Research and Software Development Projects for the Data Provenance Toolbox

Matthew K. Lau, PhD.

Harvard Forest

Data Provenance for Genomics Pipelines

- Genomics pipelines process large amounts of often noisy data using multiple software packages, which could be aided by data provenance methods that would make their processes, resulting datasets and associated errors transparent. Of particular need, is a means to track and view data processes and analyses from the multiple languages comprising these diverse, "Rube Goldberg"-esque, pipelines spread across more than one programming language.
- The RDataTracker software package designed to document data handling performed by R scripts, has a simple implementation framework and graphical output, which, because it is written in R, an open source programming language with native capabilities for communicating with other programming languages (including python, c++, java and bash), is naturally poised to incorporate data provenance from other languages and other provenance tracking systems.
- in this project, we will extend the capabilities of the RDataTracker package to incorporate provenance information from other parts of data pipelines not written in the r language, and will begin by exploring the following possible goals:

1. incorporate data provenance information from the prov package written in python
2. integrate this new information stream into the existing DDG graphical output
3. test these methods with a real genomics data pipeline

- An integrated framework will provide a large community of coders creating data pipelines with a sophisticated tool for tracking their data for the purpose of data documentation and code improvement. In addition, this project will pave the way for expanding data provenance capabilities toward the goal of tracking multiple streams of information from different sources across different software languages.
- Possible products from this project include:

1. Presentations at ecological, genetics and CS conferences
2. A Methods in Ecology and Evolution (MEE) paper
3. Applications papers

- Possible challenges:

1. Genomics pipelines work on large datasets which often require working on servers and parallelization, which could both present a challenge for using the RDataTracker software, which relies on the .Rhistory files generated by R scripts. Expanding the possible recording and tracking methods might address this and would also make RDataTracker more robust.
2. Some packages used by these pipelines may be proprietary or use proprietary data formats. This could possibly be addressed by collaborating with these software companies to make their data formats open-source.

Native viewing of DDG within R

- The RDataTracker generated DDGs can be viewed with the Java based DDG Viewer; however, some users may not have access to either the most current version of Java or Java at all.
- Adding DDG based on network visualization methods in R provide an alternative, Java independent, means to visualize DDG that could be scripted, automated and written natively in R.
- To accomplish this, we can explore extant packages within R that provide network plot functions and interactive graphing capabilities, including:
- http://bioconductor.org/packages/release/bioc/html/Rgraphviz.html
- http://christophergandrud.github.io/networkD3/
- https://plot.ly/r/
- http://leafletjs.com
- http://rcharts.io

Graph Theory Based Identification of Code Crux Points

- R scripts tend to produce DDG that tend to be graphs with high directionality and little feedback; and they are often large enough to make it difficult to identify important nodes in the code
- Graph and network theoretic analysis are often applied to complex systems of inter-elated components (e.g. food webs, the Internet, transportation systems)
- Adding such analyses to the RDataTracker package and DDG Explorer would provide additional means for users to explore their scripts in a more efficient and informative way
- Several possible avenues to explore would be:

1. Weighted graph statistics to measure node importance
2. Module and cluster identification algorithms to identify groups of nodes
3. Flow analysis statistics from Ecosystem Network Analysis theory to measure node and pathway importance

- The https://github.com/SEELab/enaR provides many of these tools and could be integrated into or adapted to work with RDataTracker DDG