Workflows and Provenance: Toward Information Science Solutions for the Natural Sciences

Michael R. Gryk and Bertram Ludäscher

Abstract

The era of big data and ubiquitous computation has brought with it concerns about ensuring reproducibility in this new research environment. It is easy to assume that computational methods self-document by their very nature of being exact, deterministic processes. However, similar to laboratory experiments, ensuring reproducibility in the computational realm requires the documentation of both the protocols used (workflows), as well as a detailed description of the computational environment: algorithms, implementations, software environments, and the data ingested and execution logs of the computation. These two aspects of computational reproducibility (workflows and execution details) are discussed within the context of biomolecular Nuclear Magnetic Resonance spectroscopy (bioNMR), as well as the PRIMAD model for computational reproducibility.

Introduction to the Problem(s)

The era of big data is upon us. Along with it, computers and computation have become ubiquitous in almost every human endeavor. It should come as no surprise that concerns have been raised about the reproducibility of computational methods in research and science (Stodden et al., 2016). Reproducibility is a cornerstone of the scientific method, addressing both the universality of the reported scientific claims and providing transparency, such that the scientific results can be trusted. In general terms a process can be reproduced if both what was done and how it was done are sufficiently documented. It is often beneficial to record who conducted the process, as well as when it was done, but a truly reproducible process is independent of either of them. What then are the requirements for

LIBRARY TRENDS, Vol. 65, No. 4, 2017 ("Spanning the Information Sciences: A Celebration Marking Seventy Years of the Doctoral Program in the School of Information Sciences, University of Illinois at Urbana-Champaign," edited by Alistair Black and Emily J. M. Knox), pp. 555–562. © 2017 The Board of Trustees, University of Illinois

sufficient documentation, and how do those requirements translate to computation?

Method sections in the natural sciences typically have two components: the protocol used, and a detailed description of the reagents, equipment, and calibrations. It is easy to assume that computational methods self-document by their very nature of being exact, deterministic processes whose outcomes are dictated by "the program." However, leaving aside nondeterministic processes for the moment, by burying the computation within a software tool it can be very difficult to reproduce the exact process without a detailed record of the tools used, their configuration and execution. This problem is amplified with each software tool added to the process stream. As emphasized by Stodden et al. (2016, p. 1240): "We thus focus on the ability to rerun the same computational steps on the same data the original authors used as a minimum dissemination standard, which includes workflow information that explains what raw data and intermediate results are input to which computations."

Mirroring the situation with laboratory experiments, to ensure reproducibility in the computational realm requires the documentation of both the protocols used (workflows) and a precise description of the computational environment: algorithms, implementations, and software environments, as well as the data ingested and execution logs of the computation. These two aspects of computational reproducibility (workflows and execution details) are discussed within the context of biomolecular Nuclear Magnetic Resonance spectroscopy (bioNMR), as well as the PRIMAD model for computational reproducibility (Rauber et al., 2016).

Workflows and Provenance

A *workflow* is a model for complex processes in which the process is decomposed into discrete operations. Many formalized procedures fit this definition—culinary recipes, for instance—as well as administrative workflows, such as the checklists used to certify trustworthy digital repositories (Bak, 2016). The rationales for representing processes as workflows can be as varied as the processes themselves. The goal of recording a cooking recipe may be to assist in the organization and timing of the interdependent tasks, as well as to increase the reproducibility of the end product by applying uniform measurements of ingredients, cooking duration, and temperature. The goal of a TRAC checklist is to improve and document the quality of an archive, increasing its trustworthiness to the community it serves.

Scientific workflows are useful for a similarly diverse array of purposes. However, such workflows have an important distinguishing characteristic from business workflows. Typical business workflows focus on subprocesses and their validity within the context of an organizational infrastructure; for instance, documenting proper oversight and approval for a requisition request. As pointed out by Bowers and Ludäscher (2005), scientific com-

putational workflows are distinguished from administrative workflows by their focus on data and dataflow (see also Ludäscher, Weske, McPhillips, and Bowers [2009]). The emphasis on data has two important aspects: first, managing the timing of a workflow (a step in a workflow may not be able to begin until a preceding step has been completed); second, managing the semantic data types used within the workflow (for example, a process that requires oranges as input should not receive apples instead).

Apart from this simplistic description of the design and operation of scientific workflow systems, there are two significantly different perspectives from which one can consider the workflow itself. At a detailed or instance level, and in retrospective, it can be viewed as an execution log, capturing the exact sequence of events that occurred and their relationships with respect to one another (both in order and in data typing). Yet, at a more abstract level, the workflow is disjoint from any particular execution event.

There are many flavors of such an abstract workflow. It might still be detailed as in the former case, but neglecting specifics of the precise execution, as in computational step A of type α followed by computational step B of type β, without recording timestamps or execution details of the individual computations. In a more abstract case it might be a broad sketch of general processing chunks: ingestion, cleaning, transformation, visualization, and result reporting, with very little detail on the underlying computation at all. Finally, the workflow may represent the idea for a future process or protocol that has not yet been executed—or it may even be the case that applications for conducting the individual steps in the putative workflow do not yet exist. At this abstract level, the workflow is more similar to a cooking recipe and less like a stack trace.

An important distinction between the former and latter workflow types is not just the level of abstraction, but this consequence that an abstract workflow is capable of describing events that have not yet occurred. This is the inherent distinction between *retrospective provenance* (a representation of prior workflow execution) and *prospective provenance* (a description of how to execute a future workflow) (Lim, Lu, Chebotko, & Fotouhi, 2010). It can also be thought of as the distinction between what was done and what is intended to be done.

Despite many similarities, these two different workflow or provenance "worlds" are vastly different, both in their conceptualization of a workflow and with the underlying tools and approaches for managing workflows. Workflow-management systems designed to operate at the execution level concentrate on the details of tool operation and interoperability. Systems like Kepler (Altintas et al., 2004) and HTCondor (Thain, Tannenbaum, & Livny, 2005) must ensure that data of the correct type are being shuttled among individual actors (Kepler) or jobs (HTCondor). These tools may also manage the invocation and resource allocation of the individual jobs and check for completion and/or any errors.

Another approach to capturing retrospective provenance is that of

noWorkflow (Murta, Braganholo, Chirigati, Koop, & Freire, 2015). In this approach the provenance is recorded from the execution of a standard processing script (for example, Python), avoiding the learning curve and overhead of a specialized workflow system like HTCondor's. Prior to execution, noWorkflow parses the script and maps the dependencies among code blocks. On execution, noWorkflow relies upon built-in Python utilities to extract the provenance and map the dataflow.

These former methods all have in common an interest in the actual execution—describing an event that occurred. The notion of *workflow thinking* is to pattern a process as a workflow regardless of the manner of its execution or whether it has been executed at all. Workflow thinking is more about conceptualizing processes as recipes and protocols, structured as dataflow graphs with computational steps, and subsequently developing tools and approaches for formalizing, analyzing, and communicating these process descriptions. An important example of one approach is the YesWorkflow annotation and query system (McPhillips et al., 2015). YesWorkflow provides a few simple syntactical annotations that can be embedded within code or within a stand-alone file. These annotations describe the flow of data through the various processes, such that many of the aspects of a workflow can be visualized, queried, and understood in the absence of any execution events.

Reproducibility in Biomolecular NMR Spectroscopy: NMRbox

The National Center for NMR Data Processing and Analysis is a recent initiative to help foster reproducibility in the field of biomolecular NMR spectroscopy. The center has three overlapping research directions:

- First, the center is provisioning virtual machines (VMs) with most of the common software tools used by bio-NMR (Maciejewski et al., 2017). Provisioning VMs with the software helps ensure that both the software and underlying computing environment will persist into the future.
- Second, the center is modeling and capturing the metadata required to replicate the computational workflow of a bio-NMR study.
- Third, the center is providing Bayesian inference modules for consistent analysis of bio-NMR data.

The research developments and directions can be examined within the context of the PRIMAD model for computational reproducibility.

PRIMAD is an acronym for six key variables of a computational system that must be controlled for reproducibility. *Platform* (P) refers to the entire computational environment of the underlying software tool; this contains the computer hardware, as well as the operating system and any ancillary software components, such as shared libraries. *Research objective* (R) refers

to the scientific goal of the research: what hypothesis is being tested or what claim supported or refuted. *Implementation* (I) refers to the actual software code, by which a particular *method* (M) is being invoked. Method refers to the computational approach taken; for example, for ordering a list or pruning outliers. *Actors* (A) refers to the human agents who conduct the experiment. *Data* (D) refers to the datasets under analysis during the computational study. The report from the working group outlines this model within the context of a few examples of computation (bubble sorts and statistical analysis). The research endeavors of the Center for NMR Data Processing and Analysis will be examined within the context of this model for computational reproducibility.

Platform (P)

As discussed by Rauber et al. (2016), computational results that are independent on platform are considered to be portable, as well as reproducible. The field of bioNMR relies upon dozens of software tools, most of which were developed in academic labs and rely upon antiquated operating systems, compilers, and code libraries. A consequence of this is that most bioNMR studies are not portable. To address this issue, the Center for NMR Data Processing and Analysis is provisioning VMs' with all available bioNMR software; maintaining a cloud-based platform as a service model for accessing these VMs; and is in the process of establishing an archive of the various versions of the NMRbox VMs.

Research Objective (R)

Following the PRIMAD model, for a process to be considered reproducible, the research objective of the replicate process must be the same. This can be the most complicated barrier to reproducibility—for instance, if two research groups do not agree on the overall purpose of the research or if a subtle difference in the objective is not fully explained. While this is difficult to address computationally, it is being addressed by the administrative structure of the center. The research developments are driven by a so-called push-pull relationship, with external investigators conducting research on driving biological projects (DBPs). By focusing technological developments on established external research projects, these external DBPs will help ensure that the research objects are agreed on by the various biomedical communities.

Implementation (I)

In some sense it can be difficult to draw decisive lines among methods, algorithms, implementation, and source code. For the purposes of this case study, we will assume that methods/algorithms are in an abstract sense, as in the "bubble sort" versus "quick sort" (Rauber et al., 2016), while implementation and source code contain the possibility of performance tweaks

and or bugs and side effects. As such, an implementation would contain the various versions of it. This is also being addressed by NMRbox in that the many VM versions will also maintain a registry of the various software-tool versions contained within each. Therefore questions of reproducibility regarding a particular implementation of a method can be explored within the NMRbox VMs.

Method (M)

NMRbox aims to include 200 software packages used by the bioNMR community. There is a great deal of overlap in the functionality of this software smorgasbord; for instance, there are perhaps a dozen software tools capable of spectral reconstruction—the process of converting time-domain data to frequency domain. While there is a great deal of overlap among the various packages, there are methods and algorithms that are unique to a given tool; for instance, the maximum entropy reconstruction algorithm within the Rowland NMR toolkit. Maintaining all of the various software packages within one common VM aids in evaluating reproducibility, as the platform dependence inherent to any computational tool is eliminated.

Actors (A)

Within the context of the PRIMAD model, actors refer to human agents. Computational agents are considered to be a combination of methods and implementation (within the overall context of a platform). The role of actors in reproducibility is addressed in part by capturing annotations of human agents when performing manual analysis. An example of such an annotation strategy is that of the reproducibility extensions to the Sparky program (Fenwick, Hoch, Ulrich, & Gryk, 2015b). In this example, Sparky was augmented with a few routines that assist in version control of the assignment process using GIT and provide a helpful conceptual model for NMR peak assignment to assist in providing meaningful snapshots along the assignment process. Ongoing research at the Center for NMR Data Processing and Analysis will expand the set of captured metadata within a bioNMR study to further foster reproducibility.

Data (D)

The final variable for computational reproducibility is the datasets used in the study. Within the context of NMRbox, this is being addressed through the partnership with the BioMagResBank (BMRB) hosted by the University of Wisconsin (Ulrich et al., 2008). The BMRB has been the national repository for bioNMR data for the past several decades. Additional goals of the Center for NMR Data Processing and Analysis are to assist in research reproducibility by tracking additional data/metadata within the VM, and to assist the researcher in reporting this data/metadata to the BMRB by providing additional software tools. Thus the BMRB will have richer data

depositions, ensuring that all of the data required for reproducing a study are made available to the community at large.

Workflows and Provenance within NMRbox

Along with provisioning the standard bio-NMR software, NMRbox will also include utilities and resources to manage workflows and provenance. A workflow-management system for bioNMR spectral reconstruction has already been developed (Fenwick et al., 2015a). Called the CONNJUR Workflow Builder (CWB), the tool allows the NMR spectroscopist to craft a spectral-reconstruction process as a workflow utilizing any of three software tools: NMRPipe, the Rowland NMR Toolkit, and CONNJUR Spectrum Translator utilities. CWB stores the spectral metadata along with the reconstructions (workflow executions) within a MySQL database. Other workflow-management systems like HT Condor and YesWorkflow are also supported within NMRbox.

Conclusion

"Workflow thinking" can be a beneficial way of conceptualizing a computational process. By documenting the computational process as a workflow, the computation is more transparent and more easily reproduced. When combined with retrospective provenance information, additional value can be derived from a workflow (Pimentel et al., 2016). The PRIMAD model describes additional variables that can be controlled to investigate the universality of the computational process. The new Center for NMR Data Processing and Analysis, while predating the Dagstuhl working group (Rauber et al., 2016), provides a good case study for how these variables can be documented and controlled in the laboratories of natural scientists.

Acknowledgment

This work was supported in part by the National Institute of General Medical Sciences of the National Institutes of Health under award numbers GM-083072 and GM-111135.

References

Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludäscher, B., & Mock, S. (2004). Kepler: An extensible system for design and execution of scientific workflows. In *Proceedings of the 16th International Conference on Scientific and Statistical Database Management* (pp. 423–424). Los Alamitos, CA: IEEE Computer Society.

Bak, G. (2016). Trusted by whom? TDRs, standards culture and the nature of trust. *Archival Science, 16*(4), 373–402.

Bowers, S., & Ludäscher, B. (2005). Actor-oriented design of scientific workflows. In L. Delcambre, C. Kop, H. C. Mayr, J. Mylopoulos, & O. Pastor (Eds.), *Lecture Notes in Computer Science: Vol. 3716. Conceptual Modeling: ER 2005* (pp. 369–384). Berlin: Springer.

Fenwick, M., Weatherby, G., Vyas, J., Sesanker, C., Martyn, T. O., Ellis, H. J., & Gryk, M. R. (2015a). CONNJUR Workflow Builder: A software integration environment for spectral reconstruction. *Journal of Biomolecular NMR, 62*(3), 313–326.

Fenwick, M., Hoch, J. C., Ulrich, E., & Gryk, M. R. (2015b). CONNJUR R: An annotation strategy for fostering reproducibility in bio-NMR-protein spectral assignment. *Journal of Biomolecular NMR*, *63*(2), 141–150.

Lim, C., Lu, S., Chebotko, A., & Fotouhi, F. (2010). Prospective and retrospective provenance collection in scientific workflow environments. In *Proceedings 2010 IEEE Seventh International Conference on Services Computing* (pp. 449–456). Los Alamitos, CA: IEEE Computer Society.

Ludäscher, B., Weske, M., McPhillips, T., & Bowers, S. (2009). Scientific workflows: Business as usual? In U. Dayal, J. Eder, J. Koehler, & H. A. Reijers (Eds.), *Lecture Notes in Computer Science: Vol. 5701. Business Process Management: BPM 2009* (pp. 31–47). Berlin: Springer.

Maciejewski, M. W., Schuyler, A. D., Gryk, M. R., Moraru, I. I., Romero, P. R., Ulrich, E. L., et al. (2017). NMRbox: A resource for biomolecular NMR computation. *Biophysical Journal*, *112*(8), 1529–1534.

McPhillips, T., Song, T., Kolisnik, T., Aulenbach, S., Belhajjame, K., Bocinsky, R. K., et al. (2015). YesWorkflow: A user-oriented, language-independent tool for recovering workflow information from scripts. *International Journal of Digital Curation*, *10*(1), 298–313.

Murta, L., Braganholo, V., Chirigati, F., Koop, D., & Freire, J. (2015). noWorkflow: Capturing and analyzing provenance of scripts. In B. Ludäscher & B. Plale (Eds.), *Lecture Notes in Computer Science: Vol. 8628. Provenance and Annotation of Data and Processes: IPAW 2014* (pp. 71–83). Berlin: Springer.

Pimentel, J. F., Dey, S., McPhillips, T., Belhajjame, K., Koop, D., Murta, L., et al. (2016). Yin & yang: Demonstrating complementary provenance from noWorkflow & YesWorkflow. In M. Mattoso & B. Glavic (Eds.), *Lecture Notes in Computer Science: Vol. 9672. Provenance and Annotation of Data and Processes: IPAW 2016* (pp. 161–165). Berlin: Springer.

Rauber, A., Braganholo, V., Dittrich, J., Ferro, N., Freire, J., Fuhr, N., et al. (2016). PRIMAD—Information gained by different types of reproducibility. In J. Freire, N. Fuhr, & A. Rauber (Eds.), *Reproducibility of data-oriented experiments in e-science* (pp. 128–132). Report from Dagstuhl seminar 16041. Saarbrücken, Germany: Dagstuhl.

Stodden, V., McNutt, M., Bailey, D. H., Deelman, E., Gil, Y., Hanson, B., et al. (2016). Enhancing reproducibility for computational methods. *Science*, *354*(6317), 1240–1241.

Thain, D., Tannenbaum, T., & Livny, M. (2005). Distributed computing in practice: The condor experience. *Concurrency and Computation: Practice and Experience*, *17*(2–4), 323–356.

Ulrich, E. L., Akutsu, H., Doreleijers, J. F., Harano, Y., Ioannidis, Y. E., Lin, J., et al. (2008). BioMagResBank. *Nucleic Acids Research*, *36*(suppl. 1), D402–D408.

Michael R. Gryk is an associate professor of molecular biology and biophysics at UCONN Health, the medical school of the University of Connecticut. He is also a doctoral student of library and information science in the School of Information Sciences, University of Illinois at Urbana-Champaign. He has worked in the field of structural biology, concentrating on bioNMR, since 1990. His research interests are in the computational and informational science aspects of biomedical research. He received his MS in chemistry from the University of Connecticut, and his doctorate in biophysics from Stanford University.

Bertram Ludäscher is a professor at the School of Information Sciences, University of Illinois at Urbana-Champaign. He directs the Center for Informatics Research in Science and Scholarship (CIRSS), and is a faculty affiliate with both the National Center for Supercomputing Applications (NCSA) and the Department of Computer Science. From 2004 to 2014 he was a tenured professor in the Department of Computer Science at the University of California, Davis. His research interests range from scientific data and workflow management to knowledge representation and reasoning. Until 2004 he was a research scientist at the San Diego Supercomputer Center and an adjunct faculty member in the Department of Computer Science and Engineering at the University of California, San Diego. He received his MS (Dipl.-Inform.) in computer science from the University of Karlsruhe, and his doctorate from the University of Freiburg.