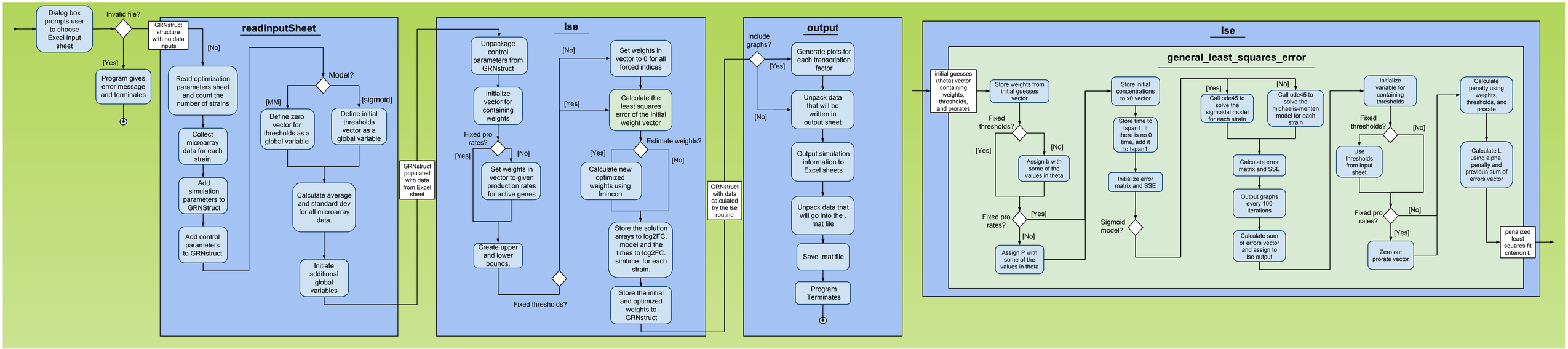
Test-Driven Development and Functionality Improvements to GRNmap, a Gene Regulatory Network Modeling Application

Trixie Anne Roque[1], Kam D. Dahlquist[2], Ben G. Fitzpatrick[3] , and Dr. John David N. Dionisio[1]

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

LMU LA McNair Scholars Program

GRNmap Activity Diagram



Abstract

A gene regulatory network (GRN) consists of a set of transcription factors that regulate the level of gene expression encoding other transcription factors. The dynamics of a GRN describe how gene expression in the network changes over time. GRNmap is a complex MATLAB software package that uses ordinary differential equations to model the dynamics of medium-scale GRNs from budding yeast, Saccharomyces cerevisiae. The program estimates production rates, expression thresholds, and regulatory weights for each transcription factor in the network based on DNA microarray data, using forward simulations of model dynamics. Since v1.0, we have made design changes, added new features, fixed bugs, implemented a testing framework, and created documentation. For example, GRNmap now accepts and outputs Excel worksheets with more descriptive names, computes the standard deviations of the log2 expression data, and outputs an optimization diagnostics sheet which includes both actual and theoretical minimum least squares errors. We have also designed sixteen manual input sheet tests to uncover and fix bugs as model and algorithm development progresses. We incorporated these tests into an automated testing framework that will speed debugging and prevent future code regressions. We have added documentation to our website and wiki and constructed a UML activity diagram to document the program's overall flow and how each function processes information. The source code and executable (which contain demo files and can run without a MATLAB license) for the updated version 1.2 are available for download at http://kdahlquist.github.io/GRNmap/ under the BSD open source license.

GRNmap Functionality

GRNmap takes in its parameters directly from an Excel file. The file is expected to have the following worksheets:

- Production rates - Initial guess
- Degradation rates - Provided by user from data
- Expression Thresholds - Initial guess
- Microarray data - log 2 fold change of expression
- Adjacency matrix to describe the graph for the GRN
- Initial guess for the network weights
- Simulation times
- Optimization parameters, including which model to use, whether or not to perform a forward simulation, whether or not to set certain parameters, and whether or not to include plots for the genes

GRNmap estimates the production rates, weights, and expression thresholds. It then performs a forward simulation using these three parameters so that model-generated expression data can be compared to the experimental data input to the model.

GRNmap then outputs an Excel file based on user preferences from the optimization parameters sheet. The file will contain the original input sheets and the following sheets:

- Optimized log2 expression for each strain - Based off the microarray data
- Standard deviations of concentrations of each strain
- Optimized production rates - Only if the user chose to estimate them
- Optimized thresholds - Only if the user chose to estimate them
- Optimized network weights
- Diagnostics - Indicates how many iterations were made, what the minimum and the actual least squares error are, and what the penalty value is.

Additionally, a MAT file containing the calculated values, and plots corresponding to each gene in the network showing gene expression over time are outputted based on user preference in the optimization parameters sheet.

Unit-Testing Framework

- In order to maintain the MATLAB package efficiently and effectively, a unit testing framework is created. Unit testing, a method for checking whether individual units of a software program function as intended, ensures that every change we make to the code does not affect its functionality.
- Unit testing allows for better means of pinpointing specific problems in the program by breaking up its operations into smaller parts and then inputting values to which we know the answer
- We have created sixteen manual input test Excel worksheets each with different optimization parameters in order to exhaustively check that the outputs for all combinations of parameters function properly.
- For adding functionality, the standard process of test-driven development is used as shown in Figure 1.
- For debugging, the process starts with analyzing the existing code and creating a unit test for one of its functions. We expect the test to pass if the code does not have an error. If it does, the unit test shows what the error is for easier editing.

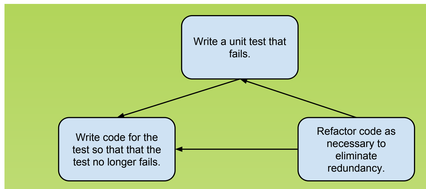


Figure 1 . Process of test-driven development

```
DEFINE main function
    CALL functiontests (localfunctions) to make a tests array
END

DEFINE function firstTest (testCase)
    actualOutput = evaluate function by using known inputs
    expectedOutput = assign expected results
    VERIFY actualOutput equals expectedOutput
END
```

Figure 2 . Pseudocode for Unit Testing Framework

Future Work

- Making GRNmap more accessible by including radio buttons and check boxes in Excel rather than typing values in cells.
- We will integrate GRNmap with GRNsight (http://dondi.github.io/GRNsight/), a web application and service that is being developed to visualize the results of the GRNmap modeling.
- Implementing the unit testing framework will accelerate model improvement and scientific inquiry by streamlining the verification, validation, and assurance process.
- Since GRNmap only runs on Windows, we intend to increase user accessibility by integrating the software into Mac machines.

Acknowledgments

We would like to thank Juan Carrillo, Nicholas A. Rohacz, Alondra Vega, Stephanie D. Kuelbs, Nathan C. Wanner, and Erika T. Camacho for previous work on the GRNmap program. This project was supported by the Summer Undergraduate Research Program at Loyola Marymount University (J.S.C.), NSF-DMS award #0921038 (K.D.D., B.G.F., and K.S), and the Clarence Wallen, S.J. Chair in Mathematics (B.G.F.).