Best practices in bioinformatics research: open source software and reproducibility

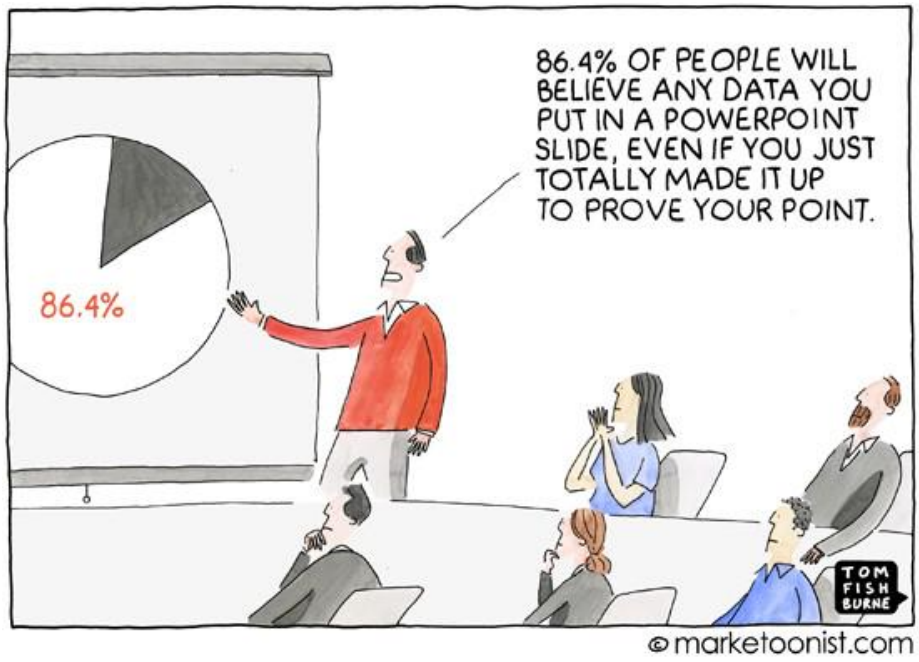

Figure 1: See, don't trust

Quick survey:

- What does *open* mean? What should be, or must be open? Is it that way?
- What is *reproducibility*? Why do we want reproducibility? Is it that way?

Next: Open Science

Open Science

> Who owns the research outputs? Who should have access to research outputs?

Science is supported by **public** funds. All outputs should thus arguably be made publicly and openly available - to the research community and the general public.

Open means free from any fees (*free as in beer*) and free from any (copyright, patents, ...) restrictions to re-use (*free as in speech*). This requires appropriate licences.

The figure below describes a simple but plausible research workflow, starting with a (successful) funding request and culminating in a scientific publication.

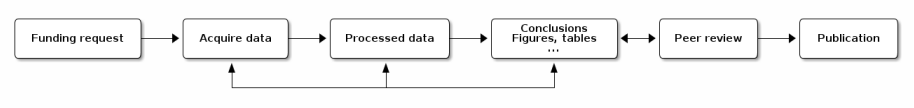

Figure 1: Research workflow

Open data

The **data**, both in raw and processed form, and its annotation (**metadata**) that accompanies a scientific paper (discovery) should be freely/publicly available and should be free to re-use.

Note that open data goes beyond the release of scientific data. Governments and city councils are publicly releasing data *to help people understand how government works and how policies are made.*

Open methodology

> An open methodology is simply one which has been described in sufficient detail to allow other researchers to repeat the work and apply it elsewhere (from Waton M., 2015)

Describe and release the process that lead raw data to processed data, and how the processed data has been further analysed to lead to results, figures and conclusions.

- **Open source**: the source code of your (scienfic) software should be openly released, so that others can read/understand/fix/re-use it. (see software licences)

- **Open protocols**: openly share the hole sample processing, data acquisition.

Open access

Papers and supplementary material should be available to **read** and **mine** by all, researchers and public.

There exist open access licences: CC-BY, CC-BY-SA, ...

Also

Open peer review, open education (Software and Data Carpentry, for example), ... open scholarship

Open Science

> Open science isn't a movement, it's just (good) science. It's also the future. (from Watson M, 2015)

- Moral argument
- Better science

Arguments against open Science:

- Too much unsorted information overwhelms scientists.
- Science will be used for bad things.
- The public will misunderstand science data.
- Increasing the scale of science will make verification of any discovery more difficult.

(from the Wikipedia Open Science page)

Some have also raised arguments about specific aspects of Open Science, often founded on confusions or mis-understandings:

- open source and the lack commercial prospects in software
- open access and quality of publication

See also

- Why Open Research? - Advance your career by sharing your work.
- The open research value proposition: How sharing can help researchers succeed
- The Open Knowledge Foundation
- University of Cambridge Research Data Management
- Open Access at the University of Cambridge (and here)
- If you are interested in *All things open* (science, education, ...), consider following/joining OpenConCam.

Prev: Introduction – Next: Reproducbile research

Reproducible research

> D Knuth: Literate programming is a methodology that combines a programming language with a documentation language, thereby making programs more robust, more portable, more easily maintained, and arguably more fun to write than programs that are written only in a high-level language.

> D. Donoho, as pointed out by John Claerbout: 'An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.'

> R. Gentleman and D. Temple Land, Statistical Analyses and Reproducible Research 2004. We introduce the concept of a compendium as both a container for the different elements that make up the document and its computations (i.e. text, code, data, ...), and as a means for distributing, managing and updating the collection.

Different levels of *reproducibility*

(There is some ambiguity when it comes to nomenclature. Here, I'll use the one that is most common in biological/bioinformatics, based on, among others V. Stodden and R. Peng - see references.)

Reproducibility/reproduce

> A study is reproducible if there is a specific set of computational functions/analyses (usually specified in terms of code) that exactly reproduce all of the numbers in a published paper from raw data.

Replication/replicate

> A study is only replicable if you perform the exact same experiment (at least) twice, collect data in the same way both times, perform the same data analysis, and arrive at the same conclusions.

A study might be reproducible (and this might not be an easy thing to achieve, in particular for large-scale studies), this does not make is replicable. The latter is stronger than the former. Replicability requires new samples and new data (*), which introduces new variability, and additional risks of errors. Reproducibility

is, to some extent, a technical challenge, while replication gives the results scientific validity.

(*) in particular biological replicates - see statistics lecture on experimental designs.

Re-usability Applying/adapt the software/methodology to a new but related question.

Other terms: mechanical reproducibility, statistical reproducibility, repeat/repeatability, ...

Why

- Reproducible/replicable science is better science. And it's even better if science is openly shared, and reproducible/replicable by others.

But also

- Five selfish reasons to work reproducibly

  1. Reproducibility helps to avoid disaster
  2. Reproducibility makes it easier to write papers
  3. Reproducibility helps reviewers see it your way
  4. Reproducibility enables continuity of your work
  5. Reproducibility helps to build your reputation

Tools

R and `Sweave/knitr`

1. Input: A text document with text and code chunks (that computer, generates tables and figures). Example here.
2. Extract the code chunks and execute the code.
3. Replace the code chunks by their respective outputs.
4. Compile the text document into a final format, such as pdf or html.

In R, one would use `Sweave` or the `knitr` package.

- Starting from markdown and R code (Rmarkdown)

Figure 1: Rmd to pdf/html

Figure 2: Rnw to pdf

- Starting from LaTeX and R code

Example:

```
library("knitr")
knit("report.Rmd") ## produces report.md
library("rmarkdown")
render("report.md", ouput_format = "pdf_document") ## produces report.pdf
render("report.md", ouput_format = "html_document") ## produces report.html
```

or, more directly

```
render("report.Rmd", ouput_format = "pdf") ## produces report.pdf
```

(You will get such an example in the statistics practical and will use RStudio.)

Jupyter notebook

Previously known as IPython notebooks: python, R, Julia, and many more. Interactive, in browser.

orgmode

`org-mode`: specific to emacs, many languages.

Automation using `make`

- `make` is an automatated build system, designed to avoid costly recomputation.
- `make` examines a `Makefile`, which contains a set of rules describing dependencies among files.
- A rule is run if the *target* is older than any of its *dependencies*.
- Older: compare creation time of files.

```
report.md: report.Rmd
    Rscript -e "knitr::knit('report.Rmd')"

report.pdf: report.md
    Rscript -e "rmarkdown::render('report.md', output_format = 'pdf_document')"

report.html: report.md
    Rscript -e "rmarkdown::render('report.md', output_format = 'html_document')"
```

Version/track your research

- Source code versioning systems such as git, subversion, hg, ... and their web interfaces (GitHub, Bitbucket) are invaluable to (1) track change over time, (2) save all intermediate states of your work and

(3) enable/facilitate collaborative work.

Reproducibility and the conduct of research

(as pdf)

Figure taken from the report of the symposium, 'Reproducibility and reliability of biomedical research', organised by the Academy of Medical Sciences, BBSRC, MRC and Wellcome Trust in April 2015. The full report is available from http://www.acmedsci.ac.uk/researchreproducibility.

Best practice

The tools described above are fantastic way to make ones computational research reproducible. But it generally not enough. As computational researchers, or even wet lab scientists that reply on some scripting or programming, it is important to follow some best practice to make our work more efficient, more tractable, this making is easier to reproduce.

Wilson G *et al.* Best Practices for Scientific Computing (2014).

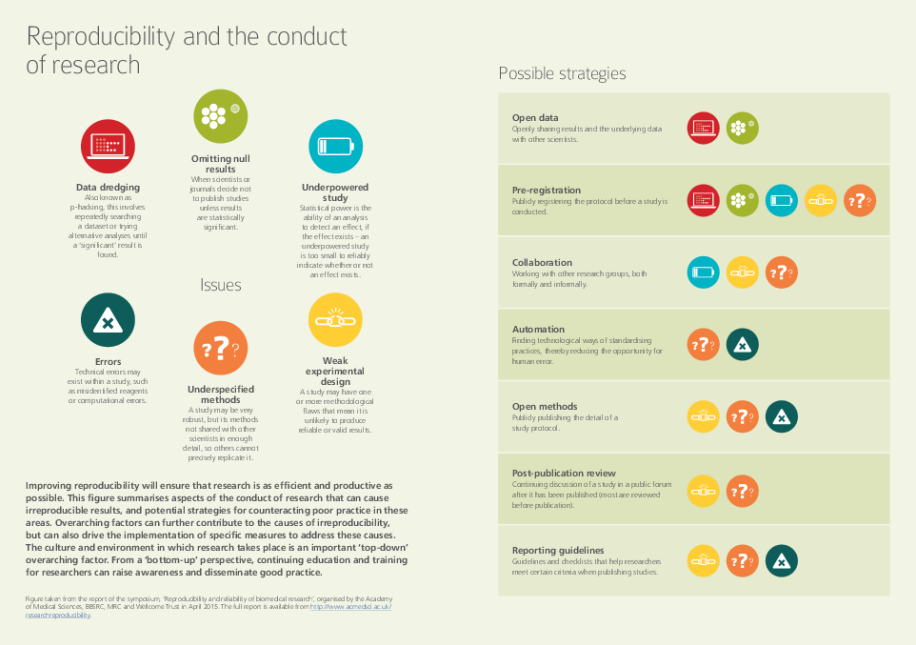

Figure 3: Reproducibility issues and possible strategies

1. Write programs for people, not computers.
2. Let the computer do the work.
3. Make incremental changes.
4. Don't repeat yourself (or others).
5. Plan for mistakes.
6. Optimize software only after it works correctly.
7. Document design and purpose, not mechanics.
8. Collaborate.

Prev: Open Science – Next: Conclusions

Conclusions

- We want to be open and transparency.
- Make sure we assure traceability of what we do.
- Make sure we and others can reproduce/replicate our work.
- Do it for you, do it for science.
- When will *open science* become simply *science*?
- This can not be an afterthought; will never work that way. **Plan** for openness and reproducibility. It takes **discipline**, but very reasonable effort (given adequate tools) for substantial benerits.
- Make your data and code **suistainable** to have a chance to make the research reproducible/replicable.

  > A piece of software is being sustained if people are using it, fixing it, and improving it rather than replacing it (ref).

See the Software Sustainability Institute for relevant activities and community.

Prev: Reproducible research – Next: References

References

- Gentleman, Robert and Temple Lang, Duncan, "Statistical Analyses and Reproducible Research" (May 2004). Bioconductor Project Working Papers. Working Paper 2.
- Wilson G, Aruliah DA, Brown CT, Chue Hong NP, Davis M, Guy RT, et al. (2014) Best Practices for Scientific Computing. PLoS Biol 12(1): e1001745. doi:10.1371/journal.pbio.1001745.
- Yihui Xie, Dynamic Documents with R and knitr, Second Edition (Chapman & Hall/CRC The R Series) 2nd Edition, 2015.
- Victoria Stodden, Friedrich Leisch, Roger D. Peng, Implementing Reproducible Research (Chapman & Hall/CRC: The R Series), 2014.
- Florian Markowetz, Five selfish reasons to work reproducibly. Genome Biology 2015 16:274 DOI:10.1186/s13059-015-0850-7.
- Peng, R. D. (2011). Reproducible Research in Computational Science. Science (New York, N.y.), 334(6060), 1226–1227.
- Leek JT, Peng RD. Opinion: Reproducible research can still be wrong: adopting a prevention approach. Proc Natl Acad Sci USA. 2015 Feb 10;112(6):1645-6. doi:10.1073/pnas.1421412111.
- Donoho DL. An invitation to reproducible computational research. Biostatistics. 2010 Jul;11(3):385-8. doi:10.1093/biostatistics/kxq028.
- Watson M. When will 'open science' become simply 'science'? Genome Biol. 2015 May 19;16:101. doi:10.1186/s13059-015-0669-2.
- Replication, psychology, and big science, simplystats blog, 18 April 2015

Prev: Conclusions