Provisioning Reproducible Computational Science[*]

Victoria Stodden
Department of Statistics
Columbia University

Sheila Miguez
Department of Statistics
Columbia University

April 5, 2014

In this thought piece we are interested in the problem of "computational reproducibility," or of replicating the computational steps carried out in producing scientific findings [1]. This definition stands in contraposition to an independent replication of a research process (including writing a new software implementation), or a de novo re-collection of data processed with the same computational tools given the same commands. Starting from the exact same input data, can another researcher run the same commands and produce the same output? This question raises three immediate concerns in our minds:

1) what do we need to capture from the original computational environment?

2) how can the information from 1) be communicated to others and be both functional and persistent over time?

3) what are the sources of errors or uncertainty, even given the same data, code, and environment?

We offer some thoughts on each of these in turn.

[*Exercise for the reader: Is it possible to usefully reduce or simplify the notion of reproducibility further?*]

1. What do we need to capture from the original computational environment?

In an ideal world we would have bit-level identical inputs in the new environment including datasets and parameters, and the original software function calls returning the same results to a precision level determined in advance. All of these

[*] The authors gratefully acknowledge support from Alfred P. Sloan Foundation award number PG004545 "Facilitating Transparency in Scientific Publishing."

digital objects, including the original research article, need meta-data that associates them together as a research unit (sometimes called a *research compendium* [2]).

We also note that each of the objects in the research compendium should be citable and cited if used in subsequent research. This includes software, novel datasets, or ideas that originate in the associated research publication [3].

It is not enough to merely provide code and data. One must also provide the necessary documentation and materials that will allow others to stand up the research environment and understand the stack. We claim if something is undocumented it is not reproducible. We more strongly claim that if something is not automated it is not reproducible in the sense we described above.

2. How can the information from 1) be communicated to others and be both functional and persistent over time?

Researchers typically use code that executes in an environment that has evolved slowly over time. One my researcher may have code that works on their machine, but which is useless for another researcher, due to missing dependencies. The original researcher may not know how to reproduce the original setup of their machine.. Reproducing research then becomes a problem of reproducing an environment before the actual code can be used that underlies the research. Gone are the days of a standardized or uniform platform on a mainframe, used by the vast majority of the computational research community. We are now beset with a proliferation of different architectures and customized programming environments, making the execution of code developed in one environment next to impossible in another.

We desire a platform intended to be a step towards allowing researchers to share their code and data in reproducible environments. We suggest providing a small selection of carefully chosen pre-built environments along with the recipes to build them [3]. We seek a platform that will provide researchers the benefits of automated orchestration of environments without requiring researchers to become experts in “dev ops.” A platform that meets this goal should be machine agnostic so that it stays out of the researcher’s way and doesn’t interfere with a preferred workflow. A platform that meets this goal should also be designed as a service that can be chained in with other reproducible research tools that work with researcher’s code and data (for examples we have been working on in the small scale context see http://labs.researchcompendia.org).

We believe this is one place where XSEDE can facilitate reproducibility. The XSEDE infrastructure could provide a service interface to such reproducible research Platform-as-a-Service (PaaS) offerings. XSEDE can also be an important part of reducing the number of proliferated computational environments since it can create a more uniform user experience, regardless of the underlying computational technology or grid being accessed.

3. What are the sources of errors or uncertainty, even given the same data, code, and environment?

A containerized approach as we have described can protect against some forms of error, for example by including dependences in the run time environment, but not all. Core failures, numerical instability, and other aspects of the computational architecture can continue to affect results. We propose collecting meta-data on results to check for accuracy, or precision in replication, that could include descriptions of system architecture, dependencies, and system use during run time.

We also propose including these aspects of running replicated computational experiments in the concept of uncertainty quantification, as they can provide additional sources of error in results that can be benchmarked against the original outputs.

In this thought piece we consider a framing for the implementation of reproducible computational science in common computing environments. We describe several tools and technological affordances with which we hope to spur this very important discussion.

References

[1] Stodden, V. "Resolving Irreproducibility in Empirical and Computational Research," IMS Bulletin, 2013.

[2] Gentleman, R. and Temple Lang, D. "Statistical Analyses and Reproducible Research," Bioconductor Project Working Papers, 2004.

[3] Stodden V. and S. Miguez, Best Practices For Computational Science: Software Infrastructure and Environments For Reproducible and Extensible Research, WSSSPE Conference Proceedings, Journal of Open Research Software, forthcoming 2014.