# Exploring the Coming Repositories of Reproducible Experiments: Challenges and Opportunities

Juliana Freire
New York University
juliana.freire@nyu.edu

Philippe Bonnet
IT University of Copenhagen
phbo@tu.dk

Dennis Shasha
New York University
shasha@courant.nyu.edu

## ABSTRACT

Computational reproducibility efforts in many communities will soon give rise to validated software and data repositories of high quality. A scientist in a field may want to query the components of such repositories to build new software workflows, perhaps after adding the scientist's own algorithms. This paper explores research challenges necessary to achieving this goal.

## 1. INTRODUCTION

A hallmark of the scientific method is that experiments should be described in enough detail that they can be repeated and perhaps generalized. The idea in natural science is that if a scientist claims an experimental result, then another scientist should be able to check it. Similarly, in a computational environment, it should be possible to *repeat* a computational experiment as the authors have run it or to change the experiment to see how robust the authors' conclusions are to changes in parameters or data (a concept called *workability*). Our goal of *reproducibility* thus encompasses both repeatability and workability. As computational experiments become ubiquitous in many scientific disciplines, there has been great interest in the publication of *reproducible papers* as well as of infrastructure that supports them [14, 6, 15, 19, 26, 3, 11, 17]. Unlike traditional papers which aim to describe ideas and results using text only, reproducible papers include data, the specification of computational processes and code used to derive the results. Motivated in part by cases of academic dishonesty as well as honest mistakes [9, 23, 24], some institutions have started to adopt reproducibility guidelines. For example, the ETH Zurich research ethics guidelines [8] require that all steps from input data to final figures need to be archived and made available upon request. Conferences, publishers and funding agencies are also *encouraging* authors to include reproducible experiments in their papers [17, 10]. In many ways, this is an extension of the very healthy demo-or-die philosophy that the database community follows for systems papers.

Science will greatly benefit as different communities start to follow these guidelines. Although in computer science, the publication of reproducible results and data sets is still in its infancy, in other fields there is already an established culture for doing so, see *e.g.,* [1, 11, 13]. When collections of computational experiments (along with their source code, raw data, and workflows) are documented, reproducible, and available in community-accessible repositories, new software can be built upon this verified base. This enables scientific advances that combine previous tools as well as ideas. For example, members of the community can search for related experiments (*e.g.,* "find experiments similar to mine") and better understand tools that have been created and how they are used. Furthermore, such repositories enable the community to evaluate the impact of a contribution not only through the citations to a paper, but also through the *use* of the proposed software and data components.

The database community has taken the lead in encouraging reproducibility for computational experiments [17, 16]. We can also take the lead in showing how to advance science by providing new techniques and tools for exploring the information in these repositories. Before discussing challenges of exploring repositories, let's take a look at the data: reproducible papers themselves.

## 2. REPRODUCIBLE EXPERIMENTS, PAPERS, AND REPOSITORIES

In reproducible papers, the results reported, including data, plots and visualizations are linked to the experiments and inputs. Having access to these, reviewers and readers can examine the results, then repeat or modify an execution. A number of ongoing efforts provide infrastructure and guidelines to make the production of such papers easier for authors [14, 10, 27, 15, 19]. Madagascar [15] is an open-source system for multi-dimensional data analysis that provides support for reproducible computational experiments. Authors describe their experiments in SCons, a rule-based language analogous to make. A reproducible publication can then be created by including the rules in a LaTeX document. Koop et al. [14] describe a provenance-based infrastructure that uses the VisTrails system [29] to support the life-cycle of publications: their creation, review and re-use. As scientists explore a given problem, VisTrails systematically captures the provenance of the exploration, including the workflows created and versions of source code and libraries used. The infrastructure also includes methods to link results to their provenance, reproduce results, explore parameter spaces, interact with results through a Web-based interface, and upgrade the specification of computational experiments to work in different environments and with newer versions of software. Documents (including LaTeX, PowerPoint, Word, wiki and HTML pages) can be created that link to provenance information that allows the results to be reproduced.[1] This year, the SIGMOD Repeatability effort has included extensive software infrastructure and guidelines to fa-

[1]Videos demonstrating this infrastructure in action are available at `http://www.vistrails.org/index.php/ExecutablePapers`.


*Proceedings of the VLDB Endowment,* Vol. 4, No. 12

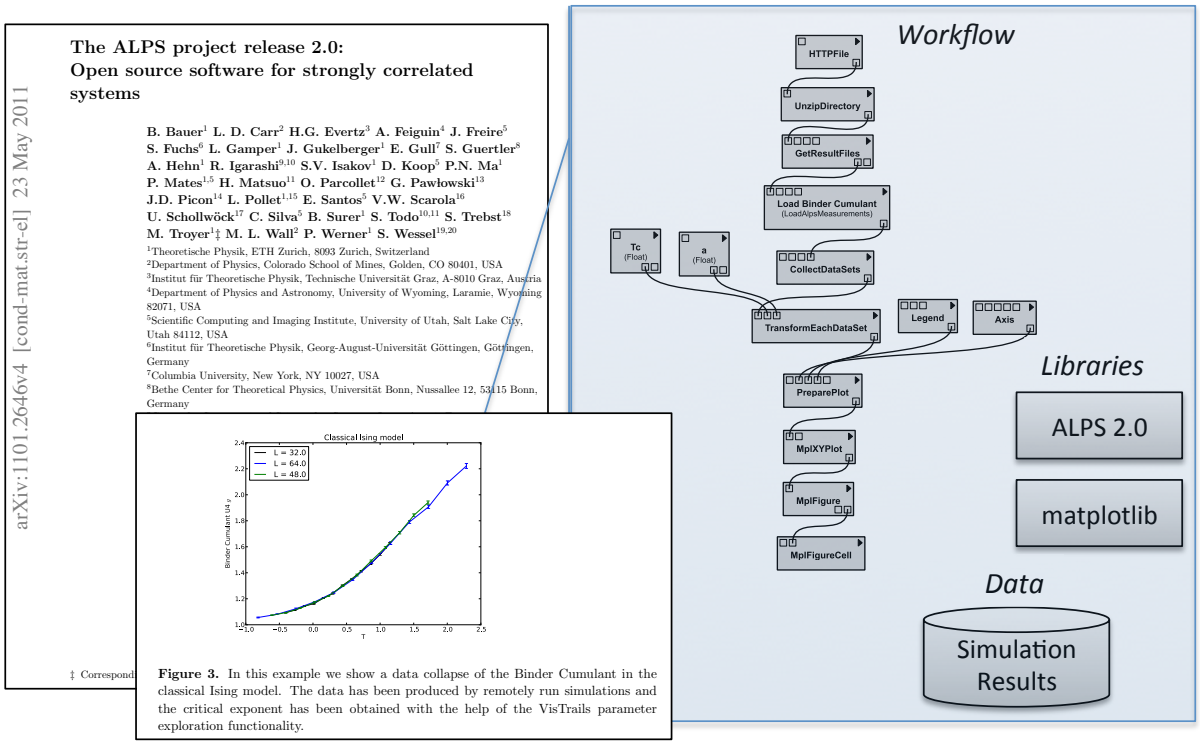


**Figure 3.** In this example we show a data collapse of the Binder Cumulant in the classical Ising model. The data has been produced by remotely run simulations and the critical exponent has been obtained with the help of the VisTrails parameter exploration functionality.

**Figure 1: An executable paper describing the ALPS 2.0 release. Figure 3 in this paper (center bottom) shows a plot which has a *deep caption* consisting of the workflow used to derive the plot, the underlying libraries invoked by the workflow, and the input data. This provenance information allows the plot to be reproduced. In the PDF version of the paper, this figure is active, and when clicked, the workflow is loaded into a workflow management system and executed on the reader's machine.**

cilitate the creation of reproducible software. Authors have used those facilities to archive their software, configuration files, and workflows. For the sake of concreteness, we give examples of the reproducible experiments along with the software.

## 2.1 Examples of Executable Papers

**Anatomy of a Reproducible Tuning Experiment** To show how a particular experimental graph was obtained, the experiment package includes the data used, the software that ran on that data, configuration parameters, and a workflow to run the software components in a particular order, perhaps with branching. It also includes a description of the hardware and operating system platform required (or a virtual machine that embodies those). Besides directly repeating the experiment, a "reader" of the paper can change the data, the configuration parameters, or the workflow. The data output of each step of the workflow can be examined or potentially be used as input to another software component (see [28] for details). That will be important in what we see below.

**Anatomy of a Reproducible WikiQuery Experiment** This experiment includes a series of workflows that were used to derive the experimental results reported in [22]. To run these workflows, readers may either copy and run experiments locally or run the experiment on the authors' machines and have the results shipped back.

**ALPS 2.0 and Physics Simulations** The ALPS project (Algorithms and Libraries for Physics Simulations) is an open-source initiative for the simulation of large quantum many body systems [2], which has been used in about two hundred research projects over the past six years. One of its core goals has been to simplify archival longevity and repeatability of simulations by standardizing input and result file formats. The paper describing the ALP2.0 [4], shown in Figure 1, is an example of a reproducible paper. It reports results from large-scale simulations that are time-consuming and run on high-performance hardware. The experiments are thus split into two parts: simulations and a set of analysis workflows. The simulation results are stored in (and made available from) an archival site, and the analysis workflows access the archival site and perform a sequence of analyses. The figures in the paper are active: clicking on a figure activates a "deep caption" which retrieves the workflow associate with the figure and executes the calculation leading to the figure on the user's machine. This paper makes use of the VisTrails publication infrastructure [14], which enables the linkage of results in a paper to their provenance.

## 2.2 Experiment and Workflow Repositories

With the growing awareness of the importance of reproducibility and sharing, several repositories have been created which cater to different aspects of this problem. nanoHUB [21] offers simulation tools which users can access from their web browsers in order to simulate nanotechnology devices. The creators of nanoHub claim that papers which make the simulation tools made available through their site enjoy a greater number of citations. Sites like crowdLabs [7, 18] and myExperiment [20] support the sharing of workflows which describe computational experiments, data analyses and visualizations. crowdlabs also supports a Web-based interface for executing workflows and displaying their results on a Web browser. PubZone (`http://www.pubzone.org`) is a new resource for the scientific community that provides a discussion forum and Wiki for publications. The idea for PubZone emerged as part of the initiative to ensure the reproducibility of experiments reported in SIGMOD papers. The results of such reproducibility experiments will be published in PubZone.

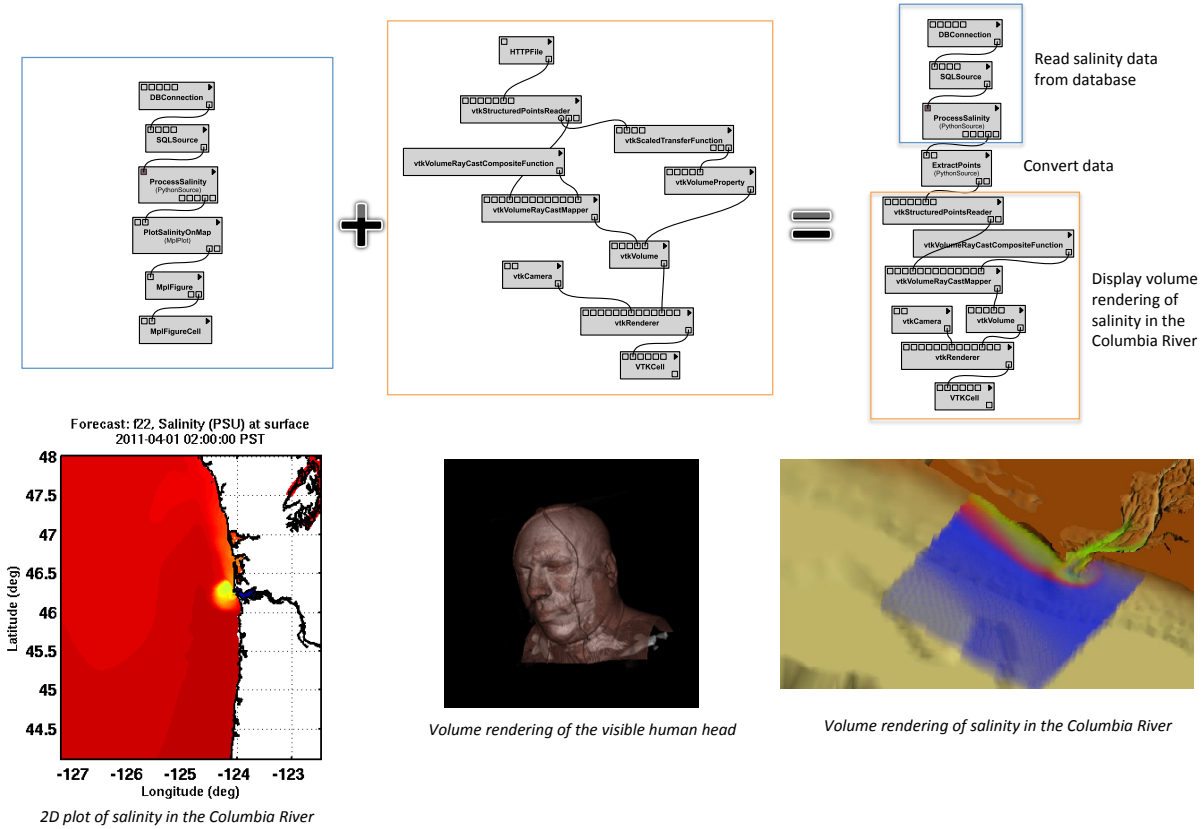


**Figure 2: A repository where scientists can publish their workflows and source code opens up new opportunities for knowledge sharing and re-use. While environmental scientists at the STC CMOP at OHSU use 2D plots to visualize the results of their simulations (left), at the SCI Institute, visualization experts are developing state-of-the-art volume rendering techniques (center). By combining their work, it is possible to generate more detailed (and insightful) 3D renderings of the CMOP simulation results (right).**

## 3. VISION

Many researchers have tools, software, and data they are willing to share in a widely available repository. To take best advantage of such contributions, we would like to propose a vision of "repository exploration" to help researchers re-use software as well as build new software components from existing ones. Here are some of the opportunities opened up by having an exploration platform and challenges involved in building such a system.

**1. How do I find tools/sofware/data that are helpful to me or related to my work?** Consider the following example. Find an experiment that uses MySQL as a back-end to store salinity information about the Columbia River and that performs volume rendering using an algorithm developed at the SCI Institute. This query spans the meta-data of the system configuration (for MySQL), the algorithm author (SCI Institute) and the data type (salinity information about a specific river). The repository querier may get very lucky and find an exact match, but will often find only a partial match. For example, the repository may have one entry that has a MySQL back-end and salinity information about a different river without volume rendering, and another that does volume rendering. This challenge entails the construction of an intuitive query interface that allows users to explore the data in the repository. Although there has been work on querying workflows [25, 5], experiments may or may not be specified as workflows. Even when they are, they can be specified at different levels of granularity. This creates new challenges for querying. In particular, techniques are needed that infer workflows from lower-level specifications (*e.g.,* scripts) as well as that support exploratory, vague queries. Furthermore, besides workflows, these repositories will contain other kinds of information, including source code, input and derived data. This requires a flexible query system that is able to cross the boundaries between structured and unstructured data perhaps in a similar way to querying on dataspace systems [12].

**2. Given several relevant entries from the repository, can they be combined into a coherent whole?** In the case of our running example, can we pipe the salinity information from one repository item into the volume rendering software of another? A workflow representation of each published computational experiment will be useful for this challenge, because a workflow exposes the specification of the computation tasks, including intermediate steps, in a structured fashion. As Figure 2 illustrates, one can imagine taking two workflows and creating a third one perhaps with extra steps for data conversion. To support this, techniques need to be developed to automatically find correspondences between computational modules (*e.g.,* module A performs a function that is similar to module B), as well as determine their compatibility (*e.g.,* can module A be connected to module B?).

**3. What is a good query language for finding repository items and assembling several repository items together?** Ideally, the query system would provide ways to query the different components of an experiment, including: meta-data about the data used; the structure as well as parameters in workflows, software headers, and system configuration parameters. Assembling different repository items together entails finding sub-parts of workflows that link together perhaps at some cost. A query processing system that incorporates such a cost measure to find the "best" answer to a query

would be most useful. Thus, the query language, if successful, would answer the first two challenges.

**4. Support "standing queries".** Once a consumer of the repository has identified a need, he or she can pose a query. If unsatisfied, the consumer can declare the query to be "standing" which means that new entries to the repository will match against the query to see whether they are more helpful.

**5. What is the "executable impact" of a given paper/result?** Given an executable paper A, go through other papers that use the components of A (directly or indirectly) and count them. To be most effective, this will tie into a visualization that shows a virtual collaboration graph to help answer questions like who uses (reuses) what; what are the most influential tools and results. Some mechanisms to support such an "executable impact" measure include: (i) the ability to capture the meta-data of a publication associated with an executable component, so the user of that component can cite the publication in an "executable bibliography"; and (ii) the ability to discover similarities of components in order to trace copyright rights.

# 4. USERS OF REPRODUCIBLE EXPERIMENT REPOSITORY EXPLORATION

Scientists would be our first target users. A base of validated workflow-described software will allow a kind of "workflow mashup" which, if combined with a capable query language, may enable the creation of a new targeted tool in days rather than years. But what's good for scientists will also help inventors and, through them, venture capitalists, as new products will be able to come online using the most advanced technology available. Repository Exploration will be a tool for the nimble.

**Acknowledgments.** This work has been partially funded by the National Science Foundation under grants IIS-1050422, IIS-0905385, IIS-0746500, CNS-0751152, N2010 IOB-0519985, N2010 DBI-0519984, IIS-0414763, DBI-0445666, DBI-0421604, DBI-0421604, and MCB-0209754.

# 5. REFERENCES

[1] The SAO/NASA Astrophysics Data System. http://adsabs.harvard.edu.

[2] The ALPS project. http://alps.comp-phys.org/.

[3] ICIAM Workshop on Reproducible Research: Tools and Strategies for Scientific Computing. http://www.mitacs.ca/events/index.php?option=com_content&view=article&id=214&Itemid=230&lang=en, 2011.

[4] B. Bauer et. al. The ALPS project release 2.0: Open source software for strongly correlated systems, Jan. 2011. Accepted for publication in Journal of Statistical Mechanics: Theory and Experiment. Paper available at http://arxiv.org/pdf/1101.2646 and workflows at http://arxiv.org/abs/1101.2646.

[5] C. Beeri, A. Eyal, S. Kamenkovich, and T. Milo. Querying business processes. In *VLDB*, pages 343–354, 2006.

[6] Beyond the PDF Workshop. https://sites.google.com/site/beyondthepdf, 2011.

[7] CrowdLabs. http://www.crowdlabs.org.

[8] Guidelines for Research Integrity and Good Scientific Practice at ETH Zurich. http://www.vpf.ethz.ch/services/researchethics/Broschure.

[9] ETH Zurich's head of research resigns. http://www.ethlife.ethz.ch/archive_articles/090921_Peter_Chen_Ruecktritt_MM/index_EN.

[10] The executable paper grand challenge, 2011. http://www.executablepapers.com.

[11] S. Fomel and J. Claerbout. Guest editors' introduction: Reproducible research. *Computing in Science Engineering*, 11(1):5 –7, 2009.

[12] M. J. Franklin, A. Y. Halevy, and D. Maier. From databases to dataspaces: a new abstraction for information management. *SIGMOD Record*, 34(4):27–33, 2005.

[13] Genbank. http://www.ncbi.nlm.nih.gov/genbank.

[14] D. Koop, E. Santos, P. Mates, H. Vo, P. Bonnet, B. Bauer, B. Surer, M. Troyer, D. Williams, J. Tohline, J. Freire, and C. Silva. A provenance-based infrastructure to support the life cycle of executable papers. In *Proceedings of the International Conference on Computational Science*, pages 648–657, 2011.

[15] Madagascar. http://www.reproducibility.org/wiki/Main_Page.

[16] S. Manegold, I. Manolescu, L. Afanasiev, J. Feng, G. Gou, M. Hadjieleftheriou, S. Harizopoulos, P. Kalnis, K. Karanasos, D. Laurent, M. Lupu, N. Onose, C. Ré, V. Sans, P. Senellart, T. Wu, and D. Shasha. Repeatability & workability evaluation of SIGMOD 2009. *SIGMOD Record*, 38(3):40–43, 2009.

[17] I. Manolescu, L. Afanasiev, A. Arion, J. Dittrich, S. Manegold, N. Polyzotis, K. Schnaitter, P. Senellart, S. Zoupanos, and D. Shasha. The repeatability experiment of SIGMOD 2008. *SIGMOD Record*, 37(1):39–45, 2008.

[18] P. Mates, E. Santos, J. Freire, and C. Silva. Crowdlabs: Social analysis and visualization for the sciences. In *Proceedings of SSDBM*, 2011. To appear.

[19] J. Mesirov. Accessible reproducible research. *Science*, 327(5964):415–416, 2010.

[20] myExperiment. http://www.myexperiment.org.

[21] nanoHub. http://nanohub.org.

[22] H. Nguyen, T. Nguyen, H. Nguyen, and J. Freire. Querying Wikipedia Documents and Relationships. In *Proceedings of WebDB*, 2010.

[23] Nobel laureate retracts two papers unrelated to her prize. http://www.nytimes.com/2010/09/24/science/24retraction.html?_r=1&emc=eta1, September 2010.

[24] It's science, but not necessarily right. http://www.nytimes.com/2011/06/26/opinion/sunday/26ideas.html?_r=2, June 2011.

[25] C. E. Scheidegger, H. T. Vo, D. Koop, J. Freire, and C. T. Silva. Querying and creating visualizations by analogy. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1560–1567, 2007.

[26] SIAM Mini-symposium on Verifiable, Reproducible Research and Computational Science. http://meetings.siam.org/sess/dsp_programsess.cfm?SESSIONCODE=11845.

[27] Repeatability section of the ACM SIGMOD 2011. http://www.sigmod2011.org/calls_papers_sigmod_research_repeatability.shtml.

[28] Computational repeatability: Tuning case study. http://effdas.itu.dk/repeatability/tuning.html.

[29] VisTrails. http://www.vistrails.org.