. Select "public" and "add a README file" (it will give me a small description of the content of the repository): during an analysis, when I do a script, this option will allow me to summarize the data that are present in the repository, otherwise I will not understand anything if the data present in there are confused. Once I created the repository, I can

modify the line of the test with the pen icon. At the bottom of the page, I can commit the changes I did in my repository.

- ls -lh
- cd = change directory.
- pwd = tells me where I am.
- If I want to go to the desktop, I write ls (so I can see everything that is inside my system), then cd Desktop → I move to the desktop.
- cd .. moves me of 2 levels.
- To make a new folder (example: prova), I have to use mkdir prova.
- I copy the path repository, present in github, in order to clone it locally on my computer. To clone it, I write git clone https://...
- Then, I type Desktop, and I see that there is a file called "hello-world". Github is a way to work with R in the Rstudio environment.

Exercise

Clone the file from github to the computer, modify it and save it. Once the local copy has been created, whatever modification is located on github only; the command "add" allows to move the modifications to the stage. "Commit" and "push". Commit allows me to prepare data to be uploaded.

Once I open my readme file and I modify it, on the terminal I write hello-world → ls → git add readme.md (if I have a single file) or git add * → git commit -m (I write a message to remind me what I am doing) "updating readme" → git push origin main (push allows me to push the updates. If I created branches, I have to indicate them. Main indicates that I do not work on the branches). Now, if I go back to github and I do an update, I will see my modifications.

SUMMARY:

- Clone = download a repository
- Add = to do some modifications of the data that have been applied
- Commit = to prepare the modifications to be delivered to the repository where I have to store them
- Push = to push the modifications.

RStudio

On the console, I write commands that are immediately run. Example: getwd() tells me where I am. The environment shows me the variables and the vectors I create in the console, while the bottom part shows me the different directories. This part is separated from the console, but if I set it as the working directory, then the path will coincide with the one present on the console. In the upper part, I create files that I can rename (example: command.R); in this case, commands are executed with "Run", and results are written on the console. On the right, I can open a project, go to the desktop and open hello world, which is considered as a project. If I select "git" at the top right of RStudio, I can export command.R into a new environment: I do "commit", I write a message (example: "creating a command.R to have the working directory) → "commit" and "push": the remote repository has been updated with the new modifications.

Markdown

It is an easy syntax that I can use in .md files. During the exam, we will be able to use all the possible online helps. Different # allow to generate chapters and sub-chapters, for example.

Docker

The docker is a “box” that can be filled by anything; the same container can transport different things. Even if I change the computer, data will always be contained in the docker, so I can analyse them even after a long time. When I have a Linux virtual machine, I have to dedicate a part of the RAM and of the disk to it. On the other hand, docker is installed on Windows, and it runs in background: the virtual machine will be run on Windows. The Linux environment is simulated on Windows, while the material (the RAM, the disk…) is used only if I run the container. The container becomes a piece of software that allows me to run my analysis that I describe on Github.

On Moodle, there are 2 small datasets.

- Create a github “data_analysis”
- Create a README.md on the remote desktop
- Create locally the data_analysis repository: I have to unzip the 2 datasets → 2 folders will be generated. They will have to be loaded on the remote repository.
- Install Docker desktop, R and RStudio, where I have to install the devtools library
- Install docker4seq (used for bulk RNA-seq) and rCASC (used for single cell RNA-seq).