

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

Trixie Anne M. Roque<sup>1</sup>, Chukwuemeka E. Azinge<sup>1</sup>, Justin Kyle T. Torres<sup>1</sup>, John David N. Dionisio<sup>1</sup>, Ben G. Fitzpatrick<sup>2</sup>, Kam D. Dahlquist<sup>3</sup>  
<sup>1</sup>Department of Electrical Engineering and Computer Science, <sup>2</sup>Department of Mathematics, <sup>3</sup>Department of Biology, Loyola Marymount University, 1 LMU Drive, Los Angeles, CA 90045

## Running Simulations using 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 GRNs by solving ordinary differential equations to estimate the weights, production rates, and expression thresholds of genes in the GRNs.
- Input data are read from a Microsoft Excel workbook. The file should contain the following worksheets:
  - Production rates
  - Degradation rates
  - Expression thresholds
  - Microarray data log2 fold change of expression (Figure 1)
  - Adjacency matrix that describes the network
  - Initial guesses for network
  - 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 CINS	0	0	0	0	0	0	0	0	0	
5 FHL1	-0.139322	-0.139322	-0.139322	-0.249852	-0.249852	-0.249852	-0.336141	-0.336141	-0.336141	

Figure 1: Example expression data for a network of 4 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.

## Data Structure Passes through Multiple 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 gets passed into a series of functions that conduct the simulation and the results are displayed in a user-friendly format.
- Figure 2 shows 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.

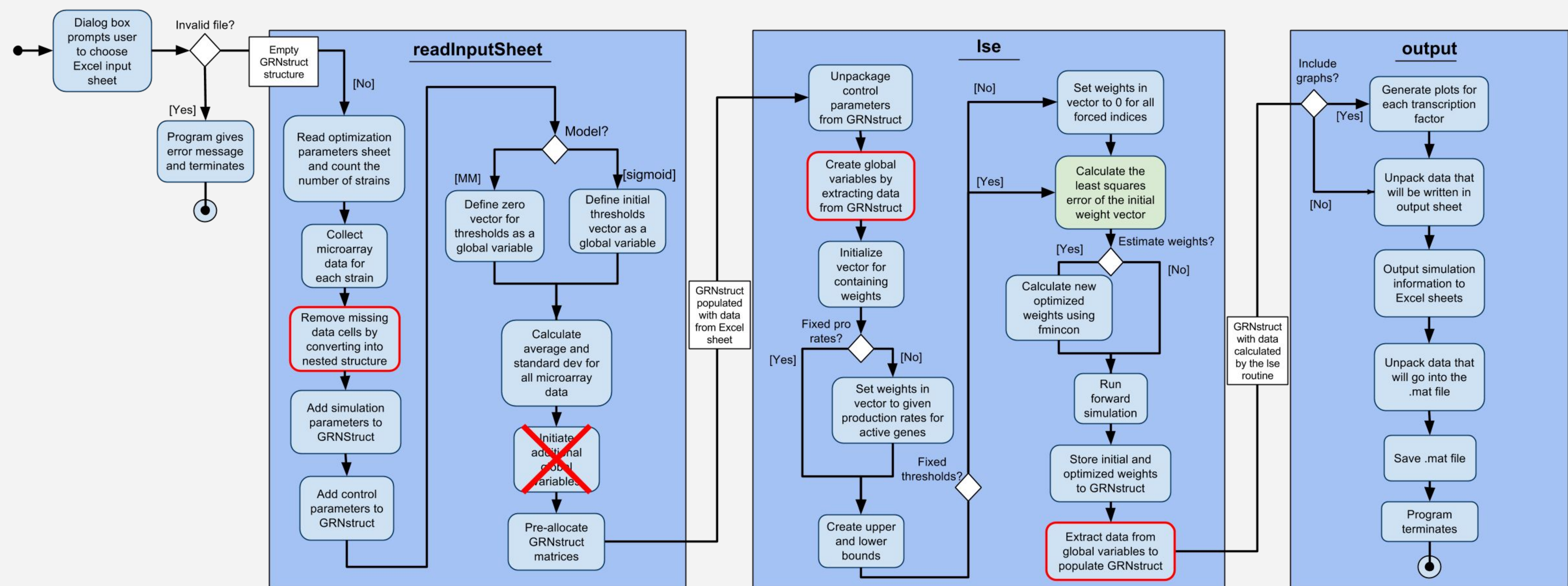


Figure 2: GRNmap uses a GRNstruct (white box above) to hold data. The GRNstruct is passed through the different routines (outer blue boxes) which change the values inside.

## Improved Data Structure

- One of the features we've added is the ability to have missing data in the worksheets loaded into GRNmap.
- The previous data structure pulled the information about the strains directly from the Excel table, preserving their position and location.
- When missing values are encountered in a replicate, their values cause errors which crash 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 the lower row of the matrix.
- As a result, we've simplified the use of our data structure within GRNmap, and allowed room for more dynamic changes in the future.

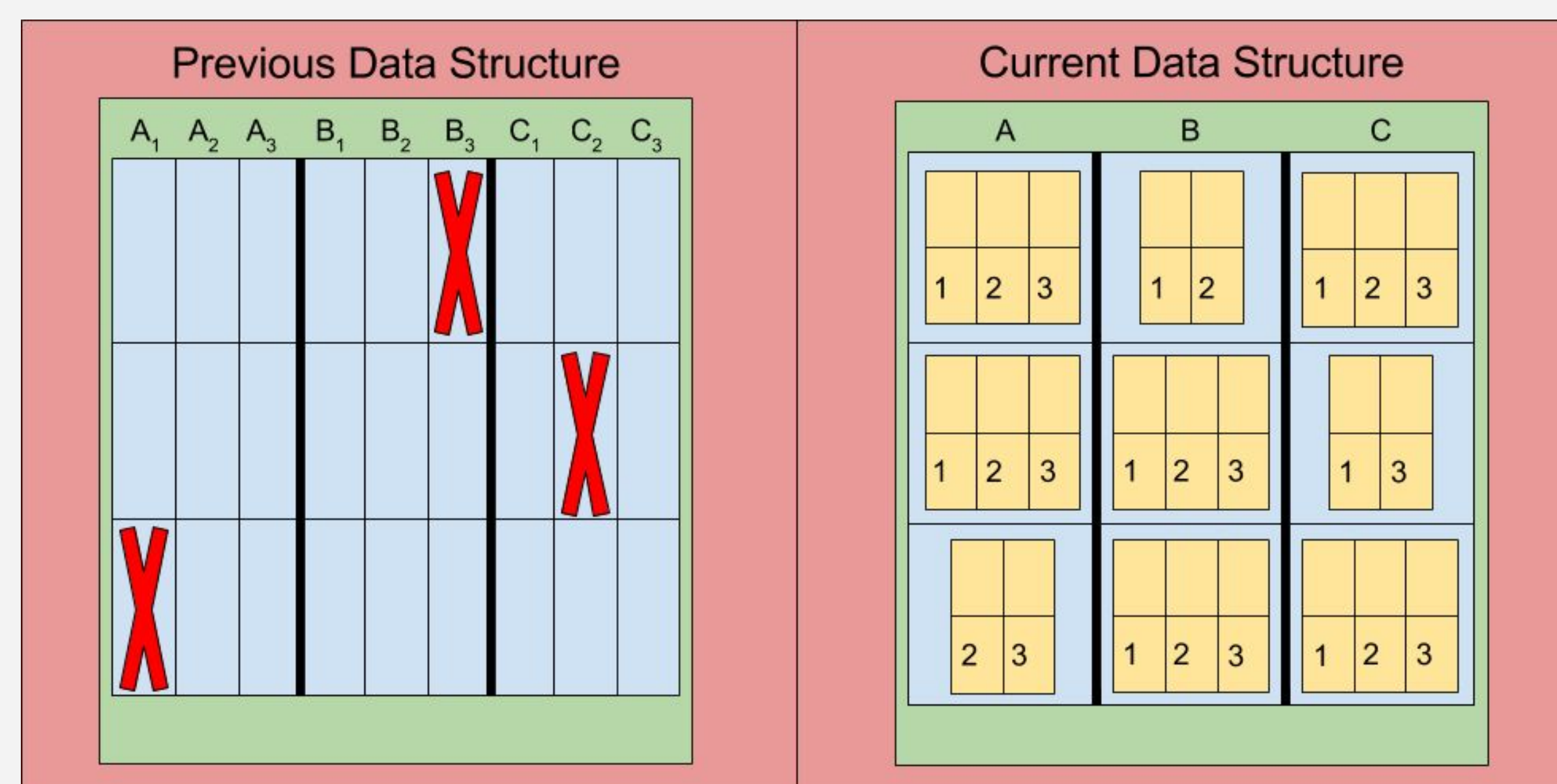


Figure 3: A comparison of the previous data structure (left) and the new data structure (right).

## Localized Global Variables

- The previous version of GRNmap was cluttered with global variables which often caused side-effects during execution.
- This often prevented us from running GRNmap on multiple files at a time, as data from previous iterations would cause unintended bugs to happen.
- 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.

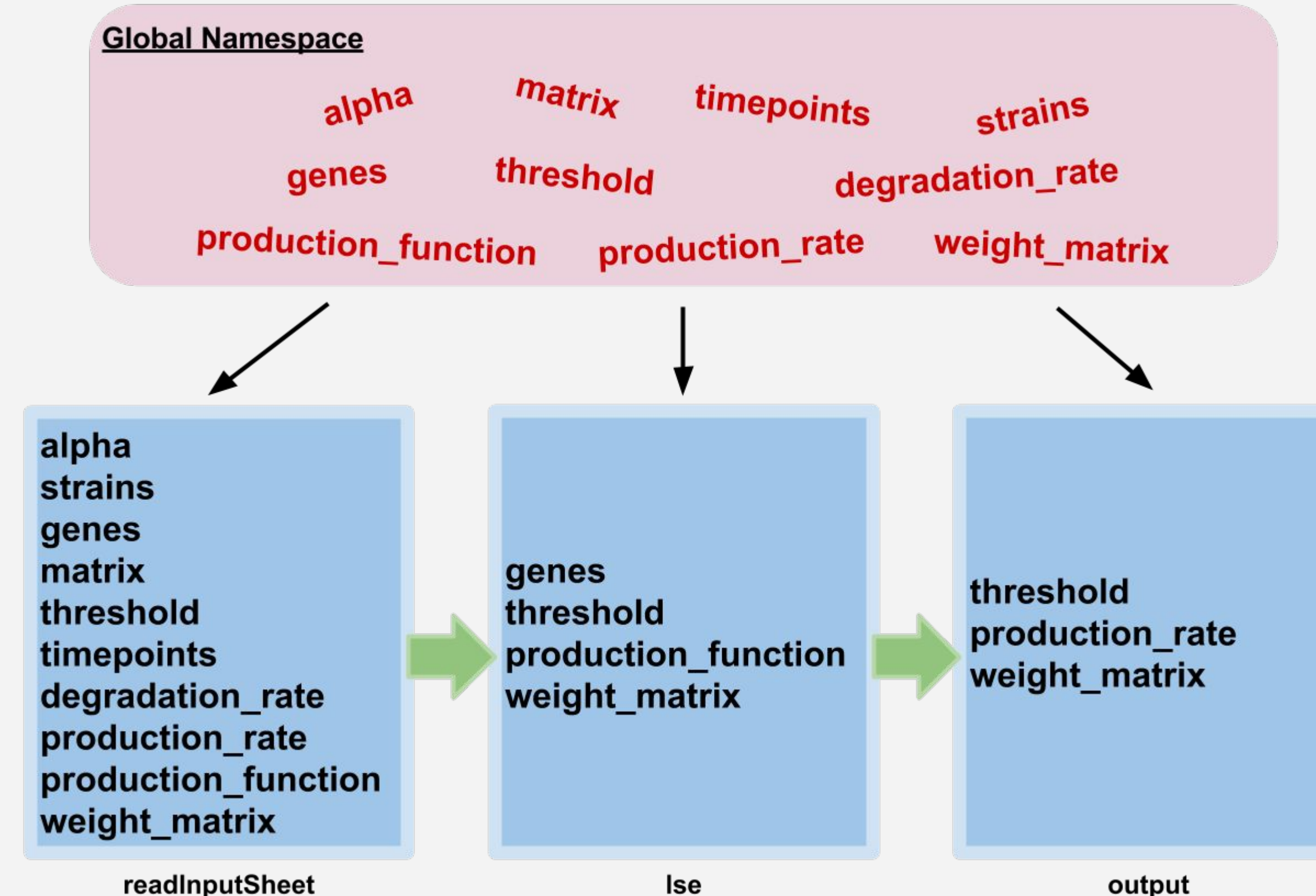


Figure 4: A comparison of the previous structure of our global variables (top) and our newly localized variables (bottom).

## Future Work

- Continue refactoring efforts
- Address current and future bugs that may arise
- Increase test coverage
- Add new features to simplify and embetter typical use of our software

## Links

The latest GRNmap executable and source code can be download from <http://kdahlquist.github.io/GRNmap/downloads/>

## References

Dahlquist, K.D., Fitzpatrick, B.G., Camacho, E.T., Entzminger, S.D., and Wanner, N.C. (2015) Parameter Estimation for Gene Regulatory Networks from Microarray Data: Cold Shock Response in *Saccharomyces cerevisiae*. *Bulletin of Mathematical Biology*. DOI: 10.1007/s11538-015-0092-6, Epub ahead of print.

## Acknowledgments

We would like to thank 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.).