REPRODUCIBLE RESEARCH

ADDRESSING THE NEED FOR DATA AND CODE SHARING IN COMPUTATIONAL SCIENCE

By the Yale Law School Roundtable on Data and Code Sharing

Roundtable participants identified ways of making computational research details readily available, which is a crucial step in addressing the current credibility crisis.

Progress in computational science is often hampered by researchers' inability to independently reproduce or verify published results. Attendees at a roundtable at Yale Law School (www.stodden.net/RoundtableNov212009) formulated a set of steps that scientists, funding agencies, and journals might take to improve the situation. We describe those steps here, along with a proposal for best practices using currently available options and some long-term goals for the development of new tools and standards.

Why It Matters

Massive computation is transforming science. This is clearly evident from highly visible launches of large-scale data mining and simulation projects such as those in climate change prediction,[1] galaxy formation (www.mpa-garching.mpg.de/galform/virgo/millennium/), and biomolecular modeling (www.ks.uiuc.edu/Research/namd). However, massive computation's impact on science is also more broadly and fundamentally apparent in the heavy reliance on computation in everyday science across an ever-increasing number of fields.

Computation is becoming central to the scientific enterprise, but the prevalence of relaxed attitudes about communicating computational experiments' details and the validation of results is causing a large and growing credibility gap.[2] Generating verifiable knowledge has long been scientific discovery's central goal, yet today it's impossible to verify most of the computational results that scientists present at conferences and in papers.

To adhere to the scientific method in the face of the transformations arising from changes in technology and the Internet, we must be able to reproduce computational results. Reproducibility will let each generation of scientists build on the previous generations' achievements. Controversies such as ClimateGate,[3] the microarray-based drug sensitivity clinical trials under investigation at Duke University,[4] and prominent journals' recent retractions due to unverified code and data[5,6] suggest a pressing need for greater transparency in computational science.

Traditionally, published science or mathematics papers contained both the novel contributions and the information needed to effect reproducibility—such as detailed descriptions of the empirical methods or the mathematical proofs. But with the advent of computational research, such as empirical data analysis and scientific code development, the bulk of the actual information required to reproduce results is *not* obvious from an article's text; researchers must typically engage in extensive efforts to ensure the underlying methodologies' transmission. By and large, researchers today aren't sufficiently prepared to ensure reproducibility, and after-the-fact efforts—even heroic ones—are unlikely to provide a long-term solution. We need both disciplined ways of working reproducibly and community support (and even pressure) to ensure that such disciplines are followed.

On 21 November 2009, scientists, lawyers, journal editors, and funding representatives gathered for the Yale Law School Roundtable to discuss how data and code might be integrated with tradition research publications (www.stodden.net/RoundtableNov212009). The inspiration for the roundtable came from the example set by members of the genome research community who organized to facilitate the open release of the genome sequence data. That community gathered in Bermuda in 1996 to develop a cooperative strategy both for genome decoding and for managing the resulting data. Their meeting produced the *Bermuda Principles*, which shaped data-sharing practices among researchers in that community, ensuring rapid data release (see www.ornl.gov/sci/techresources/Human_Genome/research/bermuda.shtml). These principles have been reaffirmed and extended several times, most recently in a July 2009 *Nature* article.[7] Although the computational research community's particular incentives and pressures differ from those in human genome sequencing, one of our roundtable's key goals was to produce a publishable document that discussed data and code sharing.

 Copublished by the IEEE CS and the AIP 1521-9615/10/$26.00 © 2010 IEEE

Yale Law School Roundtable Participants

Writing Group Members:

- Victoria Stodden, Information Society Project, Yale Law School;
- David Donoho, Department of Statistics, Stanford University;
- Sergey Fomel, Jackson School of Geosciences, The University of Texas at Austin;
- Michael P. Friedlander, Department of Computer Science, University of British Columbia;
- Mark Gerstein, Computational Biology and Bioinformatics Program, Yale University;
- Randy LeVeque, Department of Applied Mathematics, University of Washington;
- Ian Mitchell, Department of Computer Science, University of British Columbia;
- Lisa Larrimore Ouellette, Information Society Project, Yale Law School;
- Chris Wiggins, Department of Applied Physics and Applied Mathematics, Columbia University.

Additional Authors:

- Nicholas W. Bramble, Information Society Project, Yale Law School
- Patrick O. Brown, Department of Biochemistry, Stanford University
- Vincent J. Carey, Harvard Medical School
- Laura DeNardis, Information Society Project, Yale Law School
- Robert Gentleman, Director, Bioinformatics and Computational Biology, Genentech
- J. Daniel Gezelter, Department of Chemistry and Biochemistry, University of Notre Dame
- Alyssa Goodman, Harvard-Smithsonian Center for Astrophysics, Harvard University
- Matthew G. Knepley, Computation Institute, University of Chicago
- Joy E. Moore, Seed Media Group
- Frank A. Pasquale, Seton Hall Law School
- Joshua Rolnick, Stanford Medical School
- Michael Seringhaus, Information Society Project, Yale Law School
- Ramesh Subramanian, Department of Computer Science, Quinnipiac University, and Information Society Project, Yale Law School

In reproducible computational research, scientists make all details of the published computations (code and data) conveniently available to others, which is a necessary response to the emerging credibility crisis. For most computational research, it's now technically possible, although not common practice, for the experimental steps—that is, the complete software environment and the data that generated those results—to be published along with the findings, thereby rendering them verifiable. At the Yale Law School Roundtable, we sought to address this in practical terms by providing current best practices and longer-term goals for future implementation.

Computational scientists can reintroduce reproducibility into scientific research through their roles as scientists, funding decision-makers, and journal editors. Here, we discuss best practices for reproducible research in each of these roles as well as address goals for scientific infrastructure development to facilitate reproducibility in the future.

The Scientist's Role

Roundtable participants identified six steps that computational scientists can take to generate reproducible results in their own research. Even partial progress on these recommendations can increase the level of reproducibility in computational science.

Recommendation 1: When publishing computational results, including statistical analyses and simulation, provide links to the source-code (or script) version and the data used to generate the results to the extent that hosting space permits. Researchers might post this code and data on

- an institutional or university Web page;
- an openly accessible third-party archived website designed for code sharing (such as Sourceforge.net, BitBucket.org, or Github.com); or
- on a preprint server that facilitates code and data sharing (such as Harvard's Dataverse Network; see http://thedata.org).

Recommendation 2: Assign a unique ID to each version of released code, and update this ID whenever the code and data change. For example, researchers could use a version-control system for code and a unique identifier such as the Universal Numerical Fingerprint (http://thedata.org/book/unf-implementation) for data. Such an identifier facilitates version tracking and encourages citation.[8] (As another example, the PubMed Central reference number applies to all manuscripts funded by the US National Institutes of Health, creating a unique, citable digital object identifier for each; see http://publicaccess.nih.gov/citation_methods.htm.)

Recommendation 3: Include a statement describing the computing environment and software version used in the publication, with stable links to the accompanying code and data. Researchers might also include a virtual machine. A VM image with compiled code, sources, and data that can reproduce published tables and figures would let others explore the parameters

The Protein Data Bank

One example of agency-facilitated openness is the Protein Data Bank. Created in 1971, PDB's aim is to share "information about experimentally determined structures of proteins, nucleic acids, and complex assemblies" (see www.pdb.org/pdb/home/home.do). PDB has become a standard within the structural biology community during the nearly 40 years of effort to balance relationships among the journals, the author-scientists, and the database itself.

The PDB is part of a worldwide effort funded by a variety of agencies, with main hubs in the US, Japan, and Europe. With the rise of the Web, PDB usage became more intimately connected with publication, first with the understanding that data were to be available within months or a year of publication, then—owing to the coordinated decisions of the editors of *Nature*, *Science*, *Cell*, and the *Proceedings of the National Academy of Sciences*—as a simple and effective precondition for publication.[1] This has in turn enabled an entire field of statistical studies and molecular dynamics based on the structural data, a feat impossible without access to each publication's data.

More information on *Nature*'s data requirement policies is available at www.nature.com/authors/editorial_policies/availability.html; *Science* requirements are included in its general author information at www.sciencemag.org/about/authors/prep/gen_info.dtl#dataavail.

Reference

1. "The Gatekeepers," editorial, *Nature Structural Biology*, vol. 5, no. 3, 1998, pp. 165–166; www.nature.com/nsmb/wilma/v5n3.892130820.html.

around the publication point, examine the algorithms used, and build on that work in their own new research.

Recommendation 4: Use open licensing for code to facilitate reuse, as suggested by the Reproducible Research Standard.[9,10]

Recommendation 5: Use an open access contract for published papers (http://info-libraries.mit.edu/scholarly/mit-copyright-amendment-form) and make preprints available on a site such as arXiv.org, PubMed Central, or Harvard's Dataverse Network to maximize access to the work. However, the goal of enhanced reproducibility applies equally to both open access journals and commercial publications.

Recommendation 6: To encourage both wide reuse and coalescence on broad standards, publish data and code in nonproprietary formats whenever reasonably concordant with established research practices, opting for formats that are likely to be readable well into the future when possible.

The Funding Agency's Role

Funding agencies and grant reviewers have a unique role due to their central position in many research fields. There are several steps they might take to facilitate reproducibility.

Recommendation 1: Establish a joint-agency-funded archival organization for hosting—perhaps similar to the Protein Data Bank (see the "Protein Data Bank" sidebar)—and include a system for permitting incoming links to code and data with stable unique identifiers. For example, PubMed Central could be extended to permit code and data upload and archiving (possibly mirrored with existing version-control systems).

Recommendation 2: Fund a select number of research groups to fully implement reproducibility in their workflow and publications. This will allow a better understanding of what's required to enable reproducibility.

Recommendation 3: Provide leadership in encouraging the development of a set of common definitions permitting works to be marked according to their reproducibility status, including verified, verifiable, or inclusive of code or data.

Recommendation 4: Fund the creation of tools to better link code and data to publications, including the development of standardized unique identifiers and packages that allow the embedding of code and data within the publication (such as Sweave[11] or GenePattern[12]).

Recommendation 5: Fund the development of tools for data provenance and workflow sharing. It can often take researchers considerable time to prepare code and data for verification; provenance and workflow tracking tools could greatly assist in easing the transition to reproducibility. Examples include the UK-funded Taverna software package (www.mygrid.org.uk), the University of Southern California's Pegasus system (http://pegasus.isi.edu), Penn State University's Galaxy software (http://galaxy.psu.edu), and Microsoft's Trident Workbench for oceanography (http://research.microsoft.com/enus/collaboration/tools/trident.aspx).

The Journal Editor's Role

Journals are key to establishing reproducibility standards in their fields and have several options available to facilitate reproducibility.

Recommendation 1: Implement policies to encourage the provision of stable URLs for open data and code associated with published papers. (For an example, see Gary King's draft journal policy at http://gking.harvard.edu/repl.shtml.) Such URLs might be links to established repositories or to sites hosted by funding agencies or journals.

Recommendation 2: When scale permits, require the replication of computational results prior to publication, establishing a reproducibility review. To ease the burden on reviewers, publications could provide a server through which authors can upload their code and data to ensure code functionality before the results verification.

Recommendation 3: Require appropriate code and data citations through standardized citation mechanisms, such as Data Cite (http://thedata.org/citation/tech).

Several journals have implemented policies that advance sharing of the data and code underlying their

computational publications. A prominent example is *Biostatistics*, which instituted an option in 2009 for authors to make their code and data available at publication time.[13] The journal itself hosts the associated data and code; code written in a standard format will also be verified for reproducibility, and the published articled is labeled accordingly. Authors can choose to release only the paper itself or to also release the code, the data, or both data and code (making the paper fully reproducible), indicated as C, D, or R, respectively, on the title pages. The policy is having an impact. Since it was implemented, three issues with a total of 43 papers have been published; of those, four papers have been marked with code availability, two with data availability, one with both, and two as fully reproducible.

In addition to traditional categories of manuscript (research, survey papers, and so on), the ACM journal *Transactions on Mathematical Software* has for many years let authors submit under a special "Algorithm" category (http://toms.acm.org). Submissions in this category include both a manuscript and software, which are evaluated together by referees. The software must conform to the *ACM Algorithms Policy*, which includes rules about completeness, portability, documentation, and structure designed "to make the fruits of software research accessible to as wide an audience as possible" (see www.cs.kent.ac.uk/projects/toms/AlgPolicy.html). If accepted, the manuscript component of an algorithm submission is published in the traditional fashion, but flagged prominently in the title as an algorithm, and the software becomes part of the AMC's collected algorithms, which are available for download and subject to the ACM Software Copyright and License Agreement. Although not appearing as frequently as traditional research papers, algorithm articles still make up a significant fraction of published articles in the journal despite the additional effort required of both authors and referees. In 2009, for example, seven out of 22 articles were in the algorithm category.

Geophysics, a prominent journal in the geosciences, created a special section on "Algorithms and Software" in 2004 (http://software.seg.org). Authors in this section must supply source code, which is reviewed by the journal to verify reproducibility of the results. The code is archived on the website. The journal *Bioinformatics* encourages the submission of code, which is actively reviewed, and an option is available for letting the journal archive the software (see www.biomedcentral.com/bmcbioinformatics/ifora/?txt_jou_id=1002&txt_mst_id=1009). *Nucleic Acids Research* publishes two dedicated issues annually: one entirely devoted to software and Web services useful to the biological community, and the other devoted to databases. The software is reviewed prior to publication and is expected to be well tested and functional prior to submission (www.oxfordjournals.org/our_journals/nar/for_authors/submission_webserver.html).

Unfortunately, archived code can become unusable—sometimes quickly—due to changes in software and platform dependencies, making published results irreproducible. One improvement here would be a system with a devoted scientific community that continues to test reproducibility after paper publication and maintains the code and the reproducibility status as necessary. When code is useful, there's an incentive to maintain it. Journals can facilitate this by letting authors post software updates and new versions.

Long-Term Goals

The roundtable participants also extended their discussion of recommendations beyond immediately available options. This section describes potential future developments, including ideal tools and practices that we might develop to facilitate reproducibility.

Goal 1: Develop version-control systems for data—particularly systems that can handle very large and rapidly changing data. Because many different research communities use computational tools, we should develop version-control systems for all aspects of research (papers, code, and data). Ideally, these would incorporate GUIs or Web-based tools to facilitate their use.

Goal 2: Publish code accompanied by software routines that permit testing of the software—test suites, including unit testing and/or regression tests, should be a standard component of reproducible publication. In addition, we should develop tools to facilitate code documentation. In the Python world, for example, the Sphinx machinery makes it possible to converge on a standard for documentation that produces consistent, high-quality documents in LaTeX, PDF, and HTML, with good math and graphics support that can be fully integrated in the development process (see http://sphinx.pocoo.org).

Goal 3: Develop tools to facilitate both routine and standardized citation of code, data, and contribution credits, including micro-contributions such as dataset labeling and code modifications, as well as to enable stable URL citations.

Goal 4: Develop tools for effective download tracking of code and data, especially from academic and established

third-party websites, and use these data in researcher evaluation.

Goal 5: Mark reproducible published documents as such in an easily recognizable and accepted way.[9,12,13]

Goal 6: Require authors to describe their data using standardized terminology and ontologies. This will greatly streamline the running of various codes on data sets and a uniform interpretation of results.

Goal 7: That institutions, such as universities, take on research compendia archiving responsibilities as a regular part of their role in supporting science. This is already happening in several places, including Cornell University's DataStar project.[14,15]

Goal 8: Clarify ownership issues and rights over code and data, including university, author, and journal ownership. Develop a clear process to streamline agreements between parties with ownership to facilitate public code and data release.

Goal 9: Develop deeper communities that maintain code and data, ensure ongoing reproducibility, and perhaps offer tech support to users. Without maintenance, changes beyond individual's control (computer hardware, operating systems, libraries, programming languages, and so on) will break reproducibility. Reproducibility should become the responsibility of a scientific community, rather than rest on individual authors alone.

Novel contributions to scientific knowledge don't emerge solely from running published code on published data and checking the results, but the ability to do so can be an important component in scientific progress, easing the reconciliation of inconsistent results and providing a firmer foundation for future work.

Reproducible research is best facilitated through interlocking efforts in scientific practice, publication mechanisms, and university and funding agency policies occurring across the spectrum of computational scientific research. To ultimately succeed, however, reproducibility must be embraced at the cultural level within the computational science community.[16] Envisioning and developing tools and policies that encourage and facilitate code and data release among individuals is a crucial step in that direction.

References

1. R. Stevens, T. Zacharia, and H. Simon, *Modeling and Simulation at the Exascale for Energy and the Environment*, report, US Dept. Energy Office of Advance Scientific Computing Research, 2008; www.sc.doe.gov/ascr/ProgramDocuments/Docs/TownHall.pdf.
2. D. Donoho et al., "Reproducible Research in Computational Harmonic Analysis," *Computing in Science & Eng.*, vol. 11, no. 1, 2009, pp. 8–18.
3. "The Clouds of Unknowing," *The Economist*, 18 Mar. 2010; www.economist.com/displaystory.cfm?story_id=15719298.
4. K. Baggerly and K. Coombes, "Deriving Chemosensitivity from Cell Lines: Forensic Bioinformatics and Reproducible Research in High-Throughput Biology," *Annals Applied Statistics*, vol. 3, no. 4, 2009, pp. 1309–1334.
5. B. Alberts, "Editorial Expression of Concern," *Science*, vol. 327, no. 5962, 2010, p. 144; www.sciencemag.org/cgi/content/full/327/5962/144-a.
6. G. Chang et al., "Retraction," *Science*, vol. 314, no. 5807, 2006, p. 1875; www.sciencemag.org/cgi/content/full/314/5807/1875b.
7. Toronto International Data Release Workshop, "Prepublication Data Sharing," *Nature*, vol. 461, pp. 168–170; www.nature.com/nature/journal/v461/n7261/full/461168a.html.
8. M. Altman and G. King, "A Proposed Standard for the Scholarly Citation of Quantitative Data," *D-Lib Magazine*, vol. 13, nos. 3-4, 2007; www.dlib.org/dlib/march07/altman/03altman.html.
9. V. Stodden, "Enabling Reproducible Research: Licensing for Scientific Innovation," *Int'l J. Comm. Law & Policy*, vol. 13, Jan. 2009; www.ijclp.net/issue_13.html.
10. V. Stodden, "The Legal Framework for Reproducible Scientific Research: Licensing and Copyright," *Computing in Science & Eng.*, vol. 11, no. 1, 2009, pp. 35–40.
11. F. Leisch, "Sweave and Beyond: Computations on Text Documents," *Proc. 3rd Int'l Workshop on Distributed Statistical Computing*, 2003; www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/Leisch.pdf.
12. J. Mesirov, "Accessible Reproducible Research," *Science*, vol. 327, no. 5964, 2010, pp. 415–416.
13. R. Peng, "Reproducible Research and *Biostatistics*," *Biostatistics*, vol. 10, no. 3, 2009, pp. 405–408.
14. G. Steinhart, D. Dietrich, and A. Green, "Establishing Trust in a Chain of Preservation: The TRAC Checklist Applied to a Data Staging Repository (DataStaR)," *D-Lib Magazine*, vol. 15, nos. 9-10, 2009.
15. G. Steinhart, "DataStar: An Institutional Approach to Research Data Curation," *IAssist Quarterly*, vol. 31, no. 3-4, 2007, pp. 34–39.
16. V. Stodden, *The Scientific Method in Practice: Reproducibility in the Computational Sciences*, paper no. 4773-10, MIT Sloan Research, 9 Feb. 2010; http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1550193.