Barbagroup reproducibility syllabus :: Lorena A. Barba Group

Barbagroup reproducibility syllabus

Lockheed P-80A airplane (1946). Credit: NASA Commons. —A reminder to test your code.

Posted on 10.31.2016

Also published on the Medium publication "Hacker Noon."

After my short piece, "A hard road to reproducibility," appeared in *Science*, I received several emails and Twitter mentions asking for more specific tips—both about tools and documents we use in the group to train the team about reproducibility.

> In answer to popular demand, then, I have collected here what we could call the **"Barba-group Reproducibility Syllabus."**

Top-10 Readings in Reproducibility

Early this year, my student Olivier and I were getting started writing a book chapter and later a full-length journal article; the first was about our reproducible-research workflow and the second on our CFD replication study. These represented about three years of work, not exclusively on this project, but taking most of the graduate student's time. As part of our "pre-writing" tasks, we decided to build—collectively as a group—our list of Top 10 papers discussing reproducible research in computational science. Here's our current reading list (modified from our first version of Feb. 2016):

1. Schwab, M., Karrenbach, N., Claerbout, J. (2000) Making scientific computations reproducible, *Comp. Sci. Eng*. 2(6):61–67, doi: 10.1109/5992.881708
2. Donoho, D. et al. (2009), Reproducible research in computational harmonic analysis, *Comp. Sci. Eng.* 11(1):8–18, doi: 10.1109/MCSE.2009.15
3. Reproducible Research, by the Yale Law School Roundtable on Data and Code Sharing, *Comp. Sci. Eng.* 12(5): 8–13 (Sept.-Oct. 2010), doi:10.1109/mcse.2010.113
4. Peng, R. D. (2011), Reproducible research in computational science, *Science* 334(6060): 1226–1227, doi: 10.1126/science.1213847
5. Diethelm, Kai (2012) The limits of reproducibility in numerical simulation, *Comp. Sci. Eng.* 14(1): 64-72, doi: 10.1109/MCSE.2011.21
6. Setting the default to reproducible (2013), ICERM report of the Workshop on Reproducibility in Computational and Experimental Mathematics (Providence, Dec. 10-14, 2012), Stodden et al. (eds.), https://icerm.brown.edu/tw12-5-rcem/ // report PDF
7. Sandve, G. K. et al. (2013), Ten simple rules for reproducible computational research, *PLOS Comp. Bio.* (editorial), Vol. 9(10):1–4, doi: 10.1371/journal.pcbi.1003285
8. Leek, J. and Peng, R (2015), Opinion: Reproducible research can still be wrong: Adopting a prevention approach, *PNAS* 112(6):1645–1646, doi: 10.1073/pnas.1421412111
9. M. Liberman, "Replicability vs. reproducibility — or is it the other way around?," Oct. 2015, http://languagelog.ldc.upenn.edu/nll/?p=21956
10. Goodman, S. N., Fanelli, D., & Ioannidis, J. P. (2016). What does research reproducibility mean? *Science Translational Medicine* 8(341), 341ps12-341ps12, doi: 10.1126/scitranslmed.aaf5027

Schwab et al. (2000) report on the pioneering example of reproducible research in the Claerbout lab (Exploration Geophysics, Stanford University). The first public communication of this group's approach that we could find goes back to 1992 [1]. That paper describes tools and processes in more detail, but for the same reason it is quite dated. So, we start with the summary account in *CiSE*. The Claerbout group developed an automatic build system for their published papers, including all the analyses and figures plus the typeset document. They used GNU `make`, certain standardized commands (burn, build, view, clean), and a notion of the file set or research compendium associated with the paper (data sets, programs, scripts, parameter files, makefiles). They report having used the system to-date for 14 papers involving 15 authors and hundreds of files. It's remarkable to read about their careful methods for reproducible documents, given that more than two decades later we're still struggling to adopt similar standards more widely.

Jump to Donoho et al. (2009). This could be the first group to explicitly associate reproducible research with open code and data:

> Reproducible computational research, in which all details of computations—code and data—are made conveniently available to others, is a necessary response to [the credibility] crisis.

Donoho et al. admonish that computation cannot claim to be the third branch of science because most computational results cannot be verified. In the two traditional branches, standards of practice already exist for managing the ubiquity of error: deductive science uses formal logic and the mathematical proof, while empirical science uses statistical hypothesis testing and detailed methods reporting. *“Many users of scientific computing aren’t even trying to follow a systematic, rigorous discipline that would in principle allow others to verify the claims they make.”* Ouch!

Donoho et al. cite strong influences from Claerbout’s methods, and lament that these are still not widely practiced. This paper also repeats the classic paraphrase of Claerbout: *“an article about computational science … is not the scholarship itself, it’s merely scholarship advertisement. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.”* (First appearing in a 1995 paper from this group [2].) My favorite quote from Donoho et al. (2009) is: *“… if everyone on a research team knows that everything they do is going to someday be published for reproducibility, they’ll behave differently from day one.”* The middle sections of the paper describe the various computational libraries developed to date in the Donoho group; those sections can be skimmed according to the reader’s interest. Towards the end, an interesting passage—written in the format of a Q&A—addresses the typical objections of researchers to working reproducibly. Many such objections are still hot topics today: it takes time and effort, we get no credit for it, competition, and so on. Notably, the final hypothetical objection is that “true reproducibility” should mean starting from scratch to re-create the results (rather than from the author-provided code and data). The rebuttal: *“…it proves nothing if your implementation fails to give my results because we won’t know why it fails. The only way we’d ever get to the bottom of such a discrepancy is if we both worked reproducibly…*"

The jointly authored paper of the Yale Law Roundtable participants (*CiSE*, 2010) expanded on the theme of transparency via open code and data. They defined reproducible computational research unambiguously as that making available all details (code and data) of the computations. Their additional recommendations include: assigning a unique identifier to every version of the data and code, describing within each publication the computing environment used, using open licenses and non-proprietary formats, and publishing under open-access conditions (or posting pre-prints). The rationale behind linking open access with reproducibility was absent, and some have criticized this aspect of the Roundtable recommendations. It may have grown out of the idea in Donoho et al. (2009) that “reproducibility means publication over the Internet,” and that authors should maintain a Web presence to facilitate discovery and access to their research. The connection between reproducible research and open-access publishing is, however, questionable. On the other hand, open code and data are valid components of reproducible computational research. Among future goals, the Yale Roundtable recognized the importance of enabling citation of code and data, of developing tools to facilitate versioning, testing and tracking, and of standardizing various aspects like terminology, ownership, policy.

Peng (2011) introduced the idea of a reproducibility spectrum. He says that reproducible research is a *“minimum standard for judging scientific claims when full independent replication of a study is not possible.”* Here we find an explicit distinction in terminology—something that continues to muddle the field—where full replication of a study involves collecting new data, with a different method (and code), and arriving at the same or equivalent final findings. (The distinction previously appeared in [3]) Peng mentions the Sloan Digital Sky Survey as an example of a project that would require formidable resources to fully replicate, and therefore proposes that reproducibility is a lesser standard that is more attainable. Other domains exist where full replication is unrealistic or extremely expensive. Reproducibility, says Peng, *“falls short of full replication because the same data are analyzed again.”* Nevertheless, it is a desirable minimum standard to assess the quality of the scientific claims. It requires that *“the data and the computer code used to analyze the data be made available to others.”* But, Peng laments, *“the biggest barrier to reproducible research is the lack of a deeply ingrained culture that simply requires reproducibility for all scientific claims.”*

Number 5 on our list (ordered chronologically) shifts to a different concern: numerical reproducibility in computations that involve parallel processing. In the discussion up until now, the concept of reproducible research assumed that running the same code twice with identical input will produce the same output. If the computation is done in serial, this assumption is good; but with parallel computing, it is not always the case. Diethelm (2012) ran an experiment using an application of finite-element analysis in computational mechanics. Executing the same simulation (same code, same input data) with varying number of processors gave different results! Investigating the differences and the source code pinpointed the cause of non-deterministic behavior: a direct solver for sparse linear systems (an external library). Diethelm goes through an example that illustrates how this can happen: a vector dot-product, computed in parallel over several partial sums. On each execution, individual processors may complete their portion of the sum in different order. In finite precision, addition is not associative and the final sum depends on the order of the partial sums. Under these conditions, ensuring numerical reproducibility involves introducing artificial synchronization points in the program, at the cost of additional run time. More elaborate techniques are available, but the conclusion is that in high-performance computing *“lack of reproducibility is typically a price that must be paid for speeding up the algorithm.”*

The ICERM Workshop Report (2013) builds on the contributions of the Yale Roundtable by placing particular focus on: (1) changing the culture and reward structure; (2) role of funding agencies, journals and employers; (3) teaching the skills for reproducible research. The required culture change includes valuing openness and transparency. But the academic reward structure sets critical barriers: *“The current system, which places a great deal of emphasis on the number of journal publications and virtually none on reproducibility … penalizes authors who spend extra time on a publication.”* Moreover, software development and data management are not valued scientific activities. The report addresses several ways to introduce incentives, requiring leadership from funders, journals and employers. Many of those discussions continue to this day in different venues (workshops, journal editorials, blogs, etc.).

I should add that I participated in the ICERM Workshop, giving a short talk titled

"Reproducibility PI Manifesto." The slides of this talk have been widely shared and commented [4].

Sandve et al. (2013) give us ten concrete actions we can take to make our research reproducible:

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 standard 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

Some common threads run through most of these recommendations. First, recognizing that a final result is the product of a sequence of intermediate steps (the analysis workflow), a key device for reproducibility is automation. Second, the central technology for dealing with software as a living, changing thing, is version control. And finally, archive and document everything with the best tools at hand. The one, inescapable corollary for the purposes of training researchers is that command-line skills are essential.

Item 8 of our reading list (Leek and Peng, 2015) expands on the purpose of reproducible research: to protect the integrity of science and build the public's trust on scientific results. Although a reproducible study can still suffer from poor study design, missing data, or confounding factors, reproducibility increases the rate at which we can uncover these flaws. Even so, the key is prevention via the training of more people on techniques for data analysis. Leek and Peng contribute to this goal via their massive online courses, and they also recognize the value of crowd-sourced workshops like Software Carpentry and Data Carpentry.

Next on the list is an essay by Mark Liberman, Christopher H. Browne Distinguished Professor of Linguistics at the University of Pennsylvania. He teaches introductory linguistics, as well as big data in linguistics, and computational analysis and modeling of biological signals and systems (among other topics). The subject of his essay is the big confusion of terminology that has spread on the reproducibility literature. He traces the confusion to a machine-learning workshop contribution, where the terms reproducible and replicable are swapped completely, compared to previous papers. Liberman concludes: *"Since the technical term 'reproducible research' has been in use since 1990, and the technical distinction between reproducible and replicable at least since 2006, we should reject [the] attempt to re-coin technical terms reproducible and replicable in senses that assign the terms to concepts nearly opposite to those used in the definitions by Claerbout, Peng and others."*

Our final item on the reading list is from earlier this year. Goodman et al. (2016) note that the various terms used in the field (e.g., reproducible vs. replicable) are not standardized. The importance of corroborating a previous study’s results is widely recognized. But, the authors note, *“ … the modern use of ‘reproducible research’ was originally applied not to corroboration, but to transparency, with application in the computational sciences. Computer scientist* [mistake: geophysicist] *Jon Claerbout coined the term and associated it with a software platform and set of procedures that permit the reader of a paper to see the entire processing trail from the raw data and code to figures and tables. …This concept has been* [used in] *epidemiology, computational biology, economics and clinical trials…”* [references provided]. Goodman et al. uphold the Claerbout/Donoho/Peng terminology, but propose a new lexicon as a way out of the confusion reigning the literature: methods reproducibility (original meaning of reproducibility), results reproducibility (previously called replication), and inferential reproducibility. Who knows if this new lexicon will stick, but what I like of this paper is its skillful discussion of differences among scientific domains that affect how each addresses reproducibility. In computational research, we’re used to a degree of determinism, for example, so methods reproducibility and results reproducibility are linked. Other fields have to deal with major stochastic variability. For most computational scientists, the second half of this paper will be alien, because it focuses on issues of statistical significance testing, clinical and pre-clinical research, and so on. It is good, however, to get a glimpse into this other world of science, where p-hacking and HARKing (hypothesis after results are known) are a thing.

Additional References

[1] Claerbout, Jon and Martin Karrenbach (1992). Electronic documents give reproducible research a new meaning, Proc. 62nd Ann. Int. Meeting of the Soc. of Exploration Geophysics, pp. 601-604, doi: 10.1190/1.1822162 http://library.seg.org/doi/abs/10.1190/1.1822162

[2] Buckheit, J. B. and D. Donoho (1995), WaveLab and reproducible research, In Wavelets and Statistics, edited by A. Antoniadis and G. Oppenheim, Lecture Notes in Statistics 103: pp. 55–81, doi: 10.1007/978-1-4612-2544-7_5

[3] Roger D. Peng, Francesca Dominici and Scott L. Zeger (2006), Reproducible epidemiologic research, Am. J. Epidemiol. 163 (9): 783-789. doi: 10.1093/aje/kwj093, http://aje.oxfordjournals.org/content/163/9/783.short

[4] Barba, Lorena A. (2012): Reproducibility PI Manifesto. figshare, doi: 10.6084/m9.figshare.104539.v1