

Restructuring the Data Architecture of GRNmap, a Gene Regulatory Network Modeling Application

Trixie Anne M. Roque¹, Chukwuemeka E. Azinge¹, Justin Kyle T. Torres¹, John David N. Dionisio¹, Ben G. Fitzpatrick², Kam D. Dahlquist³

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

Estimating Parameters and Running Forward Simulations with GRNmap

- A gene regulatory network (GRN) consists of transcription factors that control the gene expression of other transcription factors. The dynamics of a GRN describe how expression in the network changes over time.
- GRNmap is a MATLAB software package that models the dynamics of a GRN by solving ordinary differential equations (Equation 1) to estimate (Eq. 2) its weights, production rates, and expression thresholds of genes.

$$\frac{dx_i(t)}{dt} = \frac{P_i}{1 + \exp\left(-\left(\sum_j (w_{ij}x_j(t)) - b_i\right)\right)} - d_i x_i(t)$$

Equation 1: Differential equation for the change in expression of a particular gene.

Equation 2: Least squares error function to calculate how close the model is to input data.

- Input data are read from a Microsoft Excel workbook. The file should contain the following worksheets:
 - Production and degradation rates
 - Expression thresholds
 - Microarray data \log_2 fold change of expression (Figure 1) - the parameters of the network are estimated from this expression data
 - Network adjacency matrix and initial guesses
 - Optimization parameters

	A	B	C	D	E	F	G	H	I	J
1	id	0.4	0.4	0.4	0.8	0.8	0.8	1.2	1.2	1.2
2	ACE2	-0.376334	-0.376334	-0.376334	-0.706666	-0.706666	-0.706666	-0.987239	-0.987239	-0.987239
3	AFT2	-0.227719	-0.227719	-0.227719	-0.408018	-0.408018	-0.408018	-0.546413	-0.546413	-0.546413
4	CIN5	0	0	0	0	0	0	0	0	0

Figure 1: Example expression data for a network of 3 genes. The data contains 3 timepoints with 3 different replicates each.

- GRNmap outputs a new Excel workbook and a .mat file that contain the original input worksheets, optimized parameters, and simulation diagnostics. Graphs of gene expression over time are also created.

A Unified “GRNstruct” Data Structure Passes Data through the Multiple GRNmap Routines

- The general work flow of GRNmap was already established by previous members of our group. GRNmap first reads the spreadsheet and packages all required data. Most of the reported bugs arise from this stage due to mismatch between user inputs and GRNmap’s expected inputs. The data is then passed into a series of functions that conduct the estimation and forward simulation. An Excel file of the results with the corresponding graphs of individual gene expression over time are then returned to the user.
- GRNmap’s data structure is a GRNstruct which is filled by values read from the input spreadsheet created by the user. The GRNstruct is passed through different routines which change the values inside.

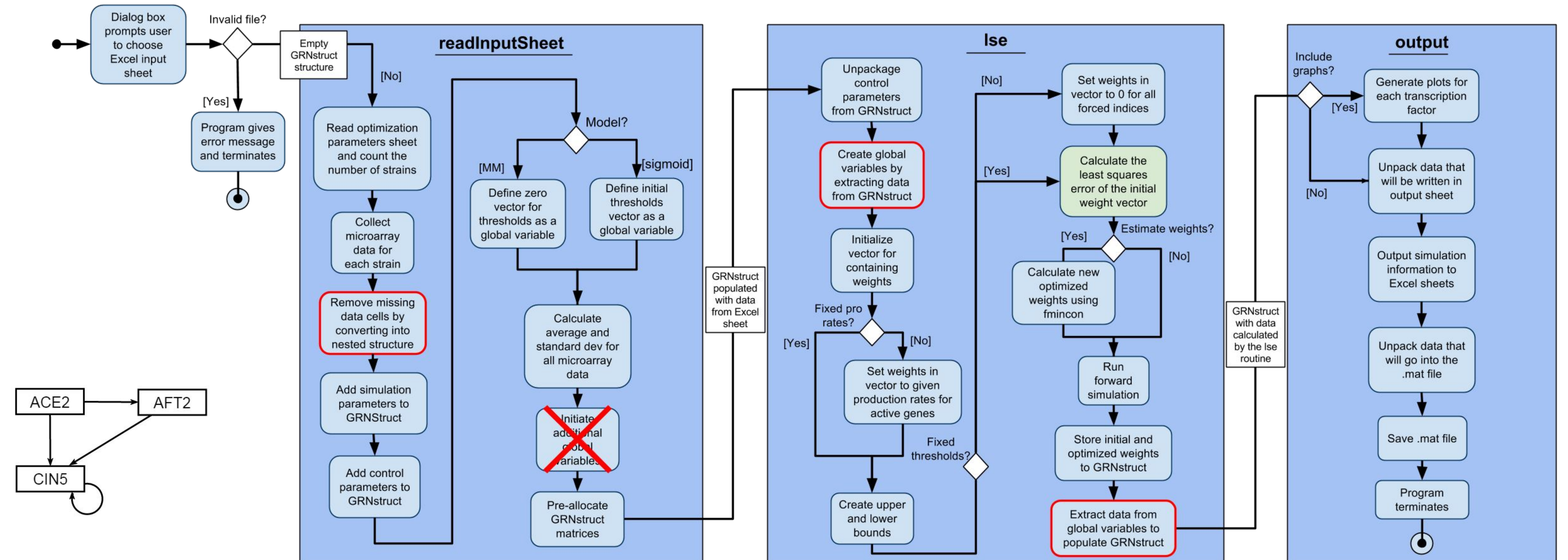


Figure 3: Above are the changes we implemented to mitigate software crashes, incorrect calculations, or readability issues. The boxes outlined in red signify the additions to the functionality of the code while the red cross signifies a removal.

Improved Data Structure Allows for Missing Expression Values

- One of the features we’ve added is the ability to have missing data in the expression data worksheets loaded into GRNmap.
- The previous data structure pulled the information directly from the Excel table, preserving their position and location.
- When missing values were encountered, it caused an error which crashed the program.
- Our change now groups together all of the data for a gene at a specific time point into a matrix, where the relative position of the data is preserved by an index in the lower row of the matrix.
- Since real-world data can contain omissions, this more complex structure allows our users to input expression data sheets that better reflect biological data gathered from a laboratory setting.

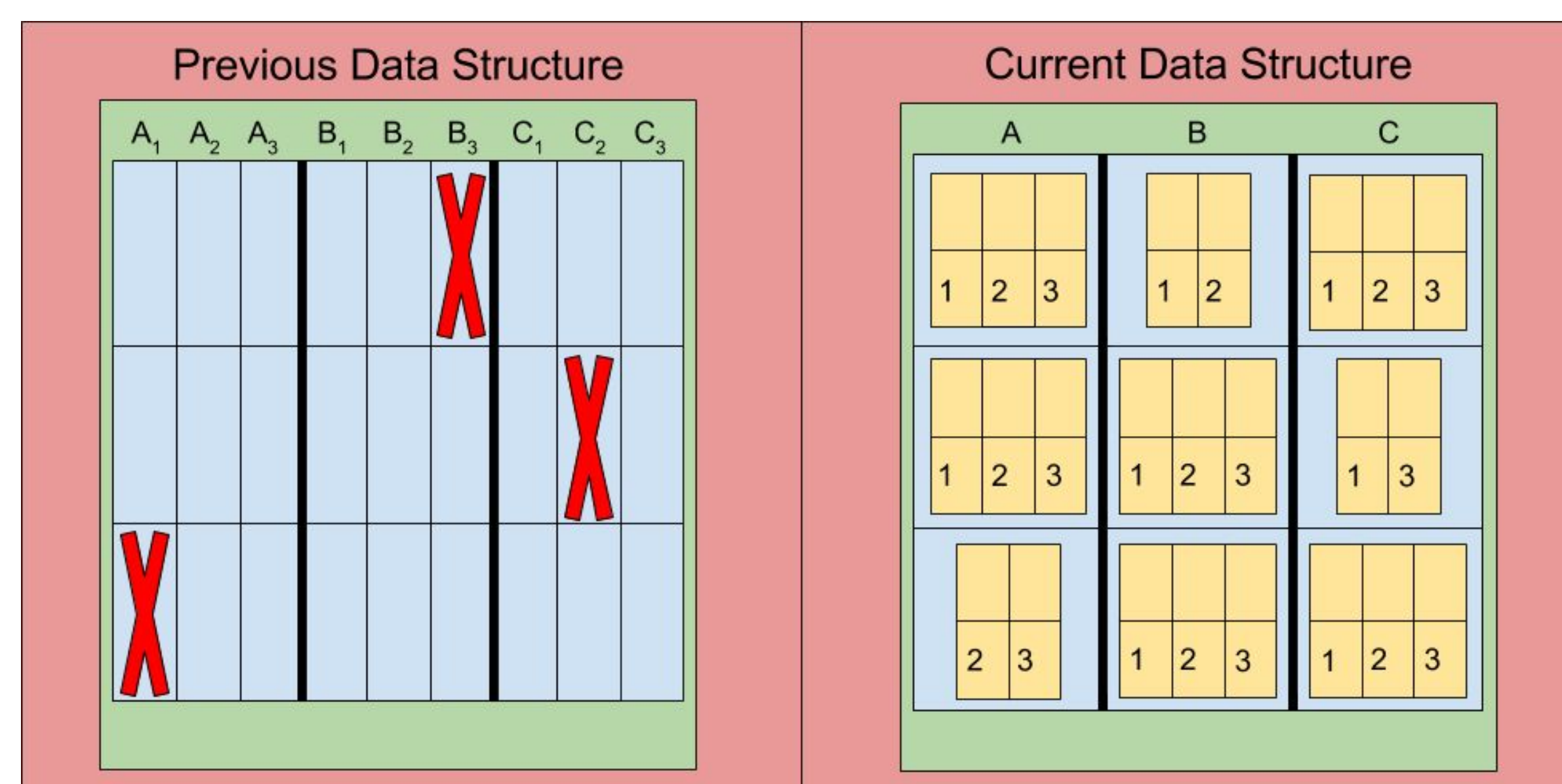


Figure 2: A comparison of the previous data structure (left) and the new data structure (right).

Code Refactored to Localize Global Variables

- The previous version of GRNmap was cluttered with global variables (shown in red in Figure 4) which often caused side-effects during execution.
- The holdover of values in the global variables caused bugs when users ran the model twice in a row.
- We’ve isolated the variables to the scopes of where they are used; opting to either place them into the GRNstruct, or dereference them once used.

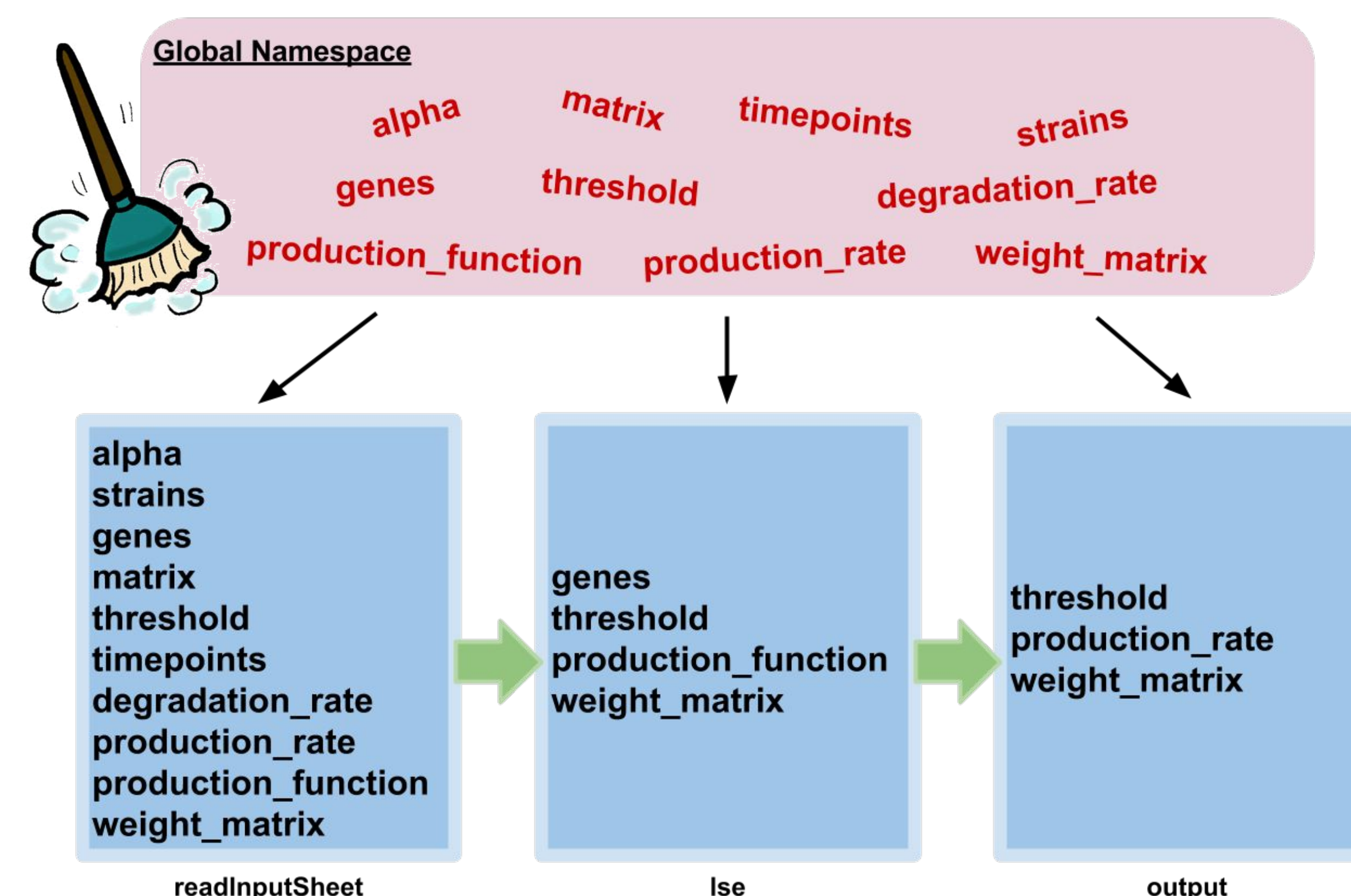


Figure 4: Global variables (top) being localized into the separate routines that need them (bottom). The blue boxes represent the main routines of GRNmap as shown in Figure 3. The words shown in black represent the variables that are changed and inserted into the GRNstruct.

Future Work

- Continue refactoring efforts
- Address current and future bugs that may arise
- Increase test coverage which is currently ~60%
- Add new features for data analysis requested by the GRNmap data analysis team members

Links

The latest GRNmap executable (57 downloads) and source code (49 downloads) are available for download at <http://kdahlquist.github.io/GRNmap/downloads> under a BSD Open Source license.

References

Dahlquist, K. D., Fitzpatrick, B. G., Camacho, E. T., Entzminger, S. D., & Wanner, N. C. (2015). Parameter Estimation for Gene Regulatory Networks from Microarray Data: Cold Shock Response in *Saccharomyces cerevisiae*. *Bulletin of Mathematical Biology*, 77(8), 1457–1492. <http://doi.org/10.1007/s11538-015-0092-6>

Acknowledgments

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