POSTER: RDataTracker and DDG Explorer

Capture, Visualization and Querying of Provenance from R Scripts

Barbara S. Lerner[1] and Emery R. Boose[2]

[1] Mount Holyoke College, South Hadley, Massachusetts 01075, USA
[2] Harvard Forest, Harvard University, Petersham, Massachusetts 01366, USA

Scientific data provenance is gaining interest among both scientists and computer scientists. The current state of the art of provenance capture requires scientists to adopt new technologies, most commonly workflow systems such as Kepler [BML+06], Vistrails [SKS+08] or Taverna [MBZ+08], among others. While there are likely additional benefits to adopting these systems, they present a hurdle to scientists who are more interested in focusing on science than in learning new technologies. The work described in this poster is aimed at exploring the extent to which we can support scientists while expecting a minimal investment in terms of additional effort on their part.

This work has been developed in collaboration with ecologists at Harvard Forest, a 3500 acre facility operated by Harvard University and serving as a Long-Term Ecological Research (LTER) site funded by the National Science Foundation. Many of these ecologists perform data analysis using R, a widely used scripting language that includes extensive statistical analysis and plotting functionality. These scientists are committed to understanding their data, making sure that their data analyses are done in an appropriate manner, and sharing their data and results with others. For these reasons, they appreciate the value that collecting data provenance may have, but they are not enthusiastic about learning new tools. In this poster, we present two tools aimed at this audience: RDataTracker and DDG Explorer. RDataTracker [LB14] is used to collect data provenance during the execution of an R script. DDG Explorer is the tool that is used to examine and query the resulting data provenance.

1 Capturing Data Provenance with RDataTracker

RDataTracker is an R library that contains functions to build a provenance graph based on the execution of an R script and/or user activity in the R console. At a minimum, the scientist needs to load the library, initialize the provenance graph at the start of execution, and save the provenance graph at the end. As a script executes or the user enters commands at the console, a provenance graph is constructed that records the operations that are executed, the data that are used, and where variables are assigned.

The user can increase the amount of information collected during execution by including more instrumentation. In particular, by doing this the user can:

- Save copies of input and output files as well as copies of plots created.

- Include details of provenance that occurs within the execution of functions.
- Introduce levels of abstraction that allow the provenance graph to be viewed with a varying amount of detail.
- Checkpoint the entire R state and restore it later, capturing the checkpoint and restore operations in the provenance graph so that the data derivation links correctly show the effects of the checkpoint and restore operations and files are restored to the contents they had at the time of the checkpoint.
- Capture error messages generated by the R interpreter or RDataTracker and include them in the provenance graph as an aid to debugging.

This work differs from CXXR [SR10,RS12], an implementation of the R interpreter that includes automated provenance collection. In CXXR, the data provenance is made available to the programmer via functions within the R session, but there is no provenance recorded within functions and the data provenance is not stored persistently.

2 Viewing and Querying Provenance with DDG Explorer

DDG Explorer is a tool that supports the querying and visualization of the provenance graphs created by RDataTracker. DDG Explorer has been carefully designed to be language agnostic and also supports the display and querying of provenance graphs created from the execution of Little-JIL processes [OCE+10,LBO+11]. This poster focuses on provenance collected in R.

With DDG Explorer, the user can load a provenance graph written by RDataTracker. In addition to the usual navigation and querying facilities provided by provenance browsers, DDG Explorer takes advantage of the abstraction and checkpoint/restore features of RDataTracker to provide additional navigation capabilities. The levels of abstraction captured in RDataTracker allow for sections of the full provenance graph to be collapsed to an individual node. This allows for navigation at a high level of abstraction. By clicking on a collapsed node, the node is expanded to expose more detail.

DDG Explorer also uses checkpoint/restore information to hide detail and selectively expose it. In particular, the provenance that occurs between a checkpoint and when that checkpoint is restored is collapsed to a single node. By clicking on the collapsed node, the user can see the details of the activity that occurred between the checkpoint and restore.

The normal mode of operation that we expect is for users to write and execute their scripts, examine the resulting provenance graphs and use the information to refine the scripts, iterating until the script behaves as expected and the provenance graph contains the desired amount of detail. At that point, the user can save the provenance graph and associated files to a database.

3 Conclusion

RDataTracker and DDG Explorer support the collection of data provenance from the execution of R scripts and R console commands. Our goal is to provide tools

that are easy for scientists to learn and that offer an immediate payback for a small effort and increasing value as the scientist becomes familiar with the tools and invests more effort in their use. Initial results have been encouraging and we will continue to improve upon the types of information that we capture and to reduce the effort required by scientists.

Acknowledgments

The authors acknowledge intellectual contributions from collaborators Leon Osterweil and Aaron Ellison and Harvard Forest REU students Sophia Taskova, Antonia Oprescu, and Shaylyn Adams. The work was supported by NSF grants DEB-0620443, DEB-1237491, and DBI-1003938, the Charles Bullard Fellowship at Harvard University, and a faculty fellowship from Mount Holyoke College and is a contribution from the Harvard Forest Long-Term Ecological Research (LTER) program.

References

[BML+06] Shawn Bowers, Timothy McPhillips, Bertram Ludäscher, Shirley Cohen, and Susan B. Davidson. A model for user-oriented data provenance in pipelined scientific workflows. In *IPAW 2006*, pages 133–147, Chicago IL, May 2006.

[LB14] Barbara S. Lerner and Emery R. Boose. RDataTracker: Collecting provenance in an interactive scripting environment. In *TAPP 2014*, Cologne, Germany, June 2014.

[LBO+11] Barbara Lerner, Emery Boose, Leon J. Osterweil, Aaron M. Ellison, and Lori A. Clarke. Provenance and quality control in sensor networks. In *Proc. of the Environmental Information Management (EIM) 2011 Conf.*, Santa Barbara, California, September 2011.

[MBZ+08] Paolo Missier, Khalid Belhajjame, Jun Zhao, Marco Roos, and Carole Goble. Data lineage model for Taverna workflows with lightweight annotation requirements. In *IPAW 2008*, pages 17–30, Salt Lake City, Utah, June 2008.

[OCE+10] Leon J. Osterweil, Lori A. Clarke, Aaron. M. Ellison, Emery R. Boose, Rodion Podorozhny, and Alexander Wise. Clear and precise specification of ecological data management processes and dataset provenance. *IEEE Trans. on Automation Science and Engineering*, 7(1):189–195, 2010.

[RS12] Andrew Runnalls and Chris Silles. Provenance tracking in R. In *IPAW 2012*, pages 237–239, Berlin, 2012.

[SKS+08] Carlos Scheidegger, David Koop, Emanuele Santos, Huy Vo, Steven Callahan, Juliana Freire, and Cláudio Silva. Tackling the provenance challenge one layer at a time. *Concurrency and Computation: Practice and Experience*, 20(5):473–483, 2008.

[SR10] Chris A. Silles and Andrew R. Runnalls. Provenance-awareness in R. In *IPAW 2010*, pages 64–72, 2010.