BEST PRACTICES

Rene Verinaud Anguita Junior

renevajr@dsee.unicamp.br ReneJunior

OVERVIEW

While developing a research is possible that you may need to reproduce someones work, which is not always easy, you have to understand what and how the author did it. Sometimes is necessary to contact the author, the answer often takes a long time and doesn't solve all the questions. With this problem in mind, this document contains a brief documentation on what to do and don't in computer research, to make your work easy to reproduce.

One of the best practices is to create an organized repository, with folders defined for each theme. Thus, when is time to assemble all the elements described below, you will have a greater synergy between them, facilitating the construction of a reproducible document.

This material is divided into two columns, the one on the left is responsible the remarks on the topics Data, Code, Workflow, Documentation and Distribution. The right column contains observations and examples of some tools, along with a brief evaluation of mine about it, giving an score from 0 to 5, based on my experience using them and the ability to find solutions online.

DATA

Is often necessary in a research to be able to change a parameter in the raw data, studying a specific behavior while recording all those changes. This topic will comment on some possible mistakes and successes when working with the raw data of your system, using topics with Do and Don't.

Do and Don't

- Check the licenses required to use this data. They may have serious restrictions and/or ethical impacts;
- Use a repository with history to upload your raw system data;
- Avoid making changes to your systems manually, but if not possible, document any changes;
- Don't forget to indicate where the data came from, with a reference in your document.

CODE

Another very important, but often overlooked topic is the code. Usually, the final code can be found in work, with or without comments, but in most cases, simply publishing the code doesn't make it able to reproduce. In order to improve your work, below are some topics that are important.

Do and Don't

- It's essential to chose a repository where you have a history of changes made to the code;
- Save as many changes as possible and, also try to comment on them. Often authors forget details and they themselves are unable to reproduce their own works;
- If you use randomness in your work, don't forget to provide seed;
- Don't forget to reference all the other works that you used.

PROGRAMS

In this section, some options will be presented for each five subjects. At the end of each topic, there will be the tool used by me, with a grade.

CODE/DATA

Git

GitHub

Kaggle

Git Large File Storage

GitHub ●●●●●

This tool is widely used, it is possible to find all the solutions for possible errors, with many materials being found online.

WORFLOW

Reana

Sacred

Comet

Sacred ●●○○○

This tool in theory is great to organize the workflow, but in reality it has a lot of issues that I couldn't find the solutions online, I used it at the beginning of the work, but I'm no longer using it.

WORKFLOW

In particular, in my area, many authors are more concerned with explaining the methodology applied in the work, than explaining how the work itself was done. Therefore, I will present some topics when building a workflow, always thinking about making it easier for the reader to understand.

Do and Don't

- Try to think about how to explain your work in a VERY simplified way, using a flowchart or a block diagram. Often, a good image is as important than paragraphs of explanation;
- Demonstrate how the data is input in your work and try to section what is data, codes (if there is more than one type of language) and the results obtained, being able to use a color chart for this.

DOCUMENTATION

All the topics cited so far are important, but if your work does not have a good documentation, it will hardly be reproducible. In this way, I will comment some remarks you should keep in mind.

Do and Don't

- There are several types of tools for documentation, something interesting is to use, for example, one that can be viewed in your repository, as is the case of Jupyter Notebook and GitHub;
- Don't forget to explain the results, flowcharts, graphs, tables, etc. Presenting one of these elements without explanation, makes your job poor.

DISTRIBUTION

For any work to be able to be reproducible, it is necessary to choose a type of format to distribute the documents, codes and the results found. But how to do it is also relevant, some important points for this are described below.

Do and Don't

- Try to choose two ways to distribute your work experiments. Often, it requires so many installations and packages, that it becomes feasible to make your work unreproducible;
- Seek to publish as many versions of your code and documentation as possible. This will facilitate for the reader to understand any changes of the problem.
- Always inform in your documentation all the software used, and its versions. If you work results involves computational time, don't forget to declare all the specifications of the hardware as well.

DOCUMENTATION

- **Google Colab**
- **Jupyterlab**
- **Nextjournal**

Jupyterlab ●●●●○

It is possible to easily find forums with the resolution of several problems. A maximum score was not given, as there is no easy way to convert the document to a PDF format, needing to use another editor to prepare a more restricted document, such as an article.

DISTRIBUTION

- **VM Virtual Box**
- **Docker**

VM Virtual Box ●●●●○

With this tool, it is possible to create several virtual machines, with limitations to your physical hardware. Thus, being able to create one, with the operating system of your choice and with the programs, for reproducing your work, already installed. A higher score was not given, because each machine file is very heavy and many times the user does not have a physical PC powerful enough to use this tool.