Tools for reproducible research

Methods of Scientific Working (for Crop Sciences) (3502-440)

21 Jan 2025

Table of contents

1 Motivation

The scientific method offers several guarantees. These guarantees ensure that the research process is *transparent* and *independent of the researcher* conducting the study. Consequently, at least in theory, the research can be replicated by others.

From these philosophical considerations, one can derive a minimal standard for experiments (both in the laboratory and the field), subsequent data analysis, and other scientific computations (e.g., simulations). This standard is that they should be **reproducible**: the protocols, generated data, analysis methods, and analytical steps must be described and stored in such a way that another research team can recreate all the results (e.g., the figures in a paper). The significance of reproducibility is widely acknowledged today. However, it is not as widely practiced as it should be. This discrepancy largely arises because many empirical or computational scientists have not fully embraced the necessary tools for reproducible research.

One reason is that modern scientific analyses have grown more complex, given the rapid increase in data volumes (we are in the era of **big data**). Analytical methods are becoming more intricate and computation-intensive, with approaches like **machine and deep learning** being prime examples.

Integrating various methods for data cleaning, storage, analysis, and result presentation becomes essential in complex **data analysis workflows**. These workflows must merge various computer programs and software tools to establish transparent and reproducible processes that meet the contemporary standards of reproducible research.

In the following we will discuss general principles for reproducible research but will focus primarily on the use of relevant **tools**.

To keep things simple, we will use the Quarto publishing system (https://quarto.org), which is tightly integrated into the R ecosystem, as a means to have an electronical laboratory notebook that will allow the construction of simple data cleaning, data analysis and visualization workflows in a single document and produce nice reports or scientific publications.

We will also briefly mention, but not further describe, other tools for

- running **analysis pipelines** like classical tool Make, SnakeMake or Nextflow. The latter two are now standards in the field and required for larger analysis projects.
- **version control** and **project management** systems like **Git** with associated websites such as GitHub or GitLab[1]

The objective of the class is to re-emphasize the fact that modern empirical research has evolved into data science, necessitating a specific set of skills and tools. Familiarity with these is crucial for achieving reproducible research. It's also essential for cultivating a mindset that prioritizes reproducibility right from the outset of a research project. This ensures that all facets of laboratory, field-based, and computational research, along with a variety of outcomes like software, data analyses, papers, presentations, and posters, are as reproducible as possible.

2 Learning goals

To be added

3 Key principles of reproducible research

The level of reproducibility in a scientific study can vary. It can be defined by the following questions:

- Can the tables and figures be reproduced using the provided code and data?
- Is the analysis code performing as expected?
- Beyond understanding what was done, is the rationale, or why it was done, clear? For instance, how were parameter settings determined? Are all parameters documented?
- Is the code adaptable for different datasets, or can it be extended for other purposes? Does the analysis code comprise simple scripts or is it a well-documented standalone software package?

It is important to recognize that reproducibility is not a binary concept. While the ideal of wholly consistent and complete results is aspirational, it is challenging to realize in practice.

For achieving a high level of reproducible research, several fundamental principles must be adhered to:

[1]The teaching materials for this module are managed with GitLab.

4 Create a written record for everything you are doing

While graphical user interfaces (GUIs) may appear convenient for interacting with computers, using code (in the form of computer programs or scripts) is far more robust for generating, cleaning, and analyzing data. Thus, it is advisable to use code or scripts whenever possible. This is because GUIs typically do not record your actions, which means if you wish to replicate an analysis later, you'll need to manually note and re-enter the parameters.

5 Automate as much as you can

Often, research involves repetitive tasks, such as manually entering tabular data from a PDF file, converting names in a spreadsheet to a computer-readable format, or labeling images (e.g., to measure parameters like leaf area).

In many instances, analyses must be repeated or can be linked in a workflow based on an analysis protocol. Aim to **automate everything** as much as possible, either by using appropriate software or by crafting simple scripts that can handle large datasets and be executed repeatedly with minimal effort. For example, the R package `tabulizer` allows you to automatically extract tabular data from PDF-formatted files.

Certainly, there is a learning curve when it comes to writing computer code. However, investing time in this is more valuable than manually handling repetitive tasks. With the former, you gain new skills; with the latter, you merely spend your time on a dull task.

6 Develop computational workflows

Data generation and analysis does not only involve running programs, but involves the development of ideas, descriptions of designs, protocols, as well as intermediate data, dead ends, etc. Make sure to use a **workflow system to document** all of these parameters, including the different steps and dependencies between them, as well as the names and versions of programs.

7 Start from original, raw data

During the processing of data, there is a chance that some information may be lost, some data may be missing, or errors may be introduced during a data cleaning or analysis step. For this reason, try to obtain your **data in the most original form possible** (such as direct measurements from devices, field trial data, etc.) to quantify and account for measurement errors, missing data, systematic biases during measurements, and errors during data processing and transformation.

8 Record and store metadata

In addition to experimental data, there are often additional data associated with a research project. These may include information about who conducted an experiment, as well as when and where it was conducted, or, in the case of field trials, details about the location, year, conditions, and design of the experiment. Obtain and store as much data and metadata as possible for a project, because they can provide useful information (e.g., uncovering biases) in later analyses.

9 Record information about data origin and any transformations

It is quite common to receive datasets from other research groups or entities that you or others have cleaned and transformed. Organize your data in a way that allows you to **maintain information about the provenance of the data** files and any associated restrictions, such as legal limitations. A suitable data structure and thorough documentation can help you trace back to the original data source and understand the quality, limitations, or restrictions of the data.

10 Understand what you are doing

Whether dealing with GUI programs or computer code, you may be tempted to utilize analysis tools that you do not fully understand, resorting to copying and pasting from other sources, or you may allow the analyses to be done by other experts. Instead, strive to comprehend all the steps in your analyses yourself as much as possible in order to complete a project and try to **be self-sufficient**. This will require acquiring different skills in the design and conducting of experiments, as well as computational and statistical analysis, through appropriate training. This involves reading tutorials, documentation, publications, and books. Knowing as much as possible about all the methods of a project allows you to think critically about your work and contributes to conducting high-quality science. However, depending on the particular project, it may be a more effective strategy to focus on certain key methods and become an expert in those areas. For other methods, it may then be more advantageous to collaborate with experts in those fields.

11 Data management

The management of research data should follow certain rules that apply to its creation, storage, and subsequent formatting. All these steps should begin with the **raw data**, so you must ensure that you possess the raw data and understand their origins and how they were generated. You should also determine for yourself what constitutes the raw data, and how far back you need to go. For instance, do you require the direct output from a spectrometer, or is it sufficient to start with derived data that have been transformed and formatted from the original readout? In any case, you should make a conscious decision in the context of reproducibility. If you are collaborating with someone, ensure that you receive the data in the most original form possible.

Having raw data is advantageous for various reasons:

- If a mistake is made during subsequent transformation and formatting steps, you can hold yourself accountable and not the collaborator, thereby avoiding embarrassment.
- Data that have undergone minimal transformation and formatting steps are more likely to be correct.
- In the long run, you may save time because you may have faster access to raw data in case of problems with downstream analyses.
- Raw data may provide greater potential for extensions based on new ideas.
- The publication of raw data often has a higher impact because it enables other researchers to use them.

12 Guidelines for implementing reproducible research

The basic concepts for working reproducibly can be implemented with a few simple rules.

Although more time-consuming, you should aim to do things properly from the beginning. This involves writing a thorough **outline of your research** and documenting all subsequent steps in writing. If you write code or analysis scripts (using R, for example), strive to write clear code, document it well, and test it thoroughly to be sure that it contains no errors. Well-documented code is also easier to debug, maintain, and extend in the future.

For many programming languages, style guides exist that detail conventions for writing correct and elegant code. The style guide for R can be found at https://style.tidyverse.org/ and https://google.github.io/styleguide/Rguide.html. Tools are available that check whether your code adheres to the style of the language (references for R tools are provided in the two links above). After a learning phase, you will become more efficient, and your high-quality research and analysis tools will have a greater impact.

You might think that no one else will use your data or your code. However, it's likely that you will be the user in a few months and will be grateful to yourself for implementing these principles. Remember that you are your most immediate collaborator.

During a research project, your advisor or colleagues will pose questions:

- "How does the code you just sent us differ from the one we previously used?"
- "I need to know where this data file comes from and whether I am allowed to use it."
- "The reviewer has requested that we repeat the analysis but leave out subject X."
- "I am running your analysis script on my data, but I am encountering this error."

A general goal should be to minimize manual tasks. You should avoid:

- Opening a file to extract data as CSV. Instead, write a program that reads the data in its original file format, such as the `.xlsx` Microsoft Excel format.
- Opening a data file for even minor edits. Keep the raw data unaltered. If you must edit data, do so using a script or by creating an intermediate data file.
- Pasting results into the text of a manuscript. Instead, use the import function that allows results from analysis programs to be directly entered into a manuscript (yes, this is possible!). Any revised analysis will then automatically update in the paper.
- Manually copying, pasting, and editing tables. Format tables programmatically or directly import them from the output of a statistical program.
- Copying, pasting, and adjusting figures by hand. Create publication-ready figures using software and import them into the manuscript.

Your general approach should be to automate as much as possible with scripts and data analysis pipelines. If you do everything "by hand" once, you will have to repeat the same manual processes every time.

13 Basic tools for reproducible research

There are many software tools that allow you implement reproducible research, and it is easy to get lost or feel overwhelmed. For a start, you should focus on a few key tools, but if you continue for your PhD, you should also include other ones. In this module, we focus on R, RMarkdown and Quarto for report generation.

Here is a list of additional tools that you may wish to explore:

Automated data analysis and documenting dependencies can be achieved with `Make`, or more modern workflow languages like Nextflow (https://www.nextflow.io/) or Snakemake (https://snakemake.github.io/). Only a single command is sufficient to (re-) run complex analysis pipelines on your laptop or supercomputers.

The Unix or Linux **command line** for running programs with the bash shell is essential for doing serious data analysis. Many bioinformatics programms run only at the command line and can be easily combined in pipelines using the bash shell or workflow languages.

Markdown and LaTeX are useful **writing** of beautiful scientific texts and executable papers. Since they are based on text only, they can be much more easily integrated into version control systems than binary Word (`.docx`) or LibreOffice (`.odt`) formatted files.

The combination of code and text is called **Literate programming** (Wikipedia) and is now widely used in data science. Tools for literate programming are Quarto and RMarkdown, and other widely used tools are Jupyter notebooks (https://jupyter.org/) for R, Python and Julia programming languages, or Pluto notebooks (https://plutojl.org/) for Julia. A more recent tool for literate programming, which is also based on Markdown and Jupyter Notebooks is JupyterBook (https://jupyterbook.org)

Version control with `git`. While version control systems are not strictly necessary for reproducibility, they greatly facilitate the management of projects and collaboration. The first steps are easy to learn, and once you know how to use git, you most likely do not want to go back.

Well documented software libraries, such as R **packages** allow you to combine diverse analyses steps more easily.

Learning a full **programming language** like Python (https://www.python.org/) or Julia (https://julialang.org/) is highly valuable (and in some circumstances better than R) for manipulating data files or some serious programming. Many tasks that are awkward or difficult in R are easy to do in Python or Julia. Julia is also much faster than R or Python.

Container and virtual machine technologies: This is a more advanced tool, but often necessary for advanced analysis pipelines. Containers make analysis pipelines reproducible because they allow to "conserve" the exact version of programs you used for analysis that may become incompatible in later versions and therefore not prevent you from redoing the analysis again. In addition, containers and virtual machines allow to transfer analysis from a laptop, on which the analysis pipeline is developed to a high-performance computer (HPC).

These basic tools that are important for reproducible computational research. They all have in common that they are mostly open source tools. This diversity may look bewildering for a beginner, however, they can be learned with reasonable effort and get you a long way. A good strategy is to explore the different options, and decide for a few systems (in particular those that are used in your scientific environment) and learn them properly.

14 Do not repeat yourself (DRY)

- In code, in documentation, etc.
- Repeated bits of code are harder to maintain
  - Write a function
- Use documentation systems like Roxygen2
  - Documentation in just one place
- Make use of others' code

DRY is among the more important concepts/techniques.

15 Summary

- Reproducible research is a concept, which aims at implementing various principles to ensure the high quality of research and to support the scientific method.
- Reproducible research refers to the fact that all data and analysis scripts are structured in the manner such that others can use them and obtain the same results.
- Following the principles of reproducible research also makes your own research more structured and of higher quality.
- There are many concepts and tools available for reproducible research. We focus on R and RMarkdown because they are the most efficient tools (for beginners) as an electronic lab notebook and even for writing master theses and scientific publications.

16 Key concepts

To be added

17 Further reading

- Markowetz (2015) - A short introduction why you should work reproducibly for your own advantage.
- Sandve et al. (2013) - Simple rules for computational reproducible research
- Noble (2009) - A guide to organizing computational biology projects
- Alston and Rick (2021) - A beginners guide to reproducible research
- @&peng_reproducible_2021 - Reproducible research - A perspective
- CRAN Task View: Reproducible Research - A collection of R packages for reproducible research https://cran.r-project.org/web/views/ReproducibleResearch.html

18 Review questions

1. How does the concept of reproducible research connect to the scientific method?
2. What are the key definitions and concepts of reproducible research?
3. Why should the implementation of reproducible research lead to higher quality in research?
4. What are tools you know for reproducible research?
5. Why is RMarkdown a good tool for reproducible research?

19 Computer lab

- Quarto Notebook

References

Alston JM, Rick JA. 2021. A Beginner's Guide to Conducting Reproducible Research. *The Bulletin of the Ecological Society of America* **102**:e01801. doi:10.1002/bes2.1801

Markowetz F. 2015. Five selfish reasons to work reproducibly. *Genome Biology* **16**:274. doi:10.1186/s13059-015-0850-7

Noble WS. 2009. A Quick Guide to Organizing Computational Biology Projects. *PLoS Comput Biol* **5**:e1000424. doi:10.1371/journal.pcbi.1000424

Sandve GK, Nekrutenko A, Taylor J, Hovig E. 2013. Ten Simple Rules for Reproducible Computational Research. *PLoS Computational Biology* **9**:e1003285. doi:10.1371/journal.pcbi.1003285