A System for Automating Reproducibility in Science

Tom Crick[1], Benjamin A. Hall[2] and Samin Ishtiaq[3]

[1]Department of Computing & Information Systems, Cardiff Metropolitan University
[2]MRC Cancer Unit, University of Cambridge
[3]Microsoft Research Cambridge
[1]tcrick@cardiffmet.ac.uk

Summary

Reproducibility (replication, repeatability) is a basic tenet of good science. This tenet holds all the more for digital science, with its fundamentally more concrete outputs of algorithms and models. However, there remain cultural and technical barriers to the sharing (using, repeating, comparing, contributing, reimplementing) of these artefacts, all the way from disseminating academic publications [1, 2, 3], through to recognition of the importance of scientific software [4] and computation [5].

We believe that an automated notify+reproduce system, which allows easy reproduction of the results of algorithms running on models, will help significantly with lifting the cultural burden and, by doing so, will vastly improve the efficiency of scientific exploration. Our Digital Science Catalyst Grant proposal builds on this belief. While there has been significant academic and policy discussion in this space over the past few years, now is the time for the creation of an extensible and adaptable open infrastructure to facilitate the automation of science reproducibility.

Description

We propose to develop a prototype open software platform which will automate reproducibility for algorithms and models. By developing a cloud-based, centralised service, which performs automated code compilation, testing and benchmarking, we will link together published implementations of algorithms and input models. This will allow the future extension of the prototype to link together software and data repositories, toolchains, workflows and outputs, providing a seamless automated infrastructure for the verification and validation of scientific models and in particular, performance benchmarks. The program of work will lead the cultural shift in both the short- and long-term to move to a world in which computational reproducibility helps researchers achieve their goals, rather than being perceived as an overhead.

A system as described here has several up-front benefits: it links papers more closely to their outputs, making external validation easier and allows interested users to explore unaddressed sets of models. Critically, it helps researchers to be more productive, rather than reproducibility being an overhead on their day-to-day work. In the same way that tools such as GitHub make collaborating easier while simultaneously allowing effortless sharing, we hope that we can design and build a system that is similarly usable for sharing and testing benchmarks online.

There are already several web services that can do aspects of this things (for example, a repository for disseminating the computational models associated with publications in the social and life sciences [6]), so a service that can integrate most if not all of these features is feasible. Such a service would then allow algorithms and models to evolve together, and be reproducible from the outset.

In summary, this proposed new infrastructure, previously highlighted and discussed by the authors [7, 8], would have a profound impact on the way that open computational science is performed, repositioning the role of models, algorithms and benchmarks and accelerating the research cycle, perhaps truly enabling a "fourth paradigm" of data intensive scientific discovery [9]. Furthermore, it would effect the vital cultural change by reducing overheads and improving the efficiency of researchers.

Vision, Objective and Goals

In the software development world, no one would (or should!) commit to a project without first running the smoke tests. You could be clever and run the tests via the version control system's pre-commit hook. That way you would never forget to run the tests. All of this can be done, at scale, on the cloud now. Services such as Jenkins[1], Visual Studio Online[2], etc, schedule the tests to run as soon as you commit. We envisage moving to a world in which benchmarks become available online, in the same vein as open access of publications and research data. It seems a small step to hook these continuous integration (CI) systems up to the algorithm implementations that are written to run on these benchmarks.

Suppose you have come up with a better algorithm to deal with some of these benchmarks. You write up the paper on the algorithm but, more importantly, you also register the implementation of your algorithm at this open service, as a possible algorithm to run on this benchmark set. The benchmarks live in distributed git repositories. Some of the servers that house these repositories are CI servers. Now, when you push a commit to your algorithm, or someone else pushes a commit to theirs, or when someone else adds a new benchmark, the service's CI system is triggered. It is also activated with the addition of a new library, firmware upgrade, API change, etc. All registered algorithms are run on all registered models, and the results are published. The CI servers act as an authoritative source, analogous to the Linux Kernel Archives[3], of results for these algorithms running on these benchmarks.

The objective of this proposal is to develop a system which, through integration with publicly available source code repositories automates the build, testing and benchmarking of algorithms and benchmarks. The system will allow testing models against competing algorithms, and the addition of new models to the test suite (either manually or from existing online repositories). The goals are to:

- Build a webserver, running a daemon which automatically pulls, and compiles code from git repositories;
- Run automated tests defined by the developers on the code;
- Perform analysis of benchmark sets supplied by both the developer and external users.
- Work with key stakeholders in the open software/open data/open access/open science space, as well as key e-infrastructural organisations e.g. Digital Science, GitHub, Microsoft Azure, Software Sustainability Institute, Mozilla Science Lab, Nesta, etc.

This will be achieved over a period of six months, including regular meetings for the design and requirements of the tool, and will involve the employment of a dedicated programmer to implement the system.

Budget

The major components of the budget are based around buying out Crick's time and enabling him to visit Cambridge to work with the rest of the project team, along with the employment of a dedicated programmer to implement the system:

1. We have costed six separate week-long visits to Cambridge for Crick: direct buy-out costs for a week are £350 (covering seven hours of teaching); one week's accommodation (seven nights in Cambridge at c.£130) £910; one week's subsistence (seven days at £25) £200; round trip travel: £180; (therefore, one individual trip costs £1600): **£9,600**
2. Programmer to implement the system, in conjuction with the project team: **£5,000**
3. Travel and subsistence for meetings with key stakeholders: **£400**

GRANT TOTAL: £15,000

[1] http://jenkins-ci.org/
[2] http://www.visualstudio.com/en-us/products/what-is-visual-studio-online-vs.aspx
[3] https://www.kernel.org/

Team Profile

Dr Tom Crick[4] is a Senior Lecturer in Computing Science at Cardiff Metropolitan University, having completed his PhD and post-doctoral research at the University of Bath. His research interests cut across computational science: knowledge representation and reasoning, intelligent systems, big data analytics, optimisation and high performance computing. He is the Nesta Data Science Fellow, a 2014 Fellow of the Software Sustainability Institute (EPSRC) and a member of *HiPEAC*, the European FP7 Network of Excellence on High Performance and Embedded Architecture and Compilation.

Dr Benjamin A. Hall[5] is a Royal Society University Research Fellow, developing hybrid and formal models of carcinogenesis and biological signalling at the MRC Cancer Unit, University of Cambridge. He previously worked at Microsoft Research (Cambridge), UCL and the University of Oxford. As part of his role at Oxford, he was one of two Apple Laureates, awarded by Apple and the Oxford Supercomputing Centre for the project *A biomolecular simulation pipeline*. Benjamin has an MBiochem and DPhil from the University of Oxford.

Dr Samin Ishtiaq[6] is Principal Engineer in the Programming Principles and Tools group at Microsoft Research Cambridge. He currently works on the SLAyer (Separation Logic-based memory safety for C programs), TERMINATOR (program termination) and BMA (analysis of gene regulatory networks) projects. Samin joined MSR in April 2008. Before that (2000-2008), he worked in CPU modelling and verification at ARM, helping to tape-out the Cortex A8, Cortex M3 and SC300 processors, and the AMBA bus protocol checker. Samin has an MEng from Imperial and a PhD in dependent type theory from Queen Mary.

References

[1] David De Roure. Replacing the Paper: The Twelve Rs of the e-Research Record. http://www.scilogs.com/eresearch/replacing-the-paper-the-twelve-rs-of-the-e-research-record/, November 2011.

[2] Victoria Stodden, Peixuan Guo, and Zhaokun Ma. Toward Reproducible Computational Research: An Empirical Analysis of Data and Code Policy Adoption by Journals. *PLoS ONE*, 8(6), 2013.

[3] Grigori Fursin and Christophe Dubach. Community-Driven Reviewing and Validation of Publications. In *Proceedings of the 1st ACM SIGPLAN Workshop on Reproducible Research Methodologies and New Publication Models in Computer Engineering (TRUST'14)*, pages 1–4. ACM Press, 2014.

[4] Carole Goble. Better Software, Better Research. *IEEE Internet Computing*, 18(5):4–8, 2014.

[5] Ian P. Gent. The Recomputation Manifesto. Available from: http://arxiv.org/abs/1304.3674, April 2013.

[6] Nathan D. Rollins, C. Michael Barton, Sean Bergin, Marco A. Janssen, and Allen Lee. A Computational Model Library for publishing model documentation and code. *Environmental Modelling & Software*, 61:59–64, 2014.

[7] Tom Crick, Benjamin A. Hall, and Samin Ishtiaq. "Can I Implement Your Algorithm?": A Model for Reproducible Research Software. In *Proceedings of 2nd International Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE2)*, 2014.

[8] Tom Crick, Benjamin A. Hall, Samin Ishtiaq, and Kenji Takeda. "Share and Enjoy": Publishing Useful and Usable Scientific Models. In *Proceedings of 1st International Workshop on Recomputability*, 2014.

[9] Tony Hey, Stewart Tansley, and Kristin Tolle, editors. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, 2009.

[4] http://drtomcrick.com

[5] http://www.mrc-cu.cam.ac.uk/hall.html

[6] http://research.microsoft.com/en-us/people/sishtiaq/