RunMyCode.org: a novel dissemination and collaboration platform for executing published computational results

Christophe Hurlin, Christophe Pérignon, Victoria Stodden

RunMyCode.org: a novel dissemination and collaboration platform for executing published computational results

Christophe Hurlin
Department of Economics
University of Orléans, France
Christophe.Hurlin@univ-orleans.fr

Victoria Stodden
Department of Statistics
Columbia University
New York City, USA
victoria@stodden.net

Christophe Pérignon
Finance Department
HEC Paris, France
perignon@hec.fr

***Abstract*—We believe computational science as practiced today suffers from a growing *credibility gap* – it is impossible to replicate most of the computational results presented at conferences or published in papers today. We argue that this crisis can be addressed by the open availability of the code and data that generated the results, in other words practicing reproducible computational science. In this paper we present a new computational infrastructure called RunMyCode.org that is designed to support published articles by providing a dissemination platform for the code and data that generated the their results. Published articles are given a *companion webpage* on the RunMyCode.org website from which a visitor can both download the associated code and data, and execute the code in the cloud directly through the RunMyCode.org website. This permits results to be verified through the companion webpage or on a user's local system. RunMyCode.org also permits a user to upload their own data to the companion webpage to check the code by running it on novel datasets. Through the creation of "coder pages" for each contributor to RunMyCode.org, we seek to facilitate social network-like interaction. Descriptive information appears on each coder page, including demographic data and other companion pages to which they made contributions. In this paper we motivate the rationale and functionality of RunMyCode.org and outline a vision of its future.**

***Index Terms*—reproducible research, reproducible computational science, dissemination platform, collaborative networks, cloud computing, executable papers, code sharing, data sharing, open science**

I. What is Reproducible Computational Science?

Over the previous 20 years the practice of science has undergone a revolution, that will finish with computation as absolutely central to the scientific enterprise. Today's academic scientist is more likely to resemble a computer jockey working at all hours to launch experiments on computer servers. Long gone is the traditional image of the scientists as a solitary person working in a laboratory or scribbling in the back aisles of the mathematics library. This transition has not been smooth, and we believe it has already brought us to a state of crisis. The vast majority of scientific results generated by current computational science practice suffer a large and growing credibility gap: it is impossible to replicate and verify most of the computational results published or presented in conferences today.

There have traditionally been two branches of the scientific method, the deductive and the inductive. The deductive branch encompasses fields such as mathematics and logic, and the inductive field comprises the empirical sciences. There has been much discussion, even at very high policy levels, over the last few years regarding the emergence of new branches of the scientific method arising from the computational revolution [1]. Both cpu-intensive simulation and the data deluge are provoking discussion of third and fourth branches of the scientific method, but we cannot elevate computational science to a new branch of the scientific method until we can generate reliable verifiable computational knowledge.

The central motivation for the scientific method is the ubiquity of error – the phenomenon that mistakes and self-delusion can creep in absolutely anywhere in the process of generating scientific findings, and that the work of the scientist is primarily about recognizing and rooting out error. Both the deductive and inductive sciences are error prone and have consequently developed standards for the dissemination of results. In the deductive sciences the formal notion of the mathematical proof is a mature response to the ubiquity of error. Similarly, the empirical sciences employ the machinery of hypothesis testing, controlled experiments, and the disciplined reporting of data, material, and methods in a standard format in published research articles. In the computational sciences Jon Claerbout, professor emeritus of geophysics at Stanford University, pioneered a system for linking code and data with the final paper or thesis and giving

the reader the capability to regenerate the results. His insight is as follows, "an article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures" [2]. In this way, the code and data that produced the computational results are available alongside the published paper, enabling readers to verify and validate the results.

The efforts pioneered by Claerbout were emulated by David Donoho, professor of statistics at Stanford University. Starting in the early 1990's, his approach was to supply the details underlying the datasets, simulations, figures, and tables in a uniform way in the standard MATLAB scripting language, and makes these files available on the Internet. Interested readers could then reproduce the calculations underlying that paper. Donoho required his students to work in this fashion. At the time he was working in computational harmonic analysis (wavelets, wavelet packets, time frequency methods) and very few computational tools were available. After several papers on the subject of wavelets had been written, it became possible to combine all the tools into a single package, *WaveLab*, which contained a unified set of wavelet and time-frequency tools and reproduced all the calculations in any of several papers. *WaveLab* emerged as a standard in the field. Since then Donoho and his students have released several subject-oriented dissemination platforms in the *WaveLab* framework for several research domains:

- *Atomizer*. This is a toolbox for overcomplete signal representation via l1 minimization; originating from Scott Chen's thesis [3] and other related journal articles [4, 5].
- *BeamLab*. This is a toolbox for multiscale geometric analysis, originating from Xiaoming Huo's and Ofer Levi's theses [6, 7], Georgina Flesia's postdoctoral work [8, 9], and other related journal papers [10, 11, 12, 13, 14, 15].
- *SymmLab*. This is a toolbox for multiscale analysis of manifold-valued data, originating from Inam Ur-Rahman's thesis and other related journal articles [16, 17].
- *SparseLab*. This is a toolbox for the estimationg f overcomplete models and facilitating sparsity-seeking decomposition and reconstruction, originating from Victoria Stodden, Joshua Sweetkind-Singer, and Yaakov Tsaig's theses [18, 19, 20], Iddo Drori's postdoctoral work, and other related papers [21, 22, 23, 24].
- *SphereLab*. This is a toolbox for multiscale decomposition of data on the sphere, originating from Morteza Shahram's postdoctoral work [25].

In 2004, Gentlemen and Temple Lang [26] introduced the concept of the *compendium*: a new way of disseminating research results that provides not only the article, but also the software tools and data required to reproduce the published findings. These pioneering efforts serve as the model upon which we built RunMyCode.org, introduced in section IV.

II. Why Practice Reproducible Computational Science?

We believe researchers will conduct better science if they begin a computational project with the knowledge that their code and data will be revealed at the time of publication, with the aim of giving independent readers the ability to verify their results. The primary reason to practice really reproducible research is for oneself. Research coding typically uses short term memory and with sufficient time, the detailed series of small decisions made in the course of research will often be difficult for even the original programmer to recreate.

Computationally oriented co-authors also need access to the data and code associated with publications to which they are making contributions, not only to understand the results they are taking responsibility for, but also to verify the integrity of the work. Incoming graduate and postdoctoral students typically replicate previously published results when starting research in a new area. If they have access to the underlying code and data the efforts expended duplicating the research will be greatly reduced. Referees of papers and grant proposals can verify and better understand submitted findings, and even future employers can obtain a sense not only of the technical expertise of the authors, but also of their awareness of the important of generating reproducible computational science.

Data are noisy, multiple steps are typically taken to ready the data for analysis, and the analysis itself can comprise a variety of different parameter settings and other decisions that affect the outcomes and findings, sometimes greatly. These steps and settings are not fully reported in a typical published article, such that the computational results can be conveniently replicated by others.

III. Barriers to Practicing Reproducible Computational Science

In a 2009 survey conducted on the machine learning committee, 723 US-based academics were asked their reasons for sharing or withholding code and data [27]. Topping the list was the time it takes to document and clean up the code and data, and then comes providing user support.

Withhold Code		Withhold Data
77%	Time to document and clean up	54%
52%	Dealing with questions from users	34%
44%	Not receiving attribution	42%
40%	Possibility of patents	-
34%	Legal barriers (ie. copyright)	41%
-	Time to verify release with admin	38%
30%	Potential loss of future publications	35%
30%	Competitors may get an advantage	33%
20%	Web/Disk space limitations	29%

Next is concern about not receiving attribution, and potential loss of future publications is seventh.

These objections and problems, including the time it takes to carry out really reproducible research, the lack of credit, advantaging competitors, and prohibitive complexity of the computational environment, are echoed in other published literature [28]. Arguments are given to counter each of these reasons and paraphrased as follows. Having students carry out reproducible research actually decreases the amount of time supervision takes, and makes it much easier to verify and explain any of your own results should the need arise. If the concern is potential lack of credit, the very fact that you are a pioneer in reproducible research practices means the effort is more likely to be noticed and more likely to be recognized and rewarded. Enabling strangers to compete with you might happen if you release code and data, but it can also happen if you publish papers at all. Complicated computing environments pose an issue to reproducible research, but they are also a case where reproducibility is *more* important than the simpler case. A more complex computing environment has more possibilities for failure and fewer opportunities to check one's work. In it in these cases that reproducibility is especially important, even if it is harder to reveal one's work.

IV. What is RunMyCode.org?

The RunMyCode.org website was launched in January 2012 to disseminate code and data associated with published computational results, resolving some of the problems described above. It is a non-for profit academic website based on the innovative concept of a *companion webpage* associated with a scientific publication (article or working paper). Specifically, a companion webpage is a webpage that allows people to run computer codes associated with a scientific publication using their own data and parameter values, or download the code and data directly. The service only requires a web browser as calculations are done on a dedicated cloud computer. Once the results are ready, they are automatically displayed to the user, as a SaaS (Software as a Service), see Figure 1. Both the use and the creation of the companion webpages are free and do not require any particular programming skills.

This concept can be viewed as a novel attempt to provide, on a large scale, an executable paper solution. The only difference with the executable paper approach proposed by the scientific publishers (see for the instance the Elsevier's Executable Paper Grand Challenge, 2011) is that the companion webpage is not encapsulated within the text of a scientific publication [29]. In that sense, a companion webpage can be considered as providing additional material for a scientific publication.

RunMyCode.org has three main objectives: (1) to allow researchers to quickly disseminate the results of their research to an international audience, (2) to provide a very large community of users with the ability to use the latest scientific methods in a user-friendly environment, and (3) to allow members of the academic community (researchers, editors, referees, etc.) to replicate scientific results and to demonstrate their robustness.

Figure 1. Structure of the RunMyCode.org website. Researchers provide the code and data associated with their publication. Users provide their own data, which are sent to the cloud along with the computer code. When ready, the results are sent back to the user.

Figure 2 shows an example of a companion webpage for a published paper in econometrics on the estimation of the future volatility of an economic variable. The page provides a link to the published paper either on the publisher website or on a pdf archive platform such as arXiv.org or SSRN; both the underlying code and data are available for download and permit the execution of the code in the cloud to directly verify published results. RunMyCode.org houses close to 100 author-contributed datasets and models today, all accessible via download and through the computational facilities at RunMyCode.org. Note that coder profiles are maintained within RunMyCode.org and displayed on each companion page to which they have contributed. This permits credit to be given to those who contributed to the programming and datasets involved in the research, and forms the basis for collaborative social networks.

There are three main parts in a companion webpage. The upper section provides information on the original scientific paper including a link, along with information about authors. The middle part displays information about the coders, i.e., the individuals who wrote the computer code that allows to implement the methodology presented in the paper, or to reproduce the findings reported in the paper. Note that the coders can be different from the authors. Finally, the lower part of the companion webpage allows users to choose parameter values and to upload data; either data provided by the coder or users' own data. Computation is launched by clicking the RunMyCode green button in the lower right corner.

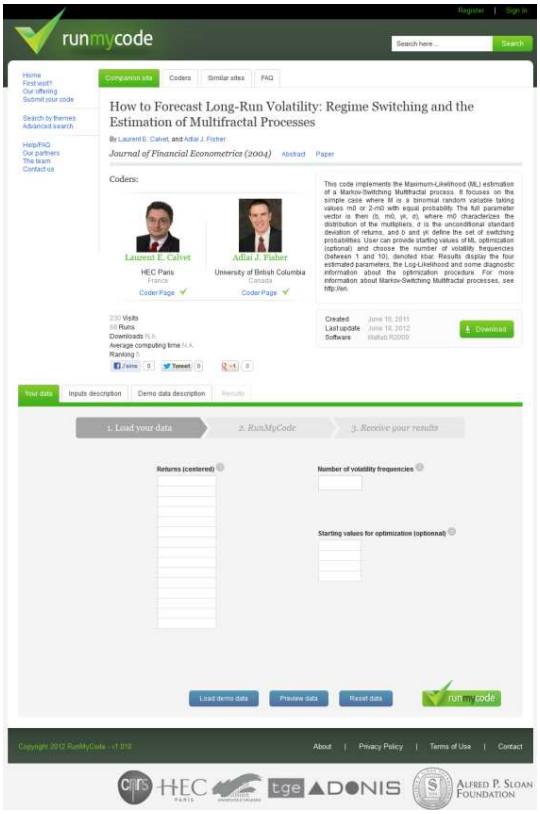

Figure 2. Example of a scientific paper's companion webpage on RunMyCode.org

Each contributor within RunMyCode.org is given a unique profile called a "code page." Figures 3 and 4 illustrate the link to the coder profiles from a sample companion page, and a coder profile for a contributor to RunMyCode.org.

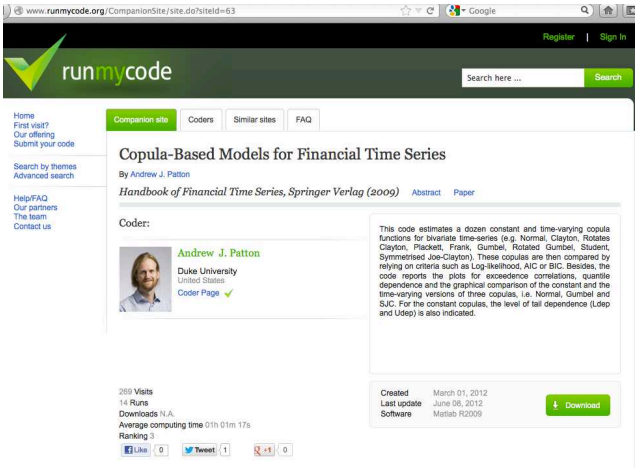

Figure 3. Abbreviated example of a scientific paper's companion webpage on RunMyCode.org, showing the link to the contributor's "coder page"

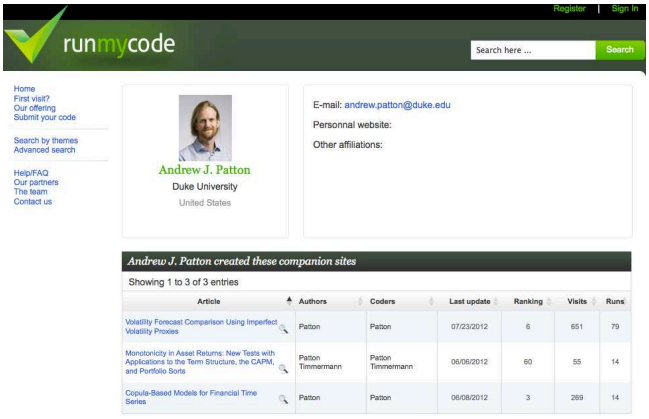

Figure 4. Example of a "coder page," listing demographic information including the contributions made by an individual to RunMyCode.org

These coder pages form the basis of a collaborative research network, permitting researcher to find and connect with people working on similar or other interesting problems. On each coder page it is possible to see what other researchers a particular person has collaborated with and browse those companion webpages.

RunMyCode.org allows researchers to create a companion webpage online without any particular computing skills. There are four main steps in the companion webpage creation process. In the first step, the researcher provides information about the scientific paper, coders, and computer code (including software version, input, and output). In practice, the companion webpage reproduces the output of the computer code (tables, figures, numerical values, text, image, etc.) as they would appear on the researcher's personal computer. The final task for the researcher is to preview and validate her companion webpage.

Note that the creation of a companion webpage does not require any modification to the scripts and, as such, requires no additional effort from the researcher. Depending on the complexity of the script, type and number of constraints, input, and output, the creation of a new companion webpage by the RunMyCode.org team takes anywhere between 20 minutes and a few hours.

Next, the editorial team checks whether the topic complies with the editorial policy of the website, similarly to any peer-reviewed academic journal. Then a technical validation of the code is undertaken, which at time the companion webpage will be certified by RunMyCode.org. The RunMyCode Lab checks four different dimensions of the code: (1) compatibility of the software; (2) robustness of the code: testing parameter values and constraints; (3) security of the code; and (4) CPU requirements and computing time. Then, the code is uploaded on the cloud and the companion webpage goes online.

Currently, it is possible to create a companion webpage from code written in C++, Fortran, MATLAB, R, and Rats. More software will be added in the near future. For all the applications, cloud facilities are provided by the French National Research Agency (CNRS)'s TGE Adonis.

The RunMyCode.org website was first developed in social sciences, with a strong emphasis on economics and

management. An important stream of research in this field is the modeling of financial risk. Some of the risk measures developed by academics have played a central role in the current debate on banking regulation (Dodd–Frank Wall Street Reform and Consumer Protection Act, 2010) [30]. In this perspective, a companion webpage allows people to better assess the properties of a given risk measure and its expected consequences on financial markets.

With the financial support of the Alfred Sloan Foundation and of the HEC Paris Foundation, RunMyCode.org is being extended to other computational sciences, including applied mathematics, statistics, and image processing.

V. What Problems Does RunMyCode.org Solve?

In many computational sciences, the algorithm accounts for a significant fraction of the scientific contribution. However, lack of disclosure of the computer code associated with the scientific paper has two pernicious effects. First, it prevents people from using the method or replicating the findings without recoding the algorithm in the paper. Second, it prevents editors and referees from checking the codes in order to check the accuracy of the results.

Furthermore, even when the computer code is publicly disclosed, many potential users cannot use it because they do not have the necessary coding skills to implement it, nor the right versions of the software/compilers, nor the appropriate computing capacities. This situation is frustrating for the original researchers because it prevents their research output from being broadly used and cited. Similarly, potential users of the research output, including other researchers, students, corporations, administrations, and sometimes the general public, can only access and benefit from a tiny fraction of the overall scientific output. This leads to an obvious mismatch between the supply and the demand of new research ideas and methods, and little transfer of technology from the academia to society. We think that RunMyCode.org alleviates some of these problems.

First, it makes research easier to use and replicate, which in turn can boost the transfer of technology. Indeed, when a companion webpage is available for a given piece of research, all potential users can give a try right away. Of particular interest is the fact that RunMyCode.org constitute an effective way to democratize the access to the latest academic research in developing or emerging countries. Over the past six months, around 15% of all our site visits were from developing countries.

Second, RunMyCode.org enriches the refereeing process as editors and referees can now access and assess the computer code associated with a submitted paper. This additional piece of information can help them to make a more informed decision in a short period of time. They can evaluate the robustness of the methodology proposed by the authors to some changes in parameter values and data without any extra programming effort.

When results are produced by RunMyCode.org a suggested citation is supplied, for both the code and data that were used in generating the results, and for the RunMyCode.org website.

VI. What Is The Vision of RunMyCode.org?

There are many areas in which we hope RunMyCode.org will help make scientific research easier to verify and use.

- We would like RunMyCode.org to become a leading repository for scientific code and data in in computational fields.
- We would like RunMyCode.org to operate on a global scale, with support units and computing facilities in various institutions throughout the world.
- We would like RunMyCode.org to help change the refereeing process of scientific papers. We see great value to the scientific progress if academic journals' editors could access an anonymized companion webpage before reviewing the paper. Referees can then use the companion webpage to validate the main findings and check their robustness. Once the paper has been accepted, there is no need for the companion webpage to remain anonymous and the researchers' names can be added.
- We would like RunMyCode.org to become a scientific social network, on which researchers present and promote their research. The researcher can decide on whether she wants to share her code and data with the entire planet or restrict part of her material to a smaller group of people. By doing so, the researcher can better control citations of her work. Such a social network would also encourage collaborative code development.
- We would like RunMyCode.org to become a certification device for computer code and data in science. The RunMyCode.org certification will help researchers to publish and promote their research.
- We would like RunMyCode.org to become an innovative teaching tool allowing professors to bring research into the classroom and expose students to the latest research developments. There are two ways we can suggest using RunMyCode.org in a course. First, the instructor can create a portfolio of existing companion webpages within the RunMyCode.org platform. Students access a login that allows them to directly access this portfolio of companion webpages. Alternatively, a more advanced solution would be to create a companion webpage directly associated with a course, and not with a single paper. This would allow different methods to be run on a unique dataset, for example.
- We would like RunMyCode.org to become a market for scientific talent. Employers can look for people with a particular expertise in a given scientific area, especially using the "coder page" associated with each contributor.
- We would like to leverage the social network aspects built into RunMyCode.org and further encourage researchers to discover and interact with others, including messaging and interfacing with other social network sites.

- We would like RunMyCode.org to be used by funding agencies to impose standards for research grants in terms of disclosure of data and code.
- We would like RunMyCode to collaborate with publishing companies to link published papers to their RunMyCode.org companion webpages.
- We would like RunMyCode.org to model best practices for reproducible research.

RunMyCode.org solves several of the problems given in section III confronting computational scientists who wish to practice really reproducible research. It removes the difficulty of hosting the code and data, it removes the difficulty of installing and running (even correct) code on a local computer system, and by providing the ability for users to execute the code in the cloud, it minimizes the amount of support coders and authors are asked to supply. RunMyCode.org also provides suggested citations, to help encourage a reward system that encourages code and data release, by giving credit for these scientific contributions. RunMyCode.org provides a public date of creation of the companion webpage, helping to ensure primacy to those who release code and data and encourage attribution. Perhaps most importantly, RunMyCode.org provides a central field-independent platform to facilitate both code and data sharing, and the verification of published computational results [31].

REFERENCES

[1] National Science Board Report, Digital research data sharing and management. December 2011. Available at http://www.nsf.gov/nsb/publications/2011/nsb1124.pdf

[2] J. Buckheit and D. L. Donoho. Wavelets and statistics, chapter Wavelab and reproducible research. Springer-Verlag, Berlin, New York, 1995.

[3] S. S. Chen. Basis pursuit. PhD thesis, Department of Statistics, Stanford University, Stanford, CA, 1996.

[4] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing, 20(1):33–61, 1999.

[5] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. SIAM Review, 43(1):129–159, 2001.

[6] X. Huo. Sparse image representation via combined transforms. PhD thesis, Department of Statistics, Stanford University, Stanford, CA, August 1999.

[7] O. Levi. Multiscale geometric analysis of three-dimensional data. PhD thesis, Department of Statistics, Stanford University, Stanford, CA, 2005.

[8] A. G. Flesia, H. Hel-Or, A. Averbuch, E. J. Candès, R. R. Coifman, and D. L. Donoho. Beyond wavelets, chapter Digital Implementation of ridgelet packets. Academic Press, New York, 2002.

[9] D. L. Donoho and A. G. Flesia. Beyond wavelets, chapter Digital ridgelet transform based on true ridge functions. Academic Press, New York, 2002.

[10] D. L. Donoho and Xiaoming Huo. Beamlab and reproducible research. International Journal of Wavelets, Multiresolution and Information Processing, 2(4):391-414, 2004.

[11] A. Averbuch, R. Coifman, D. Donoho, M. Israeli, and J. Walden. Fast slant stack: a notion of radon transform for data on a cartesian grid which is rapidly computable, algebraically exact, geometrically faithful, and invertible. Technical report, Department of Statistics, Stanford University, 2001.

[12] D. L. Donoho and X. Huo. Beamlet pyramids: a new form of multiresolution analysis, suited for extracting lines, curves and objects from very noisy image data. In Wavelet applications in signal and image processing, volume 4119, pages 434–444, San Diego, CA, 2000.

[13] D. L. Donoho and X. Huo. Beamlets and multiscale image analysis, volume 20 of Springer Lecture Notes in Computational Science and Engineering, pages 149–196. 2002.

[14] D. L. Donoho and O. Levi. Modern Signal Processing, chapter Fast X-ray and beamlet transforms for three dimensional data. Cambridge University Press, 2003.

[15] D. L. Donoho, O. Levi, J. L. Starck, and V. J. Martinez. Multiscale geometric analysis for 3d catalogs. Proceedings of SPIE, 4847:101–111, 2002.

[16] I. Rahman. Multiscale decomposition of manifold-valued data. PhD thesis, Program in Scientific Computing and Computational Math, Stanford University, Stanford, CA, July 2006.

[17] I. Rahman, I. Drori, V. Stodden, D. L. Donoho, and P. Schroder. Multiscale representations for manifold-valued data. SIAM Multiscale Modeling and Simulation, 4(4):1201-1232, 2005.

[18] V. Stodden. Model selection when the number of variables exceeds the number of observations. PhD thesis, Department of Statistics, Stanford University, Stanford, CA, 2006.

[19] J. Sweetkind-Singer. Log-penalized linear regression. PhD thesis, Stanford University Department of Electrical Engineering, 2004.

[20] Y. Tsaig. Sparse solution of underdetermined linear systems: algorithms and applications. PhD thesis, Department of Statistics, Stanford University, Stanford, CA, 2007.

[21] D. L. Donoho, Y. Tsaig, I. Drori, and J. Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. Technical Report 2006-2, Stanford University, 2006.

[22] D. L. Donoho and Y. Tsaig. Fast solution of l1 minimization problems when the solution may be sparse. IEEE Trans. Info. Theory, 54(11):4789-4812, 2008.

[23] D Donoho and Y Tsaig. Extensions of compressed sensing. Signal Processing, 86(3):549-571, March 2006.

[24] E. H. Hurowitz, I. Drori, V. C. Stodden, D. L. Donoho, and P. Brown. Virtual northern analysis of the human genome. PLoS ONE, 2(5), 2007.

[25] M. Shahram, D. L. Donoho, and J. L. Stark. Multiscale representation for data on the sphere and applications to geopotential data. Proceedings of SPIE Annual Meeting, 2007.

[26] R. Gentleman and D. Temple Lang, Statistical analyses and reproducible research. 2004. Available at http://biostats.bepress.com/bioconductor/paper2/

[27] V. Stodden, The scientific method in practice: reproducibility in the computational sciences," MIT Sloan School Working Paper 4773-10, 2010. Available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1550193.

[28] D. L. Donoho, A. Maleki, I. Ur Rahman, M. Shahram, V. Stodden, Reproducible research in computational harmonic analysis. IEEE Computing in Science and Engineering, 11(1):8-18, 2009.

[29] Elsevier's Executable Paper Grand Challenge, 2011, Elsevier. http://www.executablepapers.com/

[30] Dodd–Frank Wall Street Reform and Consumer Protection Act, 2010, US Congress, Washington DC.

[31] Victoria Stodden, Trust your science? Open your data and code. Amstat News, July 1, 2011.