Beyond computational reproducibility, let us aim for reusability

Gaël Varoquaux

Beyond computational reproducibility, let us aim for reusability

Gaël Varoquaux*

Science is based on the ability to falsify claims. Thus, reproduction or replication of published results is central to the progress of science. Researchers failing to reproduce a result will raise questions: Are these investigators not skilled enough? Did they misunderstand the original scientific endeavor? Or is the scientific claim unfounded? For this reason, the quality of the methods description in a research paper is crucial. Beyond papers, computers –central to science in our digital era– bring the hope of automating reproduction. Indeed, computers excel at doing the same thing several times.

However, there are many challenges to computational reproducibility. To begin with, computers enable reproducibility only if all steps of a scientific study are automated. In this sense, interactive environments –productivity-boosters for many– are detrimental unless they enable easy recording and replay of the actions performed. Similarly, as a computational-science study progresses, it is crucial to keep track of changes to the corresponding data and scripts. With a software-engineering perspective, version control is the solution. It should be in the curriculum of today's scientists. But it does not suffice. Automating a computational study is difficult. This is because it comes with a large maintenance burden: operations change rapidly, straining limited resources –processing power and storage. Saving intermediate results helps. As does devising light experiments that are easier to automate. These are crucial to the progress of science, as laboratory classes or thought experiments in physics. A software engineer would relate them to unit tests, elementary operations checked repeatedly to ensure the quality of a program.

Once a study is automated and published, ensuring reproducibility should be easy; just a matter of archiving the computer used, preferably in a thermally-regulated nuclear-proof vault. Maybe, dear reader, the scientist in you frowns at this solution. Indeed, studies should also be reproduced by new investigators. Hardware and software variations then get in the way. Portability, *ie* achieving identical results across platforms, is well-known by the software industry as being a difficult problem. It faces great hurdles due to incompatibilities in compilers, libraries, or operating systems. Beyond these issues, portability also faces numerical and statistical stability issues in scientific computing. Hiding instability problems with heavy restrictions on the environment is like rearranging deck chairs on the Titanic. While enough freezing will recover reproducibility, unstable operations cast doubt upon scientific conclusions they might lead to. Computational reproducibility is more than a software engineering challenge; it must build upon solid numerical and statistical methods.

Reproducibility is not enough. It is only a means to an end, scientific progress. Setting in stone a numerical pipeline that produces a figure is of little use to scientific thinking if it is a black box. Researchers need to understand the corresponding set of operations to relate them to modeling assumptions. New scientific discoveries will arise from varying those assumptions, or applying the methodology to new questions or new data. Future studies build upon past studies, standing on the shoulders of giants, as Isaac Newton famously wrote.

*Inria Saclay Ile-de-France, Parietal team, Palaiseau, France

Originaly published: newsletter of the IEEE Technical Committee on Cognitive and Developmental Systems, Vol. 13 (2016)

In this process, published results need to be modified and adapted, not only reproduced. Enabling reuse is an important goal.

To a software architect, a reusable computational experiment may sound like a library. Software libraries are not only a good analogy, but also an essential tool. The demanding process of designing a good library involves isolating elementary steps, ensuring their quality, and documenting them. It is akin to the editorial work needed to assemble a textbook from the research literature.

Science should value libraries made of code, and not only bookshelves. But they are expensive to develop, and even more so to maintain. Where to set the cursor? It is clear that in physics not every experimental setup can be stored for later reuse. Costs are less tangible with computational science; but they should not be underestimated. In addition, the race to publish creates legions of studies. As an example, Google scholar lists 28 000 publications concerning compressive sensing in 2015. Arguably many are incremental and research could do with less publications. Yet the very nature of research is to explore new ideas, not all of which are to stay.

Computational research will best create scientific progress by identifying and consolidating the major results. It is a difficult but important task. These studies should be made reusable. Limited resources imply that the remainder will suffer from "code rot", with results becoming harder and harder to reproduce as their software environment becomes obsolete. Libraries, curated and maintained, are the building blocks that can enable progress.

Originaly published: newsletter of the IEEE Technical Committee on Cognitive and Developmental Systems, Vol. 13 (2016)