

GRNmap and GRNsight: open source software for dynamical systems modeling and visualization of medium-scale gene regulatory networks

Kam D. Dahlquist¹, Ben G. Fitzpatrick², John David N. Dionisio³, Nicole A. Anguiano^{1,3}, Juan S. Carrillo^{2,3}, Trixie Anne M. Roque², Anindita Varshneya^{1,3}, Mihir Samdarshi¹, and Chukwuemeka Azinge³

¹Department of Biology, ²Department of Mathematics, ³Department of Electrical Engineering & Computer Science, Loyola Marymount University, 1 LMU Drive, Los Angeles, CA 90045 USA

Email: kdahlquist@lmu.edu

Project Websites: <http://kdahlquist.github.io/GRNmap/> and <http://dondi.github.io/GRNsight/>

Source Code: <https://github.com/kdahlquist/GRNmap/> and <https://github.com/dondi/GRNsight>

License: BSD license, <https://github.com/kdahlquist/GRNmap/blob/master/LICENSE> and <https://github.com/dondi/GRNsight/blob/master/LICENSE.txt>

A gene regulatory network (GRN) consists of genes, transcription factors, and the regulatory connections between them that govern the level of expression of mRNA and proteins from those genes. Over a period of several years, our group has developed a MATLAB software package, called GRNmap, that uses ordinary differential equations to model the dynamics of medium-scale GRNs. The program uses a penalized least squares approach (Dahlquist et al. 2015, <https://doi.org/10.1007/s11538-015-0092-6>) to estimate production rates, expression thresholds, and regulatory weights for each transcription factor in the network based on gene expression data, and then performs a forward simulation of the dynamics of the network. GRNmap has options for using a sigmoidal or Michaelis-Menten production function. Although originally developed for yeast data, the model is broadly applicable to any species. The large number of developers and time span of development led to a code base that was difficult to revise and adjust. We therefore brought the code under version control in a GitHub repository and refactored the script-based software with global variables into a function-based package that uses an object to carry relevant information from function to function. This modular approach allows for cleaner, less ambiguous code and increased maintainability. We standardized the format of the input and output Excel workbooks, adding an optimization diagnostics output worksheet which includes both the actual and theoretical minimum least squared error overall, and the mean squared errors for the individual genes. The MATLAB compiler was used to create an executable that can be run on any Windows machine without the need of a MATLAB license. Finally, we have implemented test-driven development for new features, and are improving the test coverage of previous code.

GRNsight is an open source web application for visualizing such models of gene regulatory networks (Dahlquist et al. 2016, <https://doi.org/10.7287/peerj.preprints.2068v1>). GRNsight accepts GRNmap- or user-generated spreadsheets containing an adjacency matrix representation of the GRN and automatically lays out the graph of the GRN model. It is written in JavaScript, with diagrams facilitated by D3.js. Node.js and the Express framework handle server-side functions. GRNsight's diagrams are based on D3.js's force graph layout algorithm, which was then extensively customized. GRNsight uses pointed and blunt arrowheads, and colors the edges and adjusts their thicknesses based on the sign (activation or repression) and magnitude of the GRNmap weight parameter. Visualizations can be modified through manual node dragging and sliders that adjust the force graph parameters. From the early stages, GRNsight has had a unit testing framework using Mocha and the Chai assertion library to perform test-driven development where unit tests are written before new functionality is coded. This framework consists of over 160 automated unit tests that examine over 450 test files to ensure that the program is running as expected. Error and warning messages inform the user what happened, the source of the problem, and possible solutions. Together, the life cycle of these two programs illustrates the differences between the cultures of mathematics and computing, the challenges and benefits of bringing an existing code base up to open development standards (GRNmap), and the advantages of starting a project using best practices from the beginning (GRNsight). Our goal is to facilitate reproducible research.