Reproducibility

ACM Terms

- Repeatability
- Replicability
- Reproducibity

Repeatability

- Same team, same experimental setup
  - The measurement can be obtained with stated precision by the same team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same location on multiple trials. For computational experiments, this means that a researcher can reliably repeat her own computation.

Replicability

- Different team, same experimental setup
  - The measurement can be obtained with stated precision by a different team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same or a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using the author's own artifacts.

Reproducibility

- Different team, different experimental setup
  - The measurement can be obtained with stated precision by a different team, a different measuring system, in a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using artifacts which they develop completely independently.

Ten Simple Rules for Reproducible Computational Research

- Sandve, et. al. https://doi.org/10.1371/journal.pcbi.1003285

1. For Every Result, Keep Track of How It Was Produced
2. Avoid Manual Data Manipulation Steps
3. Archive the Exact Versions of All External Programs Used
4. Version Control All Custom Scripts
5. Record All Intermediate Results, When Possible in Standardized Formats
6. For Analyses That Include Randomness, Note Underlying Random Seeds
7. Always Store Raw Data behind Plots
8. Generate Hierarchical Analysis Output, Allowing Layers of Increasing Detail to Be Inspected
9. Connect Textual Statements to Underlying Results
10. Provide Public Access to Scripts, Runs, and Results

Rule 1: For Every Result, Keep Track of How It Was Produced

- Whenever a result may be of potential interest, keep track of how it was produced.
- If the step is performed by a computer program, the critical details include the name and version of the program, as well as the exact parameters and inputs that were used.
- Specify the full analysis workflow in a form that allows for direct execution, in the form of simple shell scripts or makefiles at the command line, or in the form of stored workflows in a workflow management system.
- As a minimum, record sufficient details on programs, parameters, and manual procedures to allow yourself, in a year or so, to approximately reproduce the results.

Rule 2: Avoid Manual Data Manipulation Steps

- Rely on the execution of programs instead of manual procedures to modify data. Such manual procedures are not only inefficient and error-prone, they are also difficult to reproduce.
- If manual operations cannot be avoided, you should as a minimum note down which data files were modified or moved, and for what purpose.

Rule 3: Archive the Exact Versions of All External Programs Used

- In order to exactly reproduce a given result, it may be necessary to use programs in the exact versions used originally.
- In some cases, store a single executable or source code file.
  - GitHUB
- Dependencies on other installed programs/packages, or dependencies to specific operating system components.
    - store a full virtual machine image of the operating system and program.
    - As a minimum, you should note the exact names and versions of the main programs you use.

Rule 4: Version Control All Custom Scripts

- When a continually developed piece of code (typically a small script) has been used to generate a certain result, only that exact state of the script may be able to produce that exact output, even given the same input data and parameters.
- Track evolution of code is to use a version control system such as Subversion, Git, or Mercurial.
- As a minimum, you should archive copies of your scripts from time to time, so that you keep a rough record of the various states the code has taken during development.

Rule 5: Record All Intermediate Results, When Possible in Standardized Formats

- Browsing through intermediate results can reveal discrepancies toward what is assumed, and can in this way uncover bugs or faulty interpretations that are not apparent in the final results.
  - When the full process is not readily executable, it allows parts of the process to be rerun.
  - it allows critical examination of the full process behind a result, without the need to have all executables operational.
- When possible, store intermediate results in standardized formats. As a minimum, archive any intermediate result files that are produced when running an analysis (as long as the required storage space is not prohibitive).

Rule 6: For Analyses That Include Randomness, Note Underlying Random Seeds

- For analyses that involve random numbers, the random seed should be recorded.
  - This allows results to be reproduced exactly by providing the same seed to the random number generator in future runs.
- As a minimum, you should note which analysis steps involve randomness, so that a certain level of discrepancy can be anticipated when reproducing the results.

Rule 7: Always Store Raw Data behind Plots

- Store raw data behind figures in a systematic manner, so as to allow raw data for a given figure to be easily retrieved, one can simply modify the plotting procedure, instead of having to redo the whole analysis.
- In cases where plotting involves more than a direct visualization of underlying numbers, it can be useful to store both the underlying data and the processed values that are directly visualized.
- As a minimum, one should note which data formed the basis of a given plot and how this data could be reconstructed.

Rule 8: Generate Hierarchical Analysis Output, Allowing Layers of Increasing Detail to Be Inspected

- The final often represent highly summarized data. In order to validate and fully understand the main result, it is often useful to inspect the detailed values underlying the summaries.
- When the storage context allows, incorporate permanent output of all underlying data when a main result is generated, using a systematic naming convention to allow the full data underlying a given summarized value to be easily found.
- As a minimum at least once generate, inspect, and validate the detailed values underlying the summaries.

Rule 9: Connect Textual Statements to Underlying Results

- Throughout a typical research project, a range of different analyses are tried and interpretation of the results made.
- If you want to reevaluate your previous interpretations connect a given textual statement (interpretation, claim, conclusion) to the precise results underlying the statement.
- Connect statements to underlying results already from the time the statements are initially formulated (for instance in notes or emails).
  - E.G. a simple file path to detailed results
  - Python or Jupiter notebook
  - the ID of a result in an analysis framework, included within the text itself.
- As a minimum, provide enough details along with your textual interpretations so as to allow the exact underlying results, or at least some related results, to be tracked down in the future.

Rule 10: Provide Public Access to Scripts, Runs, and Results

- All input data, scripts, versions, parameters, and intermediate results should be made publicly and easily accessible.
- As a minimum, you should submit the main data and source code as supplementary material, and be prepared to respond to any requests for further data or methodology details by peers.
- Making reproducibility of your work by peers a realistic possibility sends a strong signal of quality, trustworthiness, and transparency.