Modeling of Gene Regulatory Networks

Authors

Istiaq Mohammad Student ID: 144421
Irtiza Chowdhury Student ID: 144430

Supervisor

Tareque Mohmud Chowdhury

Assistant Professor
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)

A Thesis submitted to the Department of Computer Science and Engineering (CSE)
In Partial Fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering (CSE)

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)
Organization of the Islamic Cooperation (OIC)
Gazipur, Bangladesh

November, 2018

Declaration of Candidates

This is to certify that the work presented in this thesis is the outcome of the analysis and investigation carried out by the candidates under the supervision of Tareque Mohmud Chowdhury in the Department of Computer Science and Engineering (CSE), IUT, Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

(Signature of Candidate)

Istiaq Mohammad
Student No. 144421
Academic Year 2017-2018
November, 2018

(Signature of Candidate)

Irtiza Chowdhury
Student No. 144430
Academic Year 2017-2018
November, 2018

(Signature of Supervisor)

Tareque Mohmud Chowdhury
Assistant Professor
Department of Computer Science & Engineering (CSE)
Islamic University of Technology (IUT)
November, 2018

Abstract

Many crucial molecular processes and cellular pathways are based on the interactions among genes. The genes in living cells regulate each other to control the production of gene products. Gene regulatory networks provide information on the control at gene expression level and can be inferred from a number of data-sets expressed in different ways. There are two types of gene expression data used for gene regulatory network construction: time series and perturbation experiments. Time series expression data enables biologists to investigate the temporal pattern in biological networks. Perturbed expression data provides the information on interactions directions. In the past, gene regulatory networks were constructed by using the clustering approach. However, this approach failed to identify significant transcriptional network interactions. Hence, many computational approaches have been developed for constructing gene regulatory networks more effectively. Reverse engineering from given data-sets can prove to computationally challenging, so the approach taken aims to construct stable and scalable gene regulatory networks from given steady state data.

Table of Contents

Declaration of Candidates....1
Abstract....2
1. Introduction....4
1.1. Gene Regulation....4
1.2. Gene Expression Using Microarrays....5
1.3. Gene Expression Data....6
1.4. Constructing Gene Regulatory Networks from Data....7
2. Problem Domain and Problem Statement....9
2.1. Challenges and Research Issues....9
2.2. Problem Statement....9
3. Literature Review....10
3.1. Chai, et al.....10
3.2. Vijesh, et al.....12
3.3. Zavlanos, et al.....14
3.4. Larvie, et al.....16
3.5. Langfelder and Horvath....18
4. Methodology....19
4.1. Input and Pre-processing....20
4.2. Construction of the Gene Regulatory Network and module detection....22
4.3. Visualization of the network....27
5. Comparative Analysis....30
6. Motivation and Future Work....34

1. Introduction

1.1. Gene Regulation

Proteins control a cell's metabolism and are the building blocks of life, responsible for most of the structural and chemical functions in a living organism. The blueprint for the organism is found in its DNA and each gene within it codes for a different protein. Gene expression usually refers to the transcriptional and translational processes within cells. Sophisticated programs of gene expression are widely observed in biology, for example to trigger developmental pathways, respond to environmental stimuli, or adapt to new food sources. Virtually any step of gene expression can be modulated or regulated by a variety of factors, such as the rate of transcription, the processing of mRNA, the stability of the mRNA, and the rate of translation. It can also be encouraged or inhibited by a type of protein known as a transcription factor. They bind to specific sites on the DNA as per their role. As transcription factors are proteins, they are also synthesized from genes, which means genes themselves play a role in the expression of genes.

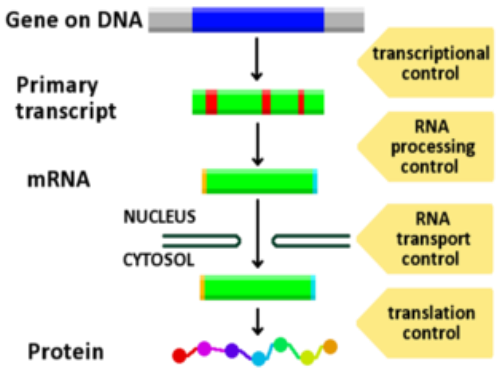

Figure 1 - The process of protein synthesis

1.2. Gene Expression Using Microarrays

DNA microarrays or DNA biochips act as trays containing the complementary sequence of the gene whose expression is required to be measured. Several cells can be tested simultaneously through extraction of cDNA and hybridization. Labelling of the separate cDNA molecules allows them to be identified in the final expression data. Once the hybridized solution reacts or binds with the complementary DNA sequence in the microarray, each specific gene is said to be expressed and can be visually represented. For example, if the same cDNA is labeled 'red' in a cancer cell and 'green' in a control cell, the individual gene expression can be monitored, displaying their participation under the different conditions as seen in Figure 2. Microarrays can pave the way to biological discovery of new and better molecular diagnostics, molecular targets for therapy, finding and refining biological pathways, and for mutation and polymorphism detection. Recent examples include molecular diagnosis of leukemia and breast cancer.

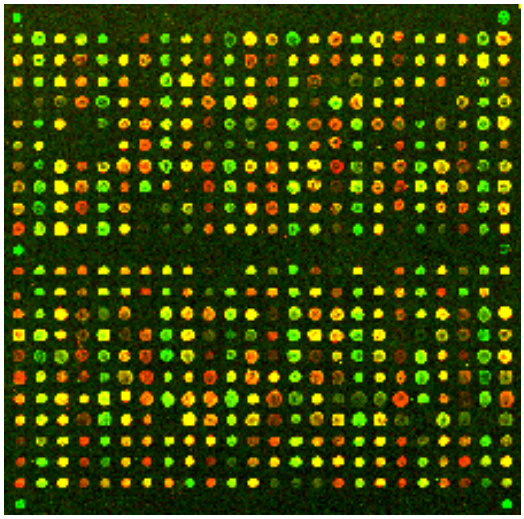

Figure 2 - The result of a DNA microarray experiment

1.3. Gene Expression Data

Results from microarrays can be converted into what we call gene expression data. Microarray data sets are commonly very large, and analytical precision is influenced by a number of variables. Statistical challenges include considering effects of background noise and appropriate normalization of the data. Image analysis and class detection analysis can be used to generate the required information and description from the visual representation of the expression data. The following table depicts several genes and their expression levels within different samples.

Gene Description	1	2	3	4	5	6	7
AFFX-BioB-5_at (endogenous control)	-214	-139	-76	-135	-106	-138	-72
AFFX-BioB-M_at (endogenous control)	-153	-73	-49	-114	-125	-85	-144
AFFX-BioB-3_at (endogenous control)	-58	-1	-307	265	-76	215	238
AFFX-BioC-5_at (endogenous control)	88	283	309	12	168	71	55
AFFX-BioC-3_at (endogenous control)	-295	-264	-376	-419	-230	-272	-399
AFFX-BioDn-5_at (endogenous control)	-558	-400	-650	-585	-284	-558	-551
AFFX-BioDn-3_at (endogenous control)	199	-330	33	158	4	67	131
AFFX-CreX-5_at (endogenous control)	-176	-168	-367	-253	-122	-186	-179
AFFX-CreX-3_at (endogenous control)	252	101	206	49	70	87	126
AFFX-BioB-5_st (endogenous control)	206	74	-215	31	252	193	-20
AFFX-BioB-M_st (endogenous control)	-41	19	19	363	155	325	-115
AFFX-BioB-3_st (endogenous control)	-831	-743	-1135	-934	-471	-631	-1003
AFFX-BioC-5_st (endogenous control)	-653	-239	-962	-577	-490	-625	-761
AFFX-BioC-3_st (endogenous control)	-462	-83	-232	-214	-184	-177	-541
AFFX-BioDn-5_st (endogenous control)	75	182	208	142	32	-94	109
AFFX-BioDn-3_st (endogenous control)	381	164	432	271	213	222	435
AFFX-CreX-5_st (endogenous control)	-118	-141	84	-107	1	-1	-129
AFFX-CreX-3_st (endogenous control)	-565	-423	-501	-101	-260	-140	-399
hum_alu_at (miscellaneous control)	15091	11038	16692	15763	18128	34207	30801
AFFX-DapX-5_at (endogenous control)	7	37	183	45	-28	65	43
AFFX-DapX-M_at (endogenous control)	311	134	378	268	118	154	80
AFFX-DapX-3_at (endogenous control)	-231	-161	-221	-27	-153	-49	-87

Figure 3 - Several genes and their expression levels

1.4. Constructing Gene Regulatory Networks from Data

Gene Regulatory Networks are composed of the participating genes and other regulatory molecules which help to govern the gene expression. These entities interact with each other in several ways, including activation and inhibition. The nodes of this network are genes and the edges between nodes rep- resent gene interactions through which the products of one gene affect those of another. These interactions can be inductive, with an increase in the expression of one leading to an increase in the other, or inhibitory, with an increase in one leading to a decrease in the other. A series of edges indicates a chain of such dependences, with cycles corresponding to feedback loops.

The modulation and functionality of the entire network can act as a blueprint for researchers who are willing to observe the relationship between genes. The systematic understanding of molecular mechanisms underlying biological processes can aid in the discovery of triggering mechanism and adaptability techniques. Several novel experimental and computational approaches have recently been developed which helps to comprehensively characterize these regulatory networks by enabling the identification of their genomic or regulatory state components.

Constructing dynamic GRNs is gaining significance in biomedical research and analysis. Reverse engineering is not a computationally simple problem because an enormous amount of time is required even with trivial approaches.

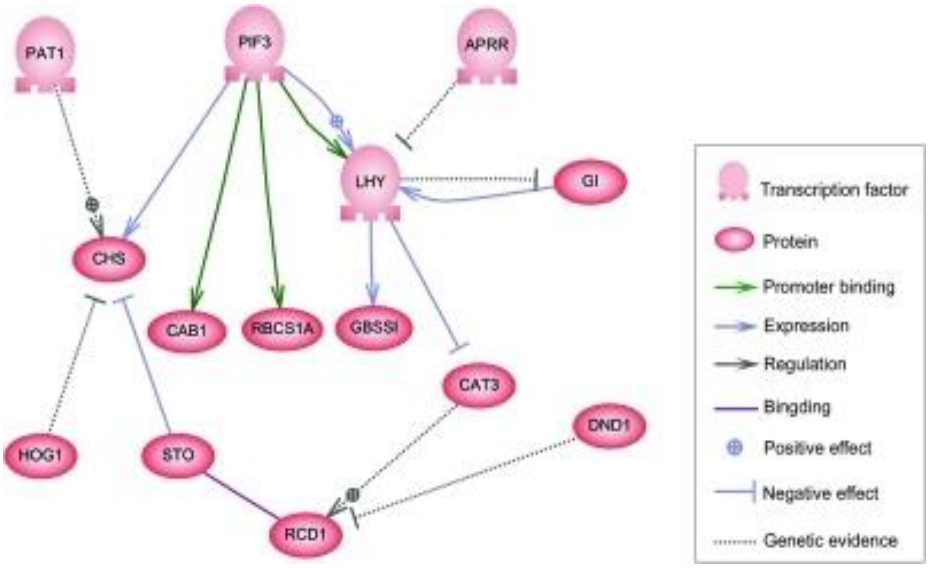

Figure 4 - An example of a gene regulatory network

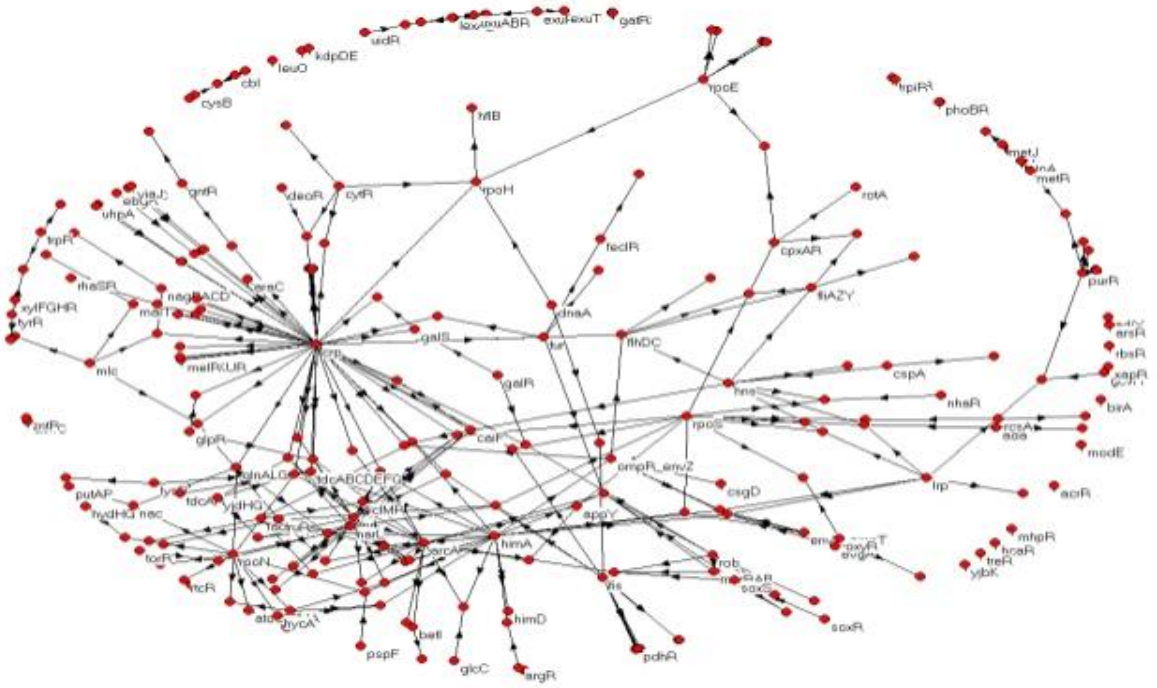

Figure 5 - A gene regulatory network in E. Coli. Nodes are operons. Some operons encode for transcription factors. Transcription factors regulate other operons

2. Problem Domain and Problem Statement

2.1. Challenges and Research Issues

The primary challenges encountered when modelling gene regulatory networks depend highly and almost entirely upon the model selected and the data set used. Usually, most models rely on data difficult to obtain, and the simpler ones do not sufficiently express the network pertaining to the given data-set. Computational effort is a major challenge when it comes to constructing networks as the simplest form can consist of thousands of genes which require extensive manipulation and calculation. Reducing computational effort compromises scalability of the approach. As a result, it is wiser to work with models which are justifiable in their process and procedure.

2.2. Problem Statement

"Model stable and scalable gene regulatory networks from given gene expression data."

3. Literature Review

3.1. Chai, et al.

"Computational Approaches for Gene Regulatory Network Construction" [1] discusses six different inference techniques to identify gene regulatory networks from gene expression data. The paper highlights several existing computational models, analyzes the methodology of each model and provides examples of recent applications of each model in the construction of gene regulatory networks. It also compares them against each other, highlighting the strengths and weaknesses of each technique.

Boolean Networks are the simplest inference model, with each gene being expressed as either on or off. They are fast and efficient from a computational perspective, and easy to visualize in the form of directed graphs due to only having two possible states for each node. The interaction types are divided into two classes: active and inactive when building a model which means the two possible states for a gene in such a network are on or off. Due to their simplicity, they may not be able to accurately represent finer details, such as a change in the *rate* of expression of a gene or other non-binary factors, and because of their deterministic nature, yield inconsistencies when exposed to noisy data. Updates are also time discrete and synchronous, whereas in most biological systems, updates are asynchronous.

One application of Boolean Networks was it being used to model the cell cycle of fission yeast, *Schizosaccharomyces pombe.* The network dynamics accurately reproduced the protein activation based on a time sequence.

Probabilistic Boolean Networks are an extension of Boolean Networks where a probability is assigned to each entity and a regulation function is selected based on probability. They can be described as a bound collection of Boolean Networks where at any given instance of time, the state transitions occur according to the rules constituent in one of the networks. This addresses the issue of the deterministic nature of Boolean Networks, however, it also makes it computationally expensive, making it more difficult to implement for larger networks.

Probabilistic Boolean Networks showed dependencies between genes and their parents, derived using tumor cells as data.

Bayesian Networks model the relationship using directed acyclic graphs and conditional probability tables. They can easily deal with noisy data, and handle uncertainty. Belief propagation is used for inference, working by updating beliefs across the network based on evidence. The network is constructed during the model selection phase, but the probability values are estimated during parameter learning. Bayesian Networks can integrate prior knowledge to strengthen causal relationships and use statistics to infer the structure of a network. However, they cannot capture temporal information, deal with larger networks, find it difficult to distinguish between the origin and the target of an interaction, and don't allow feedback loops.

Bayesian Networks were used alongside other approaches to model networks in *E. Coli*.

Dynamic Bayesian Networks are an extension of Bayesian Networks that can be used to model cyclic interactions, infer uncertainties and yield more reliable data using perturbation experiments. They infer interaction uncertainties between genes using a probabilistic graphical model. Cyclical interactions are modeled by the duplication of nodes. They can handle temporal data by using interconnecting time slices and can model direct or indirect causal relationships. However, this method is far more computationally expensive, and still cannot handle larger networks.

Ordinary Differential Equations involve continuous variables used for non-linear systems and changing concentrations of mRNA to infer stability. They are suited for steady-state and time series expression profiles and can work entirely in a classical category. Like DBN, they also allow the improvement of networks via the introduction of perturbation. They are the best analyzed approach for non-linear systems due to their use of continuous variables. However, due to the high computational cost, they are only feasible for smaller networks.

Neural Networks capture dynamic and non-linear interactions within networks. They can handle noise and feedback loops and recurrent neural networks with clustering can be used to solve the issue of scalability. They are flexible and can recognize input patterns, modeling functional relationships and data structures, and can capture the nonlinear and dynamic interactions between genes. However, they are extremely expensive computationally, and it is difficult to obtain efficient training.

3.2. Vijesh, et al.

"Modeling of Gene Regulatory Networks: A Review" [2] categorizes existing computational models and introduces new ones while highlighting their strengths and weaknesses. Their analysis can be summarized as follows:

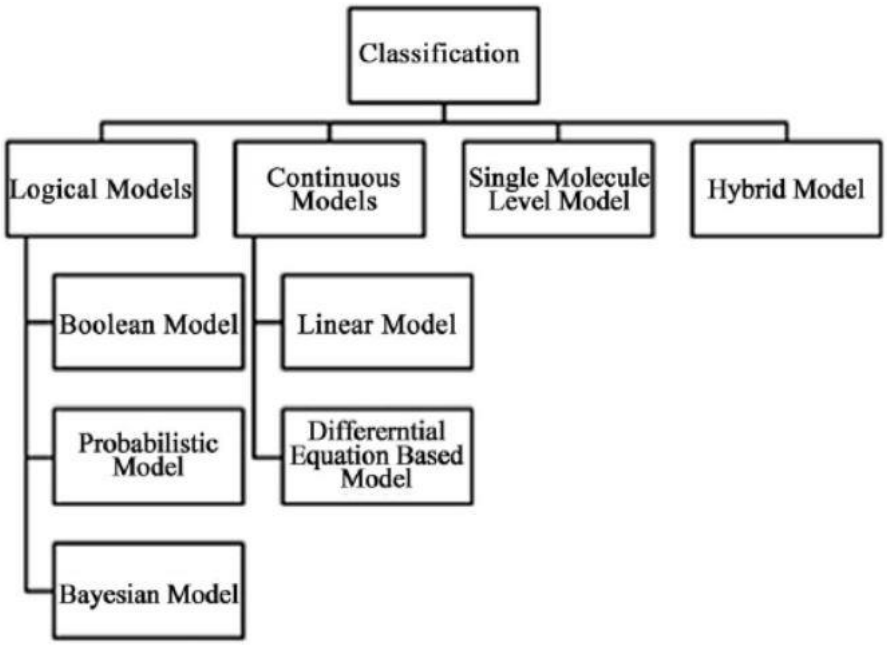

Figure 6 - The classification of various computational models

Boolean Networks, as discussed previously, are simplistic but two states are insufficient as there can be multiple levels of gene expression, and they are deterministic, making them susceptible to noise. They also update synchronously, while typical biological systems update asynchronously. Also, despite how simple they are, only small networks can be reverse engineered using this method, even when using current algorithms.

Probabilistic Boolean Networks are stochastic, overcoming the rigidity of the deterministic Boolean Network model. This is done by allowing each state to have several regulation functions, each of which is assigned a probability based on how closely it agrees with prior knowledge. However, the state space is still discrete.

Bayesian Networks are joint probability distributions over a set of random variables. Classification is done with the help of Bayes' Theorem, and the approach is made naïve, or simplified, in order to reduce computational costs. This approach combines graph theory and probability to deal with noise and is stochastic but requires high computational effort to incorporate loops and fails to consider temporal dynamic aspects of the network. One major advantage of Bayesian Networks is their ability to learn from observed data and they have become a very popular method of modeling regulatory networks as a result.

Differential Equations are highly customizable due to their use of simple homogenous structures. It is assumed that the rate of gene expression depends entirely on the concentration of products of genes from the nodes in a regulatory network, which implies that external factors such as the influence of other molecules are not considered. Despite the simplification, this approach can be used to decipher the basics of interactions between genes. However, this approach involves a large number of parameters – of the order $O(d^2)$, where d is the number of genes modeled.

Linear Models do not require extensive prior knowledge but cannot capture non-linear aspects. One of the main draws to this model is the fact that each regulator contributes independently of other regulators to the regulation functions, in an additive manner, making it very easy to make surface level estimates about the network. However, they cannot be more than a simplification of the actual system, and are therefore, not widely used to reverse engineer entire models.

Single Molecule Level Models are most detailed but extremely computationally expensive and are therefore only feasible when a small number of molecules are present. Due to the fact that the scale is of such fine grain, this provides the highest level of insight into the stochastic behavior of the gene regulatory process.

3.3. Zavlanos, et al.

"Inferring Stable Genetic Networks from Steady-State Data" [3] illustrates the many weaknesses of most models, such as when encountering loops or addressing causality. Highlights include the lack of inferring causality in Boolean networks and inability to incorporate feedbacks when it comes to Bayesian networks. The study uses genetic perturbation experiments at steady-state, representing their findings in the form of matrices. Small perturbations are introduced to equilibrium states and the resulting gene expression activity is measured. The target is to introduce stability into the network and observe the performance of the results compared to existing networks through quantitative measures.

It models networks using differential equations to develop linear constraints and algorithms - three such models are developed and scaled to deal with larger volumes of data. The matrices involved include information on pairwise interaction of genes, transcription perturbations and associated steady-state mRNA concentrations. The initial linear programming approach proves to be very slow, so a convex relaxation is included for scalability. Prior knowledge is added to further augment the validity of the findings and weak interactions are eliminated. The stability of RNA is considered by introducing it as a constraint and the performance is measured using quantities such as Sensitivity and Specificity. ROC curves are compared for the three algorithms, where the third is the intended approach. The results yielded were stable and spares, implying the underlying network was stable, showing that stability is not only important for consistency with the problem assumptions, but also for better performance. This was tested on the SOS pathway of *E. Coli* to model known and inferred gene regulatory networks.

All identifications obtained from algorithm 1 are unstable, while the obtained networks have connectivity approximately equal to 50%. Compared to this, algorithm 2 yields in a matrix with 7 false positives, 3 false negatives, 16 false zeros, and 26 false identifications in total. Algorithm 3 on the other hand has 3 false positives, 6 false negatives, 16 false zeros, and 25 false identifications in total, while it is also stable and satisfies the desired sparsity pattern.

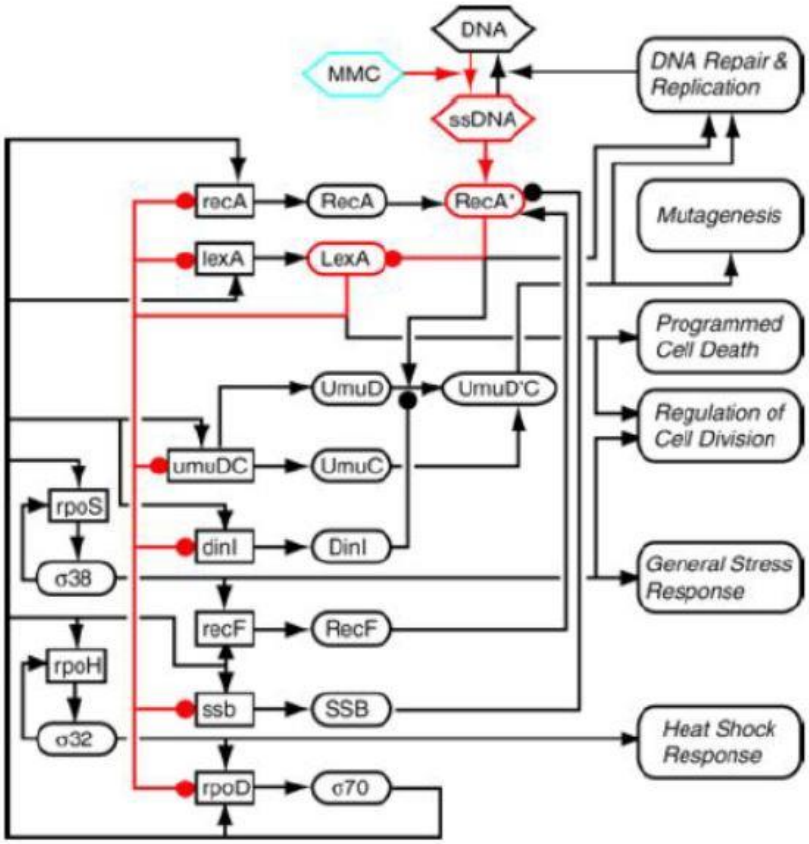

Figure 7 - SOS pathway in E. Coli

3.4. Larvie, et al.

"Stable Gene Regulatory Network Modeling from Steady-State Data" [4] is a study on steady-state data featuring genetic perturbation experiments. It is a means to return a sparse and stable network without the use of prior knowledge from said noisy perturbation experiments. The convex nature of the algorithm used contribute to an output which can be efficiently scaled.

A plethora of modelling approaches exist, but they lag behind in certain aspects such as describing causality or incorporating feedback motifs. Other models are computationally expensive or are suitable for only small-scale networks. Relying on temporal expression data also proves to be a hindrance as they are difficult to acquire.

This study uses the Vector Autoregressive (VAR) model, which was initially developed for analysis and prediction of economic and time series. There are equations for each evolving variable and data around millions of genes can be effectively scaled. The technique is most flexible and easy to use for analyzing multivariate time series, resulting in a rise in use in neuroscience and most recently, in GRNs.

It also uses the least absolute shrinkage and selection operator (LASSO) technique which selects covariates and improves prediction. The method also improves the interpretability of regression models by only selecting a subset of the original data set to be used in the problem. Variables which have substandard performance are set to zero by comparing to a penalty term. A stability constraint is introduced to work on steady-state data, and the optimization problem is solved using a penalty parameter. Zavlanos et al and Geršgorin's theorem are utilized to incorporate the stability constraint while maintaining the convex nature of the entire network. The result is a sparse and stable matrix, from which a Gene Regulatory Network is constructed by converting the time series of gene expression into a matrix where the rows are expression of various genes and the columns are observations at different time points. This study was performed on *E. Coli* and yeast cells and the sensitivity and specificity were compared. The inferred network without prior knowledge is mostly accurate, stable, scalable and sparse.

The following table shows how the proposed network identification algorithm without a priori knowledge of the network structure compares with that proposed by Zavlanos et al. with 30% a priori knowledge of the network.

	TP	FP	TN	FN	Sensitivity	Specificity	Precision
LASSO-VAR	39	11	5	26	60%	31%	78%
ZAVLANOS	40	10	15	16	71%	60%	80%

Figure 8 - The comparison of the LASSO-VAR and Zavlanos methods

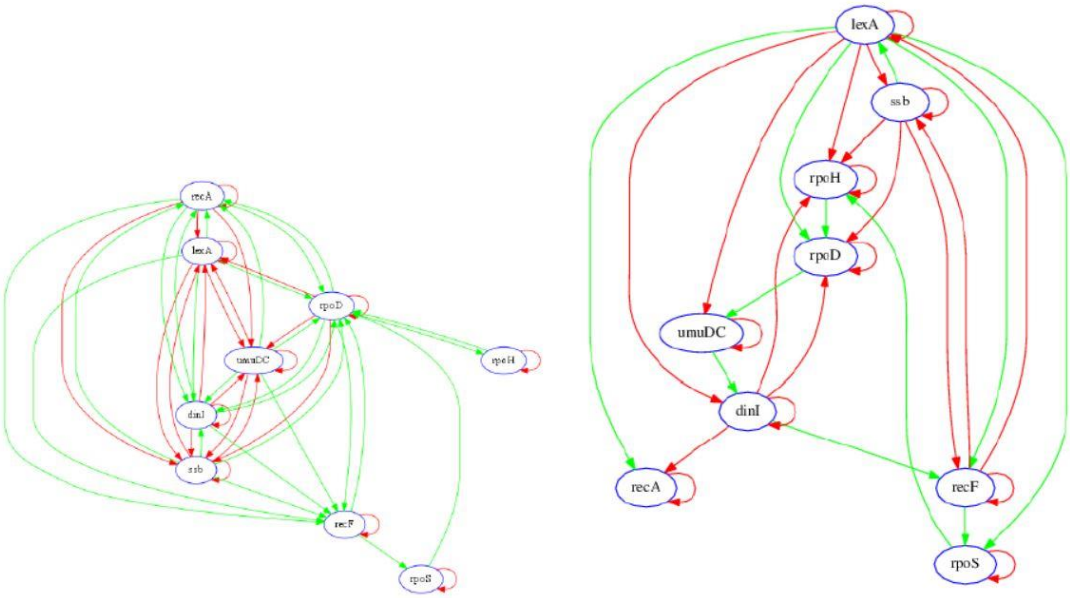

Figure 9 - Known and inferred networks of SOS pathway in E. Coli

The identification algorithm accurately identified the key regulatory associations in the network.

For instance, the model correctly shows that lexA activates recA while negatively regulating its own transcription, whereas recA negatively regulates its own transcription. In addition, the model identified lexA as having the greatest regulatory influence on the other genes in the network. Due to the differences in network topology (e.g., recA, lexA and CDC20), inaccuracies are expected from either the current published model, the LASSO-VAR GRN recovery, or both. Some of these potential differences may alternatively be dependent on the dynamic state of the system as inferred from the temporal context.

3.5. Langfelder and Horvath

"WGCNA: an R package for weighted correlation network analysis" [5] is a study into weighted gene correlation network analysis, a method for describing correlation patterns among genes across microarray samples. Using this method, several important applications are found, which have been compiled into an R package, accompanied by several tutorials to allow for consistent, user-friendly implementations of the various techniques discussed.

For instance, WGCNA can be used to find clusters, or modules of highly correlated genes. In an unsigned co-expression network, modules correspond to clusters of genes with high absolute correlations. In a signed network, modules correspond to positively correlated genes.

The clusters can then be summarized into eigengenes, defined as the first principal component of a given module. It can be considered a representative of the gene expression profiles in a module.

Trait data can be incorporated along with the module eigengenes to identify potentially significant genes. This allows the association of genes to external factors and a way to mathematically determine the significance of physical traits.

While some of the above techniques have been discussed in other papers, this paper provides a user-friendly implementation, a consistent software platform, and the tutorials required to study the code. It succeeds in doing so in the form of a library of R functions to be used in network construction, module detection, gene selection, calculations of topological properties, data simulation, visualization, and even interfacing with external software. Not only that, the paper also provides step by step tutorials detailing each step of the process.

Our main purpose of reviewing this paper was to have a better understanding of the methodology involved with analyzing gene regulatory networks from weighted coexpression data. Using the tutorials provided, a solid grasp of the procedure involved with the analysis of micro-array gene data was obtained.

4. Methodology

For the construction of the Gene Regulatory Network, we have used R v3.5.1 alongside RStudio 1.1.456. We used the R package *WGCNA* (Weighted Gene Co-expression Network Analysis) developed by Peter Langfelder and Steve Horvath [5] on a dataset of gene expression levels obtained from the livers of female mice with over 3600 expression profiles filtered from over 20000 samples collected [6].

It allows pairwise correlation between variables to be studied to a considerable extent. The functionalities provided can be widely applied to high-dimensional data sets, making it a valuable asset in the field of genomics. The method allows formation of clusters or modules and network nodes with regard to module membership, paving a simpler way to the analysis of relationships between co-expression modules, and to the comparison of network topology of different networks. Apart from the reduction of data, it is also suitable for feature selection and clustering.

Once we had a grasp of the internal concept and functions provided by the package, we applied the method on a dataset of the Yeast *Saccharomyces Cerevisiae*, containing 4000 genes and their expressions taken at different times. [7] The process of weighted gene co-expression network is detailed in the following sections.

4.1. Input and Pre-processing

We begin by loading the given expression data and removing any unnecessary auxiliary information that may be contained within. Following this, we analyze the data, which now only contains gene expression data for severe outliers and entries with too many missing values and remove them from the data to be processed. Outliers are found by clustering the samples hierarchically and then choosing an appropriate height cut as shown in Figure 10.

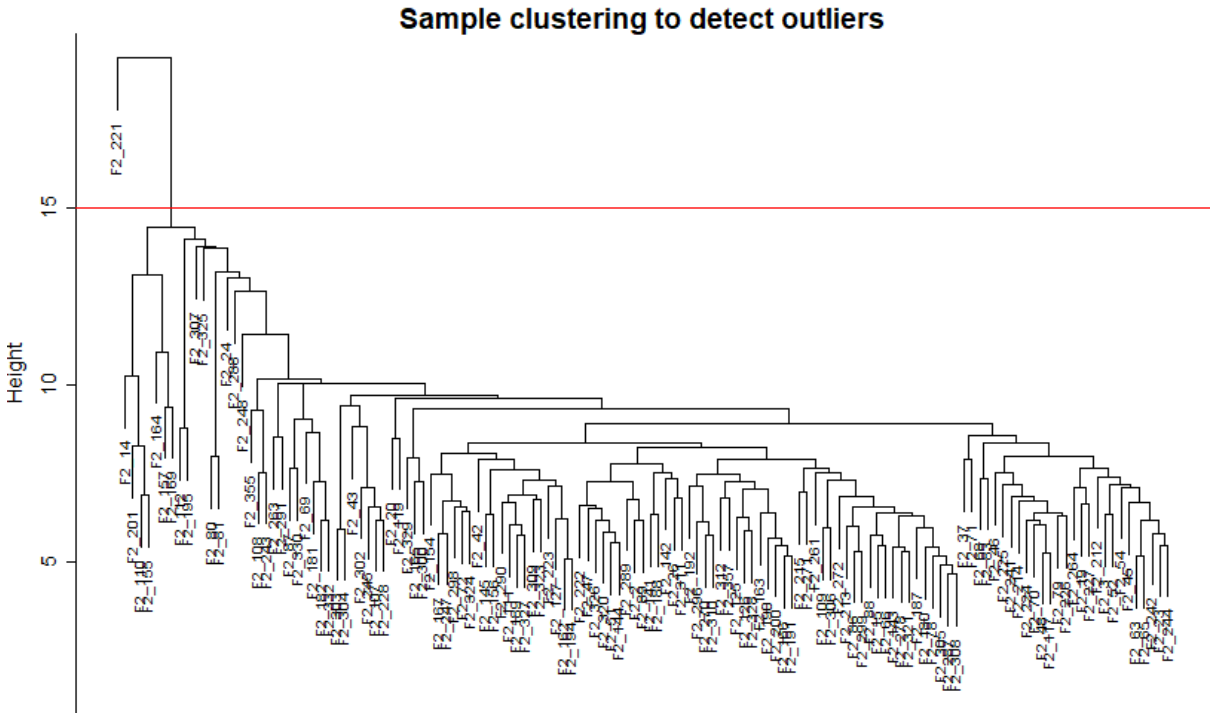

Figure 10 - A clustering dendrogram of samples based on Euclidean distance.
F2_221 is the outlier and is omitted from the data.

From Figure 10, we can see that an outlier is clearly visible. We look to identify cases such as this and remove those points from the data. We then have the expression data required for network analysis.

Compared to the liver data, the Yeast data contains no extreme outliers. Therefore, we choose not to perform a height cut. From here on outwards, all figures will refer to the yeast dataset unless mentioned otherwise.

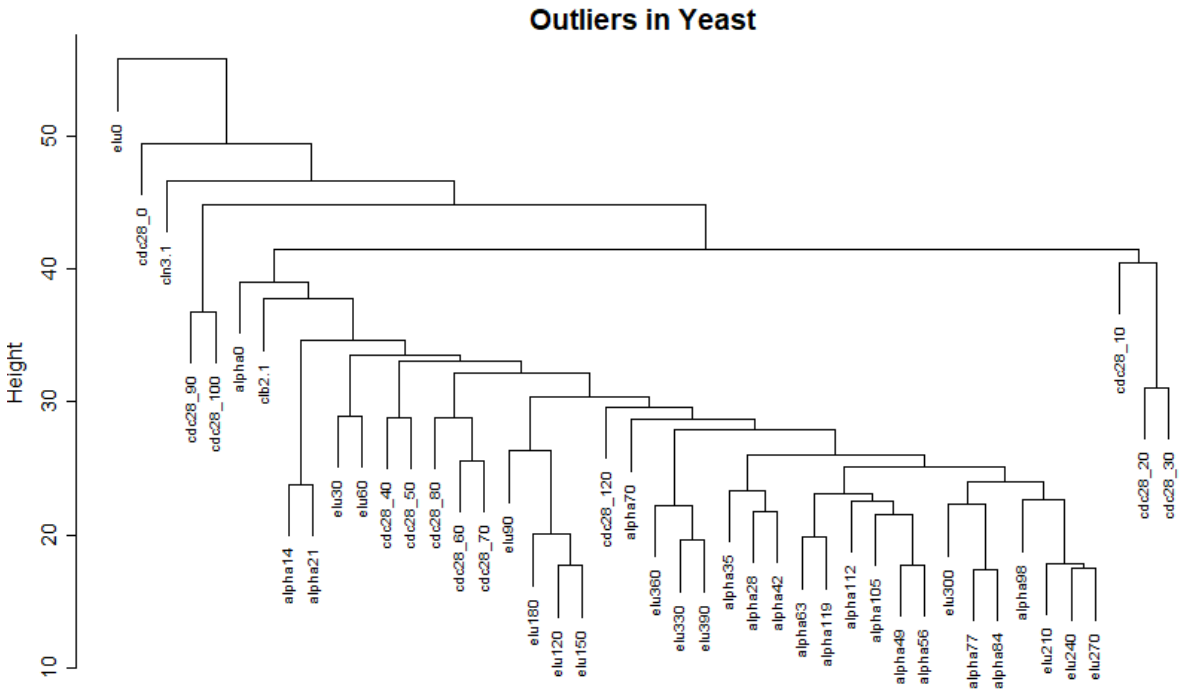

Figure 11 – Clustering dendrogram of yeast showing no extreme outliers.

4.2. Construction of the Gene Regulatory Network and module detection

The primary interest regarding genes with similar expression patterns lies in the fact that they have a greater chance of being tightly co-regulated based on their expression levels and this in turn gives an indication of their relationship in terms of their functionalities. Constructing a network mathematically and visually will aid us in identifying which genes are highly co-related and regulate the expression of other genes. Additionally, we can decipher which pathway they belong to and the different interactions between each.

Since our chosen measure is correlation, a proper coefficient of correlation needs to be selected. While multiple measures exist within the WGCNA package itself, the Pearson coefficient is likely to be most useful as it is the standard and default measure.

An adjacency matrix consisting of the correlation values is created and is later used to visualize the network. However, to amplify the disparity and easily identify between strong and weak correlations, the values are raised to a certain value to which the similarity or dissimilarity results will be raised to. For example, the following results are raised to a value of 4.

$$cor(i,j) = 0.8$$

$$cor(k,l) = 0.2$$

New values: $|0.8|^4 = 0.4096$

$|0.2|^4 = 0.0016$

As we can see, the 4-fold difference between 0.8 and 0.2 has been amplified to a 256-fold difference. With our newfound values, we are ready to construct a fully connected network consisting of genes as nodes and the edge weights from the adjacency matrix.

To construct a gene regulatory network from the data, a soft thresholding power has to be chosen, to which the co-expression similarity will be raised to in order to calculate adjacency. This power is selected based on the criterion of approximate scale-free topology. A function called pickSoftThreshold() analyses the network topology and helps the user pick the power from a set of candidate powers as shown in Figure 12.

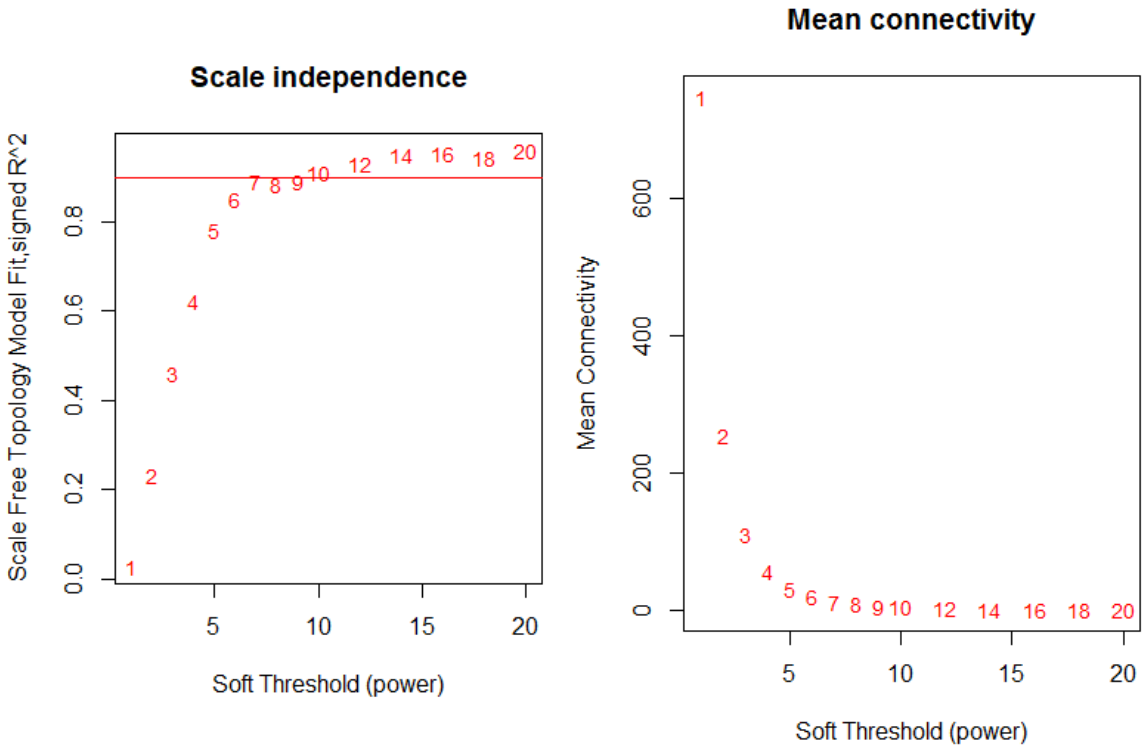

Figure 13 - Selecting a power for the soft threshold.

We aim for a scale independence of 0.9, therefore, 7 is chosen as the soft threshold power, due to it being the lowest power for which the scale-free fit index reaches 0.9. Using this, we generate an adjacency matrix raised to the power 7 from the expression data. This is the preliminary form of our network. The mean connectivity drops as power increases, so it is made certain that the connectivity does not drop too low.

Our focus is on making the network scale-free which allows to extend the scalability depending on the dimension of the data-set. A scale-free network consists of many nodes which are the genes in our case while incorporating as few connections as possible. The degree distribution follows a power law where the probability for a node having k connections is k raised to some power.

We then transform this adjacency matrix into a Topological Overlap Matrix; this reduces noise and false associations. The Topological Overlap Measure is a pairwise similarity measure between the network nodes or the genes. A high measure between two genes indicate that they have many shared neighbors. Thus, we can concur there exists a large overlap of their network neighbors or that the genes have similar expression patterns.

The process of calculating the matrix values is calculated from the following formula:

$$TOM_{ij} = \frac{\sum_u a_{iu} a_{uj} + a_{ij}}{min(k_i, k_j) + 1 - a_{ij}}$$

$$DistTOM_{ij} = 1 - TOM_{ij}$$

Since we are concerned with the dissimilarity between different genes, a dissimilarity value is calculated by subtracting the TOM result from 1. The dissimilarity value is then normalized to 0 and 1 where they indicate identical set of neighbors and no overlap of network neighbors respectively.

The matrix can be constructed using the TOMSimilarity() function, and consequently, a dissimilarity matrix is produced from this TOM to identify clustered genes. Following this, hierarchical clustering, using the hclust() function, is done to produce a dendrogram as shown in Figure 14. In the diagram, each leaf corresponds to a single gene.

Highly co-expressed genes group together in branches, from which we can determine modules by separating branches through a process known as Dynamic Tree Cutting. Then we get different modules of highly co-expressed genes which are called “gene modules” and can provide extensive biological insight.

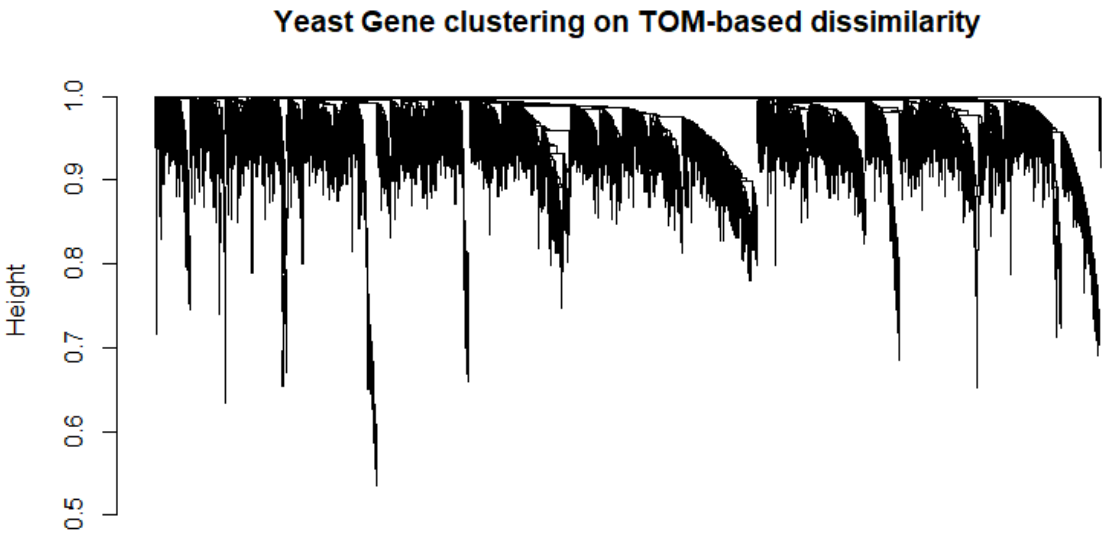

Figure 14 - Clustering dendrogram of genes, with dissimilarity based on topological overlap.

Once we have the gene modules, we can visualize them through assignment of different colors. Afterwards, similar modules can be merged by performing a form of PCA to find the "eigengenes" which will contain the most information within each module and can be transformed into a one-dimensional data vector. This is beneficial in the sense that we take only a representative of each module instead of its entirety and can be performed using the moduleEigengenes() function. Clustering the eigengenes can give us the co-expresseion similarity based on correlation. The clustered eigengenes and appropriate cut are shown below in figure 15.

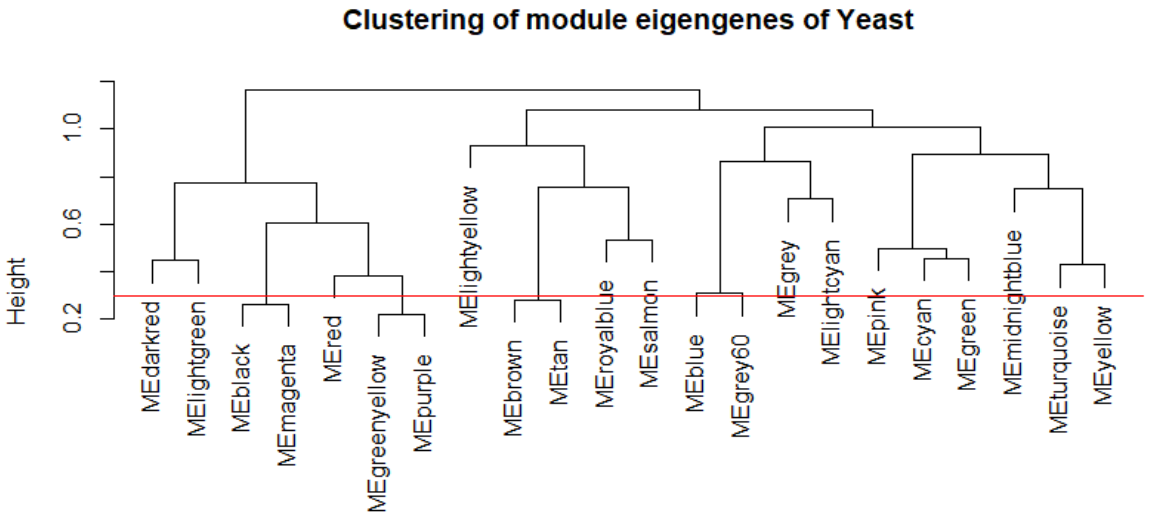

Figure 15 - Clustering of eigengenes, with dissimilarity based on topological overlap

Based on the eigengenes, we merge the modules with high co-expression similarity. The resulting modules are shown below in figure 15. Modules obtained can be used in mathematical operations to find out the degree of correlation between them or with other external traits.

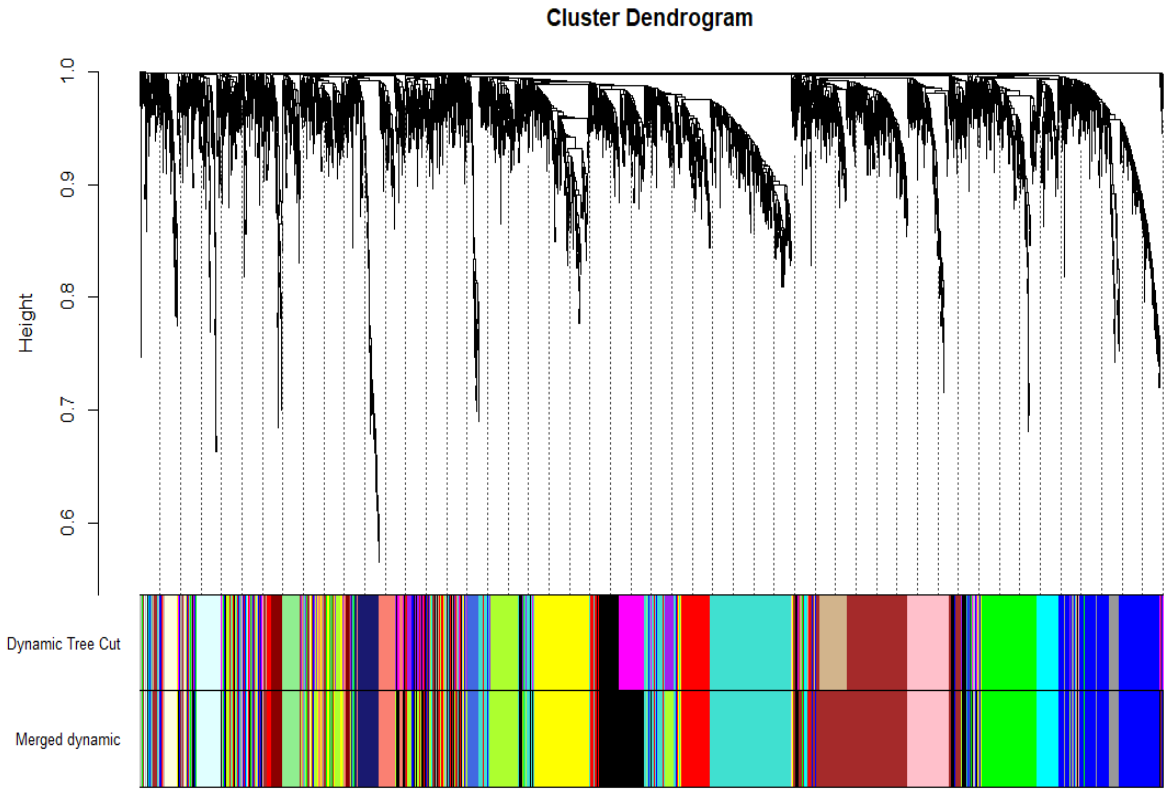

Figure 16 - Merging on highly coexpressed modules with assigned module colors and merged colors

Now we have the clustered tree along with modules, their eigengenes and other necessary information which can be used to construct necessary networks. It is also possible to relate said modules and their constituent genes with external traits and observe their relationship through the form of heatmaps. Gene significance can also be inferred. For now, we focus on constructing a network and analyzing its accuracy with existing ones.

4.3. Visualization of the network

Now that we have the network from the Topological Overlap Matrix, we can visualize it through functionalities provided by the package or exporting the matrix to external compatible software.

A heatmap can be constructed using the TOMPlot function which plots a heatmap with all the genes positioned on the row and column of the graph. The colors are an indication of the coexpression between the genes. Lighter colors indicate low adjacency and darker indicates higher adjacency or overlap. Additionally, we add the gene dendogram and module colors to indicate which module the genes belong to and show that the ones within a module are highly coexpressed.

The following figure depicts the said heatmap. It can be seen that the colors are darkest in the diagonal which indicates the modules. Throughout the matrix there are patches of lighter yellow colors and progressively darker red colors to identify which genes are highly coexpressed and which are not.

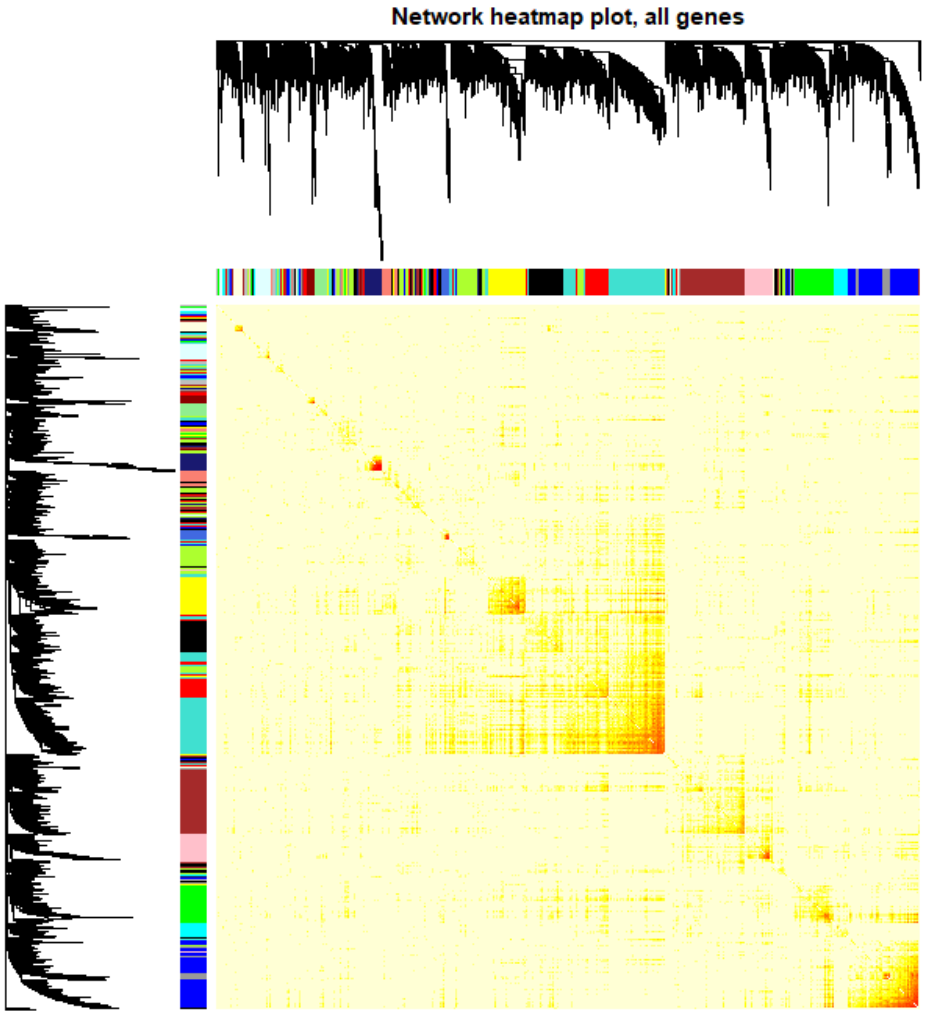

Figure 17- Visualizing the gene network using a heatmap plot. The heatmap depicts the Topological Overlap Matrix (TOM) among all genes in the analysis. Light color represents low overlap and progressively darker red color represents higher overlap. Blocks of darker colors along the diagonal are the modules. The gene dendrogram and module assignment are also shown along the left side and the top.

Since it is hard to decipher crucial information from a heatmap, we proceed to transfer the network to other software with robust visualization options to aid us in extracting valuable details. Such a software we have used is VisANT.

Here we first show the network of a single module within our network. It is possible to display a certain number of genes if needed or show relationship between modules.

We can vary the threshold before exporting to eliminate some weights which have low values.

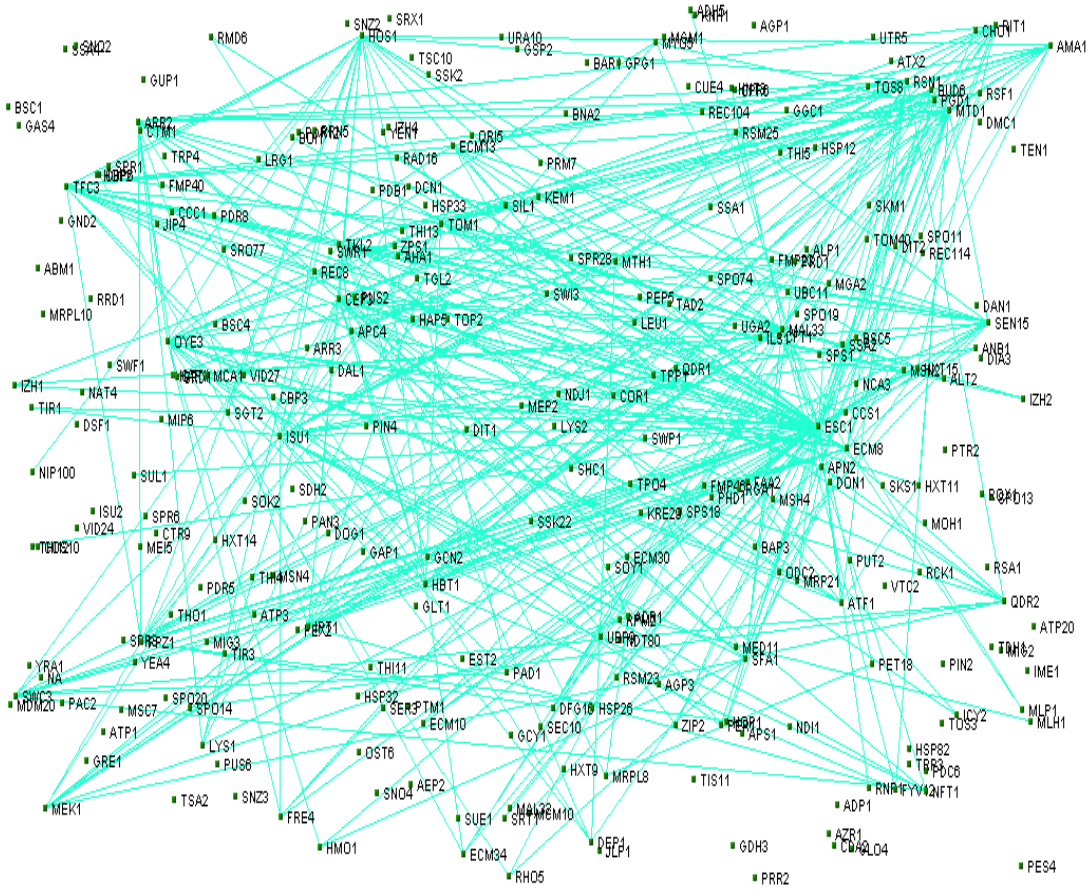

Figure 18- Network of genes in the 'brown' module of the yeast cell cycle

5. Comparative Analysis

Using our constructed network, we can extract a sub-network which is composed of 14 genes. These 14 genes are named FUS3, SIC1, FAR1, CDC6, CDC20, CDC28, CLN1, CLN2, CLN3, CLB5, CLB6, SWI4, SWI6 and MBP1 and they are known to be involved in the early cell cycle of the yeast *Saccharomyces cerevisiae*. The cell cycle describes the series of events that precedes its division and duplication. Larvie et al have produced the following gene regulatory network through their method and the details of their results are given below. [4]

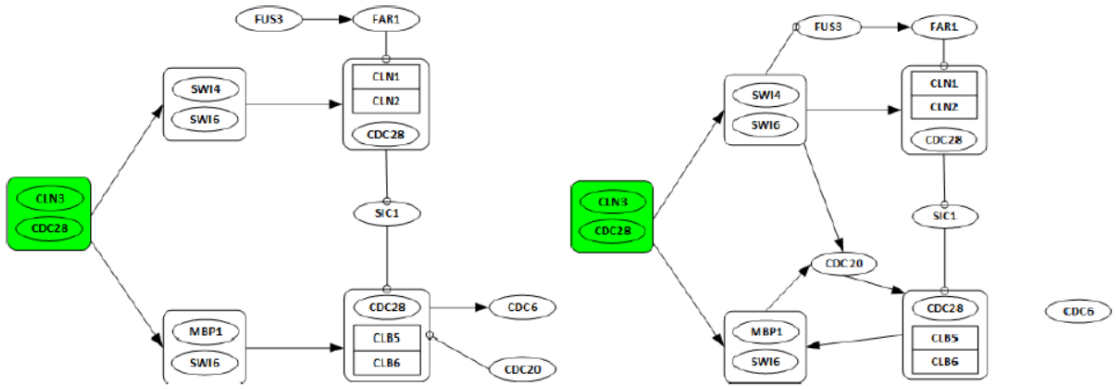

Figure 19- The figure on the left shows the known GRN of the early cell cycle of yeast. On the right, the recovered GRN using the method described by Larvie et al.

The recovered network contains complexes including one or several genes which are considered as a 'gene' in the network. There are 10 complexes, including CLN3/CDC28, SWI4/SWI6, MBP1/SWI6, CLN1/CLN2/CDC18, and CLB5/CLB6/CDC28. Other nodes that are made of one single gene only, CDC20, CDC6, SIC1, FAR1, and FUS. The following assumptions are made:

1. Genes CLN3 and CDC28 are only considered as possible regulators, as they are starters of the cell cycle network.

2. All discovered links from any gene in one complex to any other genes in a different complex are considered as a single regulation.

3. All regulations among genes in the same complex are ignored.

Through the use of weighted gene coexpression, we have constructed a network of 4000 genes which include the 14 involved in the early cell cycle development. The datasets include their expression levels at different times, similar, but not exact to the one used by Larvie et al. Our constructed network is shown in Figure 20 followed by an alternative simplified representation in Figure 21.

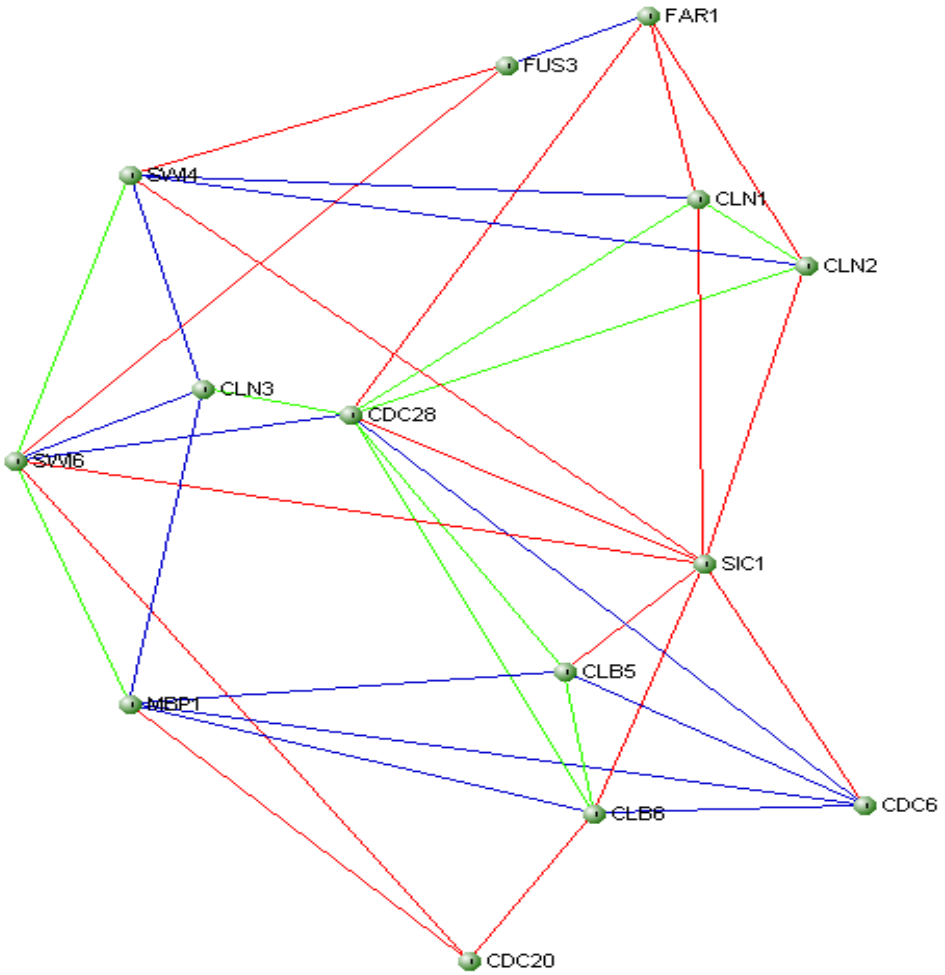

Figure 20- Constructed GRN from weighted coexpression network. Red indicates inhibition, Blue indicates activation, and Green indicates membership within the same complex.

The weight coefficients of the edges created from the Topological Overlap Matrix were related to the regulation of the genes. Coefficients for activation and inhibition lie within their respective ranges of threshold and they have been colored blue and red respectively for simplicity of understanding. Grouping of genes which constitute a complex is indicated by green edges and are highly coexpressed.

The same network is remodeled in light of the ones depicted in Larvie et al. The true positives and false positives have been identified along with comparative measures. This can be done due to this pathway being known and recorded in the Kyoto Encyclopedia of Genes and Genomes (KEGG). This allows us to calculate specificity, sensitivity and precision for both our results and the existing results in order to compare them.

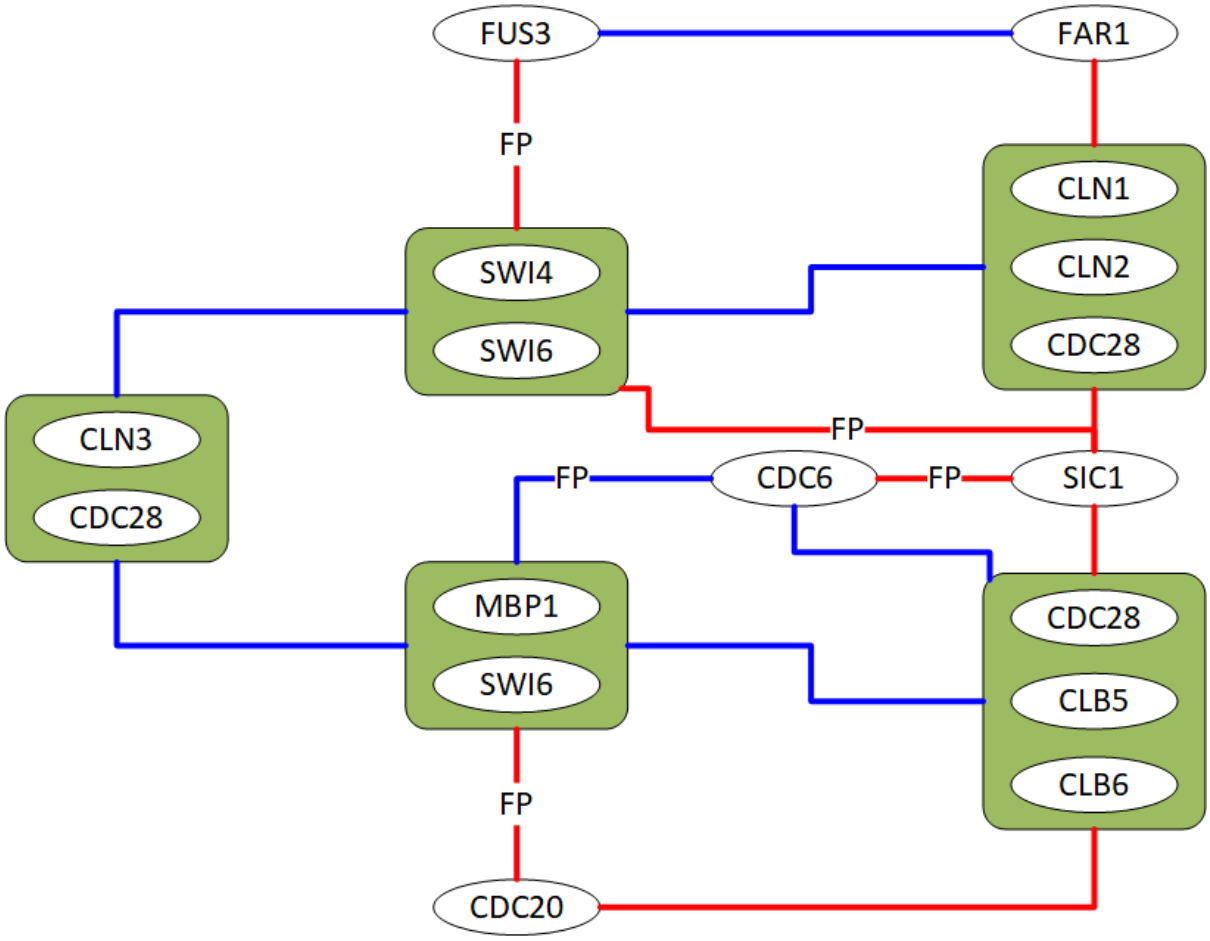

Figure 21 - Pathway drawn from the constructed network.
Blue indicates activation, Red indicates inhibition. False Positives are marked as such.

The comparative analysis is shown below:

	TP	FP	TN	FN	Sensitivity	Specificity	Precision
WGCNA	10	5	76	0	1	0.938272	0.666667
LASSO-VAR	7	3	80	1	0.875	0.963855	0.7

From the table, it can be seen that WGCNA obtained a higher sensitivity due to correctly identifying all the pathways. However, a significantly higher number of false positives were also obtained, reducing the specificity and overall precision. Furthermore, as the sample size for this calculation is quite small, it is safe to assume the sensitivity of WGCNA will go down with larger samples.

We were able to identify the CDC6 interaction correctly, however, we also encountered several false positives, particularly on inhibiting relations.

6. Motivation and Future Work

We elected to use WGCNA as our method due to several advantages it provides. Not only does it cluster samples based on expression, it further defines modules of highly interconnected genes and provides additional network statistics. Separating large networks into modules also allows external traits to be compared to a smaller number of values, greatly increasing scalability.

Some drawbacks of the LASSO method include the fact that due to being a type of model selection, it tends to eliminate variates from the system by shrinking coefficients to 0 as it progresses. This does not help if prediction is the primary focus as it would tend to suffer the drawbacks of predictive discrimination. For example, if the sample has some relevant genes and some genes unimportant to the study at hand, LASSO works well as it would eliminate the unneeded genes from the model. However, in certain studies, such as vibrational spectroscopic data sets, tend to have data spread out over large ranges, in which case dropping variates would be undesirable. For two highly correlated variates, LASSO may also drop one. In cases such as these, ridge regression is a better option. It may be possible to improve the process if these factors were also taken into consideration.

However, from our analysis, we have determined that under most circumstances, the LASSO-VAR approach is the optimal choice for the modeling of Gene Regulatory Networks. Therefore, we aim to devise an approach to improve on the work done regarding this approach. One alternate that is used in certain cases is ridge regression, a stepwise variable selection method.

Furthermore, LASSO-VAR performs without the use of prior knowledge which is favorable in most cases, but whether or not the results can be further improved by including prior knowledge merits further study as this could be one possible way of improving the predictive power of the method.

Upon researching on the strengths and weaknesses of VAR, a method initially developed for use on economic models, we came across Bayesian Vector Autoregression (BVAR). BVAR is similar to VAR, except that the model parameters are treated as random variables with prior probabilities assigned to them. It also performs well on modelling large datasets, making it an ideal candidate for further study with regards to Gene Regulatory Network modelling. We believe combining the findings of the papers analyzed with the techniques proposed would allow us to achieve our objective – a version of LASSO-VAR that can adapt to the introduction of prior knowledge and potentially improve on the results, **LASSO-BVAR**.

To utilize BVAR however, we will require datasets containing probability values. We can also get better results from WGCNA itself if trait data for a given dataset is also incorporated. The trait data can be used in tandem with the module data to identify which modules are responsible for which traits. Doing so would allow us to determine which gene pathways are responsible for which physical characteristics. For example, the trait data of the liver dataset [6] can be incorporated to form a module-trait relationship as shown in the following figure.

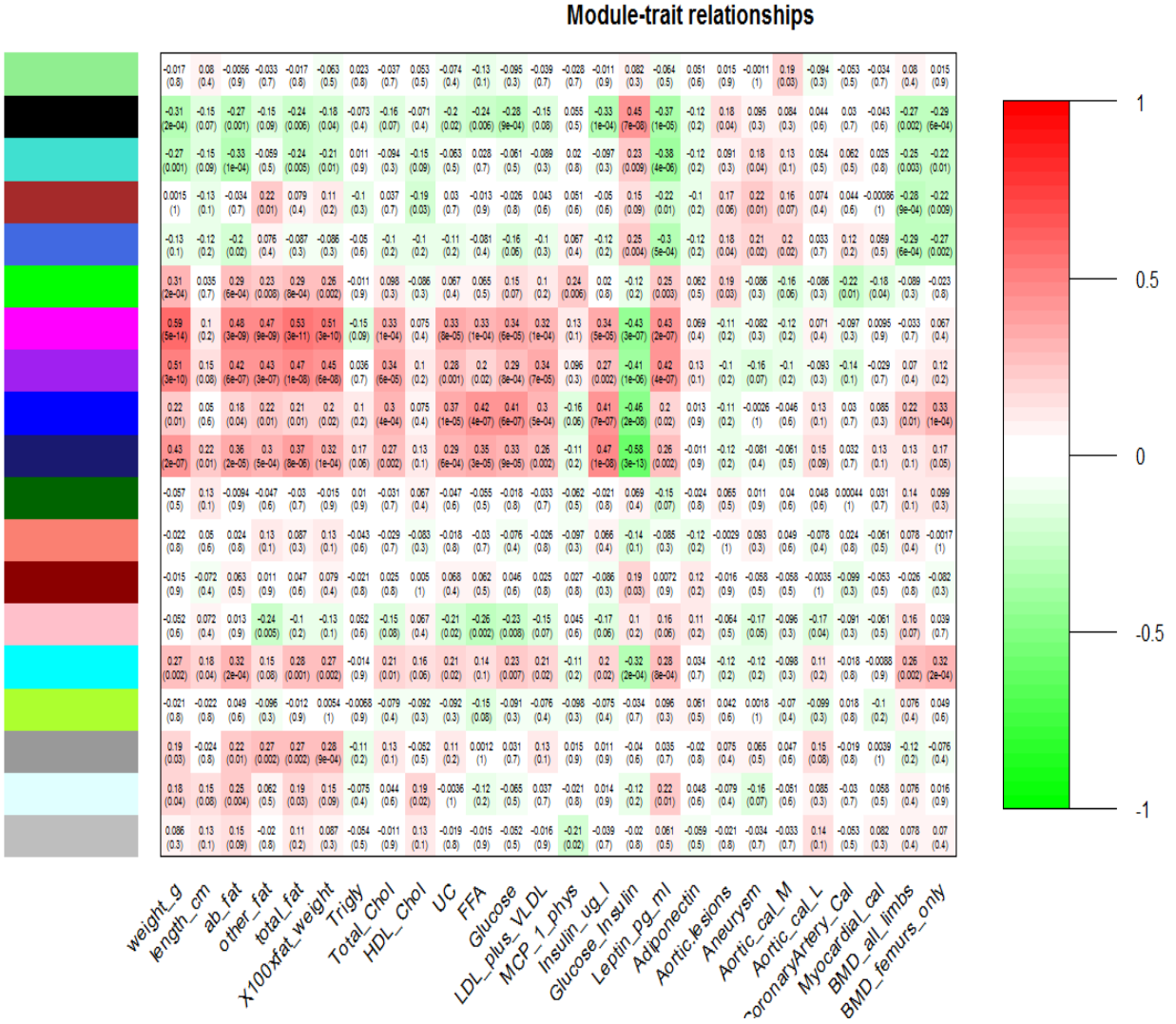

Figure 22- Module-trait relationship of mice liver dataset. The more red a value is, the more the trait is expressed by the module.

References

[1] L. E. Chai, S. K. Loh, S. T. Low, M. S. Mohamad, S. Deris and Z. Zakaria, "A review on the computational approaches for gene regulatory network construction," *Computers in Biology and Medicine,* pp. 55-65, 2014.

[2] N. Vijesh, S. K. Chakrabarti and J. Sreekumar, "Modeling of gene regulatory networks: A review," *Journal of Biomedical Science and Engineering,* pp. 223-231, 2013.

[3] M. M. Zavlanos, A. A. Julius, S. P. Boyd and G. J. Pappas, "Inferring Stable Genetic Networks from Steady-State Data," *Automatica,* pp. 1113-1122, 2011.

[4] J. E. Larvie, M. G. Sefidmazgi, A. Homaifar, S. H. Harrison, A. Karimoddini and A. Guiseppi-Elie, "Stable Gene Regulatory Network Modeling From Steady-State Data," *Bioengineering,* p. 12, 2016.

[5] P. Langfelder and S. Horvath, "WGCNA: an R package for weighted correlation network analysis," *BMC Bioinformatics,* 2008.

[6] A. Ghazalpour, S. Doss, B. Zhang, S. Wang, C. Plasier, R. Castellanos, A. Brozell, E. E. Schadt, T. A. Drake, A. J. Luis and S. Horvath, "Integrating Genetic and Network Analysis to Characterize Genes Related to Mouse Weight," *Plos Genetics,* 2006.

[7] P. Spellman, G. Sherlock, M. Zhang, V. Iver, K. Anders, M. Eisen, P. Brown, D. Botstein and B. Futcher, "Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization," *Molecular Biology of the Cell ,* 1998.

[8] V. A. Huynh-Thu and G. Sanguinetti, "Gene Regulatory Network Inference: An Introductory Survey," *arXiv - Quantitative Biology,* 2018.

[9] T. S. Gardner, D. d. Bernardo, D. Lorenz and J. J. Collins, "Inferring Genetic Networks and Identifying Compound Mode of Action via Expression Profiling," *Science,* no. 301, pp. 102-105, 2003.

[10] G. Michailidis and F. d'Alché-Buc, "Autoregressive models for gene regulatory network inference: Sparsity, stability and causality issues," *Mathematical Biosciences,* no. 246, pp. 326-334, 2013.

[11] C. Panse and D. M. Kshirsagar, "Survey on Modelling Methods Applicable to Gene Regulatory Network," *International Journal on Bioinformatics & Biosciences (IJBB),* vol. 3, no. 3, pp. 13-23, 2013.