Mihir Samdarshi

March 1, 2019

BIOL 588 – Advanced Systems Biology Research

**Refactor of code to the Model-View-Controller paradigm improves** **GRNsight: a web application for visualizing small- to medium-scale gene regulatory networks**

Introduction

The central model of molecular biology describes how the flow of information in a cell during gene expression goes from DNA to RNA to protein. Transcription factors control gene expression by binding to regulatory DNA sequences. Activators increase gene expression. Repressors decrease gene expression. Transcription factors are themselves proteins encoded by genes. A gene regulatory network (GRN) consists of genes, transcription factors, and the regulatory connections between them, which govern the level of expression of mRNA and proteins from those genes. These gene regulatory networks can be represented in a graph format, with nodes and edges; each node in a graph represents a regulatory transcription factor, with each edge representing a regulatory relationship, either an activation or repression relationship.

The Dahlquist lab has been focused on the early cold shock transcriptional response in budding yeast. The lab has been continually screening various deletion strains for impaired growth, and using data derived from the transcriptional responses to identify further genes to study. This has led to the identification and creation of gene regulatory networks that have been determined to be important to the regulation of this particular type of environmental stress response. As a systems biology laboratory defined as a laboratory which studies, “the structure and dynamics of cellular and organismal function, rather than the characteristics of isolated parts of a cell or organism,” it is important to be able to study cellular pathways, such as cold shock response in a eukaryotic model organism, as a whole. One of these ways that a systems biology laboratory can do this is by mathematically modelling these pathways, especially so as to predict the response of these pathways from different extracellular or intracellular inputs. Thus, one of the activities that the Dahlquist lab engages in is mathematically modelling the variation of gene expression over time in yeast. GRNmap is a sister project of GRNsight that focuses on this project.

GRNmap is a differential equations model of the changes in gene expression over time for a gene regulatory network (Dahlquist et al. 2015). Each gene (node) in the network has an equation. The parameters in the model are estimated from laboratory data. The weight parameter, w, gives the direction (activation or repression) and magnitude of the regulatory relationship. GRNmap produces an Excel spreadsheet with an adjacency matrix representing the network. 0 represents no relationship. Initially, 1 represents a regulatory relationship where the gene specified by the column controls the gene specified by the rows. After the parameters have been estimated, a positive weight value indicates activation and a negative weight value indicates repression. The magnitude of the weight defines the strength of the relationship. However, GRNmap does not generate a visual representation of GRNs. Thus, a need for a visualization tool arises.

GRNsight was developed after the Dahlquist lab observed that commonly used graphing software for similar adjacency matrices was primarily optimized for large data sets. Software like Cytoscape (Shannon et al., 2003) and Gephi (Bastian et al., 2009) are common software packages that allow for the visualization of networks. However, these tools are most effective in organizing and displaying large data sets (larger than 100 nodes). This means that the emphasis of the visualization is on important or central nodes. This is especially disappointing for the potential applications of these graph applications in the Dahlquist lab, where the primary interest lies in the relationships between the nodes (or genes) rather than the nodes themselves. Furthermore, Cytoscape is not an open source project; this means that there is no obligation for developers to listen to their primary audience, no opportunity for contributions to the project, and the software cannot be built upon. Gephi, on the other hand is not targeted towards biological applications and thus lack visualization features and ease of use that would make it a valuable addition to a dynamic systems biology laboratory like the Dahlquist lab.

In order to develop a successful visualization tool, the Dahlquist lab determined that it was essential that GRNsight contain features common to any successful visualization tool for biological pathways. These features are best described by Pavlopoulos et al. (2015) and Saraiya et al. (2005). One of the first requirements is that pathways must be automatically constructed, and not in a static representation; for example, this would mean that a pathway constructed in an Excel sheet is not a good representation for visualizing a pathway. Other requirements included being able to gain context for the pathway within the application. Additionally, they stated that the visualizations should be able to provide “information attributes that visualizations should reveal through their visual representations” (Saraiya et al., 2005) Detailing further, Gostner et al. (2015) state that the aesthetic appearance of the model must promote the “the readability and maintainability of models.” Finally, many of the reviews of current systems biology visualization tools state that the visualization tool must promote sharing of the data and results throughout the scientific community; this includes features like data interoperability, visual sharing, and ability for collaboration.

GRNsight is optimized for visualizing the relationships in small- to medium-scale gene regulatory networks. GRNsight uses the Data-Driven Documents (D3) JavaScript library to generate a graph derived from input network data. D3 dynamically manipulates HTML and Scalable Vector Graphics (SVG) to form the elements of the graph. GRNsight implements D3’s force layout algorithm which applies a physics-based simulation to the graph; users are able to modify the physics of this simulation; GRNsight then allows users to easily modulate nodes or lay them out automatically in an alphabetized grid. Nodes also display time course data sets to allow users to better understand the significance of the overall edge relationships over time. Users can also direct attention to certain edge weights by setting a threshold to color edges gray. Comparison of different graphs can also be facilitated by setting a factor against which graph data can be normalized against. GRNsight also allows users to import adjacency matrices in a variety of commonly used formats such as SIF and GraphML. Such features have enabled GRNsight to become an essential utility in the data processing pathway utilized by the Dahlquist lab.

GRNsight has been built and revised to its current version over the course of five years. The initial development began in 2014, and since then has had around 8 different developers contribute towards its codebase. GRNsight has consistently tried to follow the Test-Driven-Development best practice that encourages developers to write tests for any potential changes before actually making those changes (Bissi, 2016). However, even given this best practice, the codebase has gotten quite haphazard having changed so many hands and developed fairly rapidly by undergraduate students. This has led to detrimental effects, such as a rapidly increasing number of bugs in the software, correlated with the increasing number of features, along with the codebase equally not as user-friendly.

A central state register was determined to be the best way to resolve the core issues GRNsight was having, namely an increasing number of bugs and vastly disorganized code. It was determined that the Model-View-Controller (MVC) software architectural pattern was the best implementation of this system. MVC was first described to the general public by Krasner & Pope (1988) as a pattern for implementation of user interfaces in 1988. After the rapid rise of web applications in the dawn of the internet, MVC quickly became the software architecture of choice for most use-cases. Leff and Rayfield (2001) described the implementation of a basic web application using this paradigm. It called for the separation of the central state of the application (the Model) from the implementation of the code that was the basis of what the user saw (View) from the code that handled any particular interactions the user could have with the web application (Controller).

This paper will describe the major revisions that GRNsight has gone through since its last implementation. This includes the addition of new features to make GRNsight more accessible to users, allow for better sharing of graphs, and increase visualization options. Furthermore, I will also describe the implementation of the new MVC architecture for GRNsight and its significance to how GRNsight operates today. I will show that GRNsight is a powerful tool that accomplishes its task of visualizing small- to medium-scale gene regulatory networks in a lightweight, easy-to-use, and feature-rich manner, all while implementing industry-standard best-practice software development practices.

Works Cited

Bastian, M., Heymann, S., & Jacomy, M. (2009, March). Gephi: an open source software for exploring and manipulating networks. *In Third international AAAI conference on weblogs and social media*.

Dahlquist, K. D., Dionisio, J. D. N., Fitzpatrick, B. G., Anguiano, N. A., Varshneya, A., Southwick, B. J., & Samdarshi, M. (2016). GRNsight: a web application and service for visualizing models of small-to medium-scale gene regulatory networks. PeerJ Computer Science, 2, e85. doi: 10.7717/peerj-cs.85

Gostner, R., Baldacci, B., Morine, M. J., & Priami, C. (2015). Graphical modeling tools for systems biology. *ACM Computing Surveys (CSUR)*, *47*(2), 16. doi: 10.1145/2633461

Krasner, G. E., & Pope, S. T. (1988). A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object oriented programming*, *1*(3), 26-49.

Leff, A., & Rayfield, J. T. (2001). Web-application development using the model/view/controller design pattern. In *Proceedings fifth ieee international enterprise distributed object computing conference* (pp. 118-127). IEEE.

Pavlopoulos, G. A., Wegener, A. L., & Schneider, R. (2008). A survey of visualization tools for biological network analysis. *Biodata mining*, 1(1), 12. doi: 10.1186/1756-0381-1-12

Saraiya, P., North, C., & Duca, K. (2005). Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. *Information Visualization*, *4*(3), 191-205. doi: 10.1057/palgrave.ivs.9500102

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., ... & Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, *13*(11), 2498-2504. doi: 10.1101/gr.1239303