Is reproducible science achievable?

Kaur Alasoo

Replicability *versus* reproducibility

Replicability - obtaining the same results using using independent investigators, methods, data, equipment, and protocols.

Reproducibility - ability to rerun the same computational steps on the same data

“We [...] focus on the **ability to rerun the same computational steps on the same data** the original authors used as a minimum dissemination standard, which includes workflow information that explains what raw data and intermediate results are input to which computations.”

-- Stodden *et al*, “Enhancing reproducibility for computational methods”

Levels of reproducibility

Levels of reproducibility

- Release original data in an open repository
- Sufficient description of data analysis methods and choices (in English)
- Release source code for inspection
- Link analysis code directly to reported results (i.e. RMarkdown, sweave, iPython notebooks)
- Enable others to rerun analysis from raw data to results

Reproducibility is good, because:

Reproducibility is good, because:

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

From Florian Markowetz - “Five selfish reasons to work reproducibly.” (https://genomebiology.biomedcentral.com/articles/10.1186/s13059-015-0850-7)

But complete reproducibility can be a lot of work for little reward

“I think the uses for our data outside our lab are relatively limited beyond the scientific conclusions we have made.”

Arjun Raj - From reproducibility to over-reproducibility (http://rajlaboratory.blogspot.com.ee/2016/02/from-reproducibility-to-over.html)

Arjun Raj - Another approach to having data available, standardized and accessible: who cares? (http://rajlaboratory.blogspot.com.ee/2014/08/another-approach-to-having-data.html)

My code is ugly. I don’t want to support it.

Data analysis is much more like prototyping than software engineering

Not all data are easily available for re-analysis

1. Legal restrictions (company data, restricted data from human subjects, etc).
2. Practical restrictions (data too big to download and process in reasonable amount of time).

Software breaks over time. It does.

“My conclusion is that, on a decadal time scale, we cannot rely on software to run repeatably.”

Two solutions:

1. Run everything all the time (i.e. continuous integration, “continuous analysis”)
2. Admit that repeatability has a short half life, focus on “inspectability”.

C. Titus Brown - How I learned to stop worrying and love the coming archivability crisis in scientific software (http://ivory.idyll.org/blog/2017-pof-software-archivability.html)

Docker, Virtual Machines, etc can help, but ultimately

“The issue of whether I can *use* your algorithm is largely orthogonal to the issue of whether I can *understand* your algorithm. The former is engineering progress; the latter is scientific progress.”

C. Titus Brown - “The post-apocalyptic world of binary containers” (http://ivory.idyll.org/blog/2014-containers.html)

Moby/Docker in Production: A History of Failure
https://thehftguy.com/2016/11/01/docker-in-production-an-history-of-failure/

Reproducibility does not remove the need to trust the researcher

Simple steps to reproducibility

- Release your data! Use Zenodo, Figshare, etc or other domain-specific repositories.
- Use workflow engines that make it easier to rerun analyses, share code and make it inspectable (Snakemake, …).
- Link data analysis to reported results (RMarkdown, RStudio notebooks, iPython notebooks).
- Realise that 100% reproducibility is not always achievable without excessive amount of work and that’s ok.