Unsupervised Reconstruction of Gene Regulatory Network

- CS 598 SS Final Project Report

Liming Wang
Wittney Mays

1. Introduction:

Living cells have a dynamic environment that involves the interaction of many molecules. The systems within these living cells have many molecular interactions, thus there are many types of molecular networks including protein-protein interactions, gene regulatory networks, gene co-expression networks, metabolic networks, signaling networks. In this study, we will focus on gene regulatory networks. Gene regulatory networks (GRNs) reveal the processes which drive cellular functions and the genes expressed within a cell [4] . Gene regulatory networks represent sets of molecular regulators that directly and indirectly interact to regulate gene expression [5]. The process of the decision which controls the rate at which genes in the network are transcribed into mRNA is largely due to the behavior of related genes in a GRN. Often times biologists perform experiments (i.e. CHIP-seq, knockout) to better understand the steady-state of a cell and reveal these related genes, but this technology is costly and time consuming. With recent technology advances in biological techniques, the ability to generate high throughput data can provide a better understanding of the mechanisms within biological systems that are now attainable. Data from gene expression studies coupled with computational methods provide the necessary framework to model gene regulatory networks. Reconstructing gene regulatory networks (GRNs) from experiments has resulted in a significant increase in understanding systems biology.

Regulation of gene expression is correlated with the amount of gene products; therefore, most mathematical models attempt to capture the relationships between the regulation of gene expression to understand the complex process of regulation [6]. The goal of reconstructing GRNs is to create abstract models to mimic actual biological processes. The GRNs inference through mathematical modeling is highly dependent on the assumptions made in the model, thus producing a dynamic model with minimal assumptions provide the flexibility to statistically reconstruct the molecular interactions. Some of the most common GRN models are based on popular statistical methods including information theory, boolean networks, differential equations models, bayesian statistics, and neural network models [7]. Each approach described often have limitations based on the assumptions, which could lead the inability to capture novel information of the biological systems behavior. The application of feature selection, accurate parameter estimation, structure optimization and prior information integrated into the model circumvents the limitations of each statistical method. All methods provide advantages and disadvantages but assessing which methodology to use is dependent upon the data included inside the model. Often the use of different data types create a relative amount of noise, thus creating a model that can infer accurate relationships between genes can be difficult [8].

Regression-based models are among the most simplistic approach to implement and scale for the reconstruction of networks [9]. GRN models constructed with the use of regression models can accurately model relationships based on the changes in gene expression levels. Simplified model and the ability to predict a network under multiple conditions are primary advantages of regression-based models. A few disadvantages of the regression models include the limitation of the size of the network size, limitation of modeling linear functions, model parameters that may overestimate a relationship, and the inability to circumvent data with a large amount of noise. One methodology used to circumvent many of the disadvantages accompanied by regression based models is through the integration of prior information from experiments (i.e. CHIP-seq, knockout gene expression, motif prediction), though the use of prior information. Although the incorporation of supporting prior information could potentially increase the amount of noise and contribute to the difficulty of identifying true genetic patterns.

The easy accessibility to -omic data has contributed great insights for the modeling of biological systems. In regards to the reconstruction of regulatory networks, previous literature has inferred that the inclusion of omics data as prior information may contribute to the increased ability of mathematical models to predict associations between genes and regulators. The quality and robustness of expression data, typically acquired through NGS technologies, has significantly decreased the amount of noise allowing the identification of linear and non-linear relationships in studies. Although the inclusion of additional experimental data in gene regulatory network models as prior information could be advantageous in some cases, it contributes to the amount of noise captured with incorporating more data and increases the non-linear associations observed. Though the incorporation of prior information has been successful in established GRN software, an observation that is consistent between each software is method is often limited to small gene expression data sizes. Thus understanding the limitations of each mathematical model and understanding which software to use based on known limitations has been the approach for most biologist.

In this study, we implemented a LASSO regression-based method that includes a penalty for prior information. We will investigate the value including the prior information and which characteristics of this prior information provide better potential in training and supervising of the regression-based model. Similar to the baseline LASSO regression model initially implemented, but this model incorporates an additional penalty based on prior information from empirical data, which leads to a sparse and more definite estimation of edges in GRNs. The aim of this research is to develop a new and more accurate method for the construction of GRNs containing the important relationships, presented as edges, which eludes to relationships that play essential roles in biological systems. In this study, we will also discuss the limitations of our methods, which would be more appropriate for application.

2. Problem Formulation:

A gene regulatory network (GRN) $G = (V, E)$ is a directed, acyclic graph with n genes as its nodes, and an edge from i to j represents gene i is a transcription factor(TF) of gene j. For each gene, d expression profiles $x_i = (x_i^1, ..., x_i^D)$, $i = 1, ..., n$ are observed. Further, G is assumed to be *modular*, which means that G can be separated into modules of genes $M_1, ..., M_m$. Within each module, genes are more likely to be regulated by the same set of TFs than when they are from different modules. A GRN reconstruction algorithm then takes as inputs, the expression profiles, X, and outputs the optimal edges of the GRN G^*. To this end, the model tries to maximize the posterior probability of the GRN given the observed expression profiles X:

$$\Theta^* = argmax_\Theta P(\Theta|X) = argmax_\Theta \sum_{G, M} P(\Theta)P(M|\Theta)P(G|M, \Theta)P(X|\Theta, G) \quad (2.1)$$

Since (2.1) is hard to optimize directly due to the summation over all module assignment and graph M, G, we can instead maximize over the posterior probability over all hidden variables and assume that $P(\Theta|G)$ is uniform across the possible range of Θ, that is, no parameter values are preferable given only the GRN. In this case, (2.1) is simplified to be:

$$M^*, G^*, \Theta^* = argmax_{G,\Theta} P(M)P(G|M)P(X|\Theta, G) \quad (2.2)$$

Such formulation is called the *greedy* formulation. From (2.2), the problem now boils down to the modeling of the GRN prior probability and the likelihood of the expression profiles. For the graph prior probability, a natural assumption will be that the edges for GRN are generated independently given the modules M. To slightly abuse the notation, we also use G to denote the set of edges of G. Let $P(i \to j|M)$ denotes the probability that gene i is regulating j given the information about modules, we have

$$P(G|M) = \prod_{(i,j)\in G} P(i \to j|M) \prod_{(i,j)\notin G} (1 - P(i \to j|M)) = \prod_{i=1}^{n} P(i \to j|M)^{b_{ij}} (1 - P(i \to j|M))^{(1-b_{ij})} ,$$

(2.3)

where b_{ij} is the indicator function whether (i,j) is an edge of G, and $P(i \to i|M) = 0, b_{ii} = 0$.

For the observation probability, the gene expressions from different experiments, $x^1, ..., x^D$ are assumed to be independent and can be modeled using the same model:

$$P(X|\Theta, G) = \prod_{d=1}^{D} P(X^d|\Theta, G) \quad (2.4)$$

To model joint probabilities $P(X^d|\Theta, G)$, we need to find a way to incorporate the network information into the probability. One strategy will be to treat the directed graph G as an *undirected* graph G' by connecting an edge between two genes if one of them is regulating the other and connect between genes that co-regulate the same gene. Such an assumption allows

simpler modeling of the joint probabilities at the price of losing information about the gene regulatory directions. While the generative model does not depend on the direction of the regulation, the regulatory relations can often be found during the inference step, using heuristics such as weight comparison. To evaluate the performance of the algorithm, we can generate all possible directed graph given the undirected graph and compare with the gold standard network.
Alternative model will be to use the directed information of G during the modeling of $P(X^d|\Theta, G)$. Since to infer the regulatory relation amounts to learn a causal relation between the change of expression level between genes, and generally requires additional information such as the order of the gene expression events, or requires additional modeling assumptions, we can instead *approximate* $P(X^d|\Theta, G)$ as follows. Suppose we have information about the order of the expression events and the expression of each gene at time t depends only on the expression of gene in the previous time step $t-1$. Let the expression at time t be $X^{(t)}$, and R_i be the set of TFs for gene i, and let X_A be the expression profile of genes in the set A, then we have:

$$P(X^{(t+1)}|X^{(t)}, \Theta, G) = \prod_{i=1}^{n} \prod_{j \in R_i} P(X_i^{(t+1)} | X_{R_i}{}^{(t)}, \Theta) \tag{2.5}$$

If we use $X^{(t+1)}$ itself as a proxy for $X^{(t)}$, we have approximately that:

$$P(X^{(t+1)}|X^{(t)}, \Theta, G) \approx \prod_{i=1}^{n} \prod_{j \in R_i} P(X_i^{(t+1)} | X_{R_i}{}^{(t+1)}, \Theta). \tag{2.6}$$

Since the right-hand side does not depend on $X^{(t)}$, we have:

$$P(X_i^{(t+1)}|\Theta, G) \approx \prod_{i=1}^{n} \prod_{j \in R_i} P(X_i^{(t+1)} | X_{R_i}{}^{(t+1)}, \Theta). \tag{2.7}$$

In words, we treat the expressions of the transcription factor as if it is expressed *before* the target gene, in which case the conditional probability of X_i at time $t+1$ depends only on the expression of its parents at time t.

Looking back the original problem (2.1), we can decompose the probability differently as:

$$\Theta = argmax_\Theta P(\Theta) \sum_M P(M) P(X|M, \Theta) \approx argmax_\Theta P(\Theta) \sum_M P(M) \prod_{i=1}^{n} P(x_i|X, M, \Theta), \tag{2.9}$$

Such a formulation is called the *module-based* formulation. Since M forms an undirected graph, we can perform clustering on the genes to find the optimal M and remove summation. In this case, the GRN can be found using the weight parameters in Θ using the heuristic that they are confidence score of the edges. If every module contains a single gene, (2.9) becomes simply:

$$\Theta = argmax_\Theta P(\Theta) P(X|\Theta) \approx argmax_\Theta P(\Theta) \prod_{i=1}^{n} P(x_i|X, \Theta), \tag{2.10}$$

Such a formulation is called the *module-free* formulation.

3. Model Description:

3.1. LASSO Regression

To understand how LASSO can be applied to the problem of GRN, notice the module-free formulation (2.7) can be rewritten as:

$$X_i^{(t+1)} = f_i(X^{(t)}{}_{G\setminus\{i\}}) + \varepsilon_i \approx f_i(X^{(t+1)}{}_{G\setminus\{i\}}) + \varepsilon_i,\ i = 1, ..., n,$$

(2.8)

where ε_i is some noise independent of X, and f_i is a deterministic function between the regulators and the target gene i. Using this formula, reconstruction of GRN can be decomposed as n sub-problems of finding the TFs of each gene, $R_1, ..., R_n$, by learning $f_1, ..., f_n$.

In the model, the least absolute selection and shrinkage operator (LASSO) contributed to the estimation of the *sparsity* parameter in the model. The estimated predictor variable β, which is equal to the number of nonzero coefficients in β. However, without prior knowledge about how sparse β. Therefore, we set a hard constraint on the sparsity of β, to minimize the following objective function:

$$\widehat{\beta}^{LASSO} = argmin_{\beta} \frac{1}{2} \sum_{i=1}^{n} (y_i - x_i{}^T \beta)^2 + \lambda ||\beta||_1, \qquad (3.1.1)$$

where $||\beta||_1 := \sum_{k=1}^{d} |\beta_k|$ and λ is a hyperparameter chosen to control the importance between the two terms. From the form of the estimator, we can see that if $y_i = 0$, as long as the noise level is below $|a_i \lambda|$, the LASSO will always give the correct prediction, while the ordinary least square solution will swing as the noise level changes.

We can also use the module-based formulation by incorporating the prior information about the GRN into LASSO as follows. Suppose from other measurements, we determine that an edge between gene i and gene j is $p_{0,ij}$ and let P_0 be the $n \times n$ matrix of all such priors. Further, as stated in the problem formulation, the network is modular, meaning large set of genes are the targets of the same TF. It is natural to have a prior that encourage the assignment of a gene to a TF if most of the members in the module are the targets of that TF. The experimental information and the module information can be incorporated into the model as parameters of the graph-based prior:

$$P(i \to j | M,\ P_0) = \frac{p_{0,ij}}{1 + exp(-(p + r * f_{ij}))},$$

where p is the sparsity constraint and f_{ij} is the normalized frequency of genes in the module of i are inferred to regulate genes in the module of gene i in the previous iteration. To use prior information in the LASSO, notice that the parameter λ_{ij} roughly controls the threshold the

weight of an edge has to go beyond to exist in the network. And the probability of an edge to be zero is roughly $\int_0^{\lambda_{ij}} p(w)dw$. if $p(w)$ is approximately standard gaussian, $\int_0^{\lambda_{ij}} p(w)dw = 1 - Q(\lambda_{ij}) \approx 1 - exp(-\lambda_{ij}{}^2/2) = 1 - P(i \to j|M, P_0)$. This intuition suggests the choice $\lambda_{ij}{}^2 = -ClogP(i \to j|M, P_0)$. In practice, we found that $\lambda_{ij} = \lambda\,(-logP(i \to j|M, P_0))^{\gamma}$ works well, where $\gamma > 0$ is called the confidence power, and larger γ places more emphasis on the prior network.

Therefore, we modify the objective (3.1.1) as:

$$\widehat{\beta j}^{LASSO} = argmin_{\beta} \frac{1}{2n} \sum_{i=1}^{n} (x_j - X_{\setminus j}{}^T \beta)^2 + \sum_{i=1}^{n} \lambda_{ij} |\beta_i| , \qquad (3.1.1)$$

Another way to incorporate module information is to find it using some clustering algorithms and treat it as *constraints* for the LASSO regression. That is, we assume the module is given and for genes inside the module, we assume there coefficients to be similar. To find the module, we can assume that genes within the same module tends to have similar expression profiles and therefore can be determined by standard clustering algorithms. To enforce the module constraint, we can use the grouped LASSO [14] algorithm:

$$(\beta_i{}^*)_{i=1,\ldots,n} = argmin_{\beta_i} \frac{1}{2n} \sum_{i=1}^{n} (x_i - X_{\setminus i}\beta_i)^2 + \sum_{i=1}^{|M|} \lambda_i \|\beta_{M_i}\|_g , \qquad (3.1.2)$$

Where $\|x_G\|_g := (\sum_{i \in G} \|x_i\|^2{}_2)^{1/2}$ is the group norm, which is the L_2 norm of the concatenated vectors of the group. Intuitively, this objective forces the predictors for genes in the same module to be almost the same.

Further, we can extend this model by using a generalized notion of the modules to allow different *levels* of modularity. This generalization is necessary when most genes in the datasets are regulated by more than one TF and as a result, exact match of the TF set seems far-fetched. Instead, the genes may match by none, one, two or more of the TFs. This amounts to form a *hierarchical* set of clusters, for example, by using agglomerative clustering. The objective is in the same form as in (3.1.2), except that M_i needs not to be disjoint but a tree-like structure. Further, we can assign importance to different levels of the clusters by tuning the regularization weights. As shown in [14], we can define the "tree-structure" sparsity norm as recursively as follows:

$$\|\beta_G{}^*\|_{g,\,tree} := v_G \sum_{G_i:child\ of\ G} \|\beta_{G_i}{}^*\|_{g,\,tree} + \lambda_G \sum_{i \in G} \|\beta_i\|_1 , \qquad (3.1.3)$$

And by specifying the set of (ν_G, λ_G) we can determine the relative importance between individual sparsity and group sparsity.

3.2. Greedy gene-wise selection:

Following the greedy objective (2.2), we can find the optimal module and graph incrementally based on some confidence score. For the module, we can simply use some existing clustering algorithms. For the graph, we can use as confidence scores the log-likelihood computed with (2.3) and (2.7):

$$score(R_i \cup \{j\}) = log\ P(x_i|X_{R_i}, x_j,\ \Theta^*) + log\ P(j \to i|M, \Theta^*)$$
$$= logP(x_i|X_{R_i}, x_j,\ \Theta^*) - log(1 + exp(-b + r_1|R_i| - r_2 f_{ij})), \qquad (3.2.1)$$

Where b controls the initial prior probability, r_1 controls the sparsity of the graph and r_2 controls the importance of the module information. To decide the order of the gene to choose, we can use the Pearson-correlation coefficient between the current residual $r^{(t)}{}_i = x_i - X_{R_i^{(t)}}\beta_{R_i^{(t)}}$ and the candidate TF. For each gene, we continue the selection process until the score starts to decrease.

3.3. Bayesian Regression models:

One Bayesian approach suitable for GRN inference is called the "Bayesian subset selection". We start from the module-free objective (2.10), and let $\Theta = (\mu, \sigma)$. Suppose the gene expression data for the d^{th} condition $X^{(d)}$ is generated as follows [11]:

$$\beta \sim N(0,\ g\sigma^2(X^TX)^{-1})$$
$$X_i^{(d)} \sim N(X_{\setminus i}^{(d)T}\beta_{\setminus i},\ \sigma^2 I_n), \qquad (3.3.1)$$

with $P(\sigma) = \frac{\sigma_0}{\sigma^2},\ \sigma > \sigma_0$ and 0 otherwise. Expression data for each condition are generated independently. It has been shown in [11] that the posterior of β is the convex combination of β_0 and β^{LS}, the least-square estimate with g controls the relative importance. The posterior of $1/\sigma^2$ is a Gamma distribution with shape parameter $n/2$ and scale parameter $\frac{1}{2n}\sum_{i=1}^{n}||x_i - X_{\setminus i}{}^T\beta_{\setminus i}||^2{}_2 + \frac{1}{2}(\beta - \beta_0)^T(X^TX)^{-1}(\beta - \beta_0)$. The key problem is how to choose a subset of predictors that represent the regulators. To this end, we can use the Bayesian information criterion (BIC) defined as:

$$BIC(\Theta) = -\frac{k}{2}log\ n + log\ P(X|\sigma) = -\frac{k}{2}log\ n - nlog\sigma + Const\ . \qquad (3.3.2)$$

To improve robustness of the estimate, we take the conditional expectation over σ given X:

$$-E[BIC(\Theta)|X] \sim klog\ n + n(log(shape) - Digamma(scale))$$

We can then choose train 2^n regression models and choose the best k. One drawback from this scheme is, of course, training 2^n models is computationally intractable for n, say, larger than 20. To reduce the computation, we can instead use methods such as orthogonal matching pursuit to narrow the number of TFs to be within 10 and then perform the subset selection.

Under the module-based formulation, we can modify (3.3.1) as:

$$\begin{aligned} m_i &\sim P(M_i) \\ \beta_m &\sim N(0, {\sigma_0}^2) \\ {\beta^{(i)}}_{m_i} &\sim N(\beta_m, g{\sigma}^2 (X^T X)^{-1}) \\ x_i &\sim N(X_{\setminus i}\beta_{m_i}, \sigma^2 I). \end{aligned}$$

4. Dataset Description:

The data, provided by [2], contains gene expression profiles obtained using three different experimental conditions: natural variation, knockout or stress response from the same yeast GRN. We only used the genes that appear in the MacIsaac gold standard network mentioned in [2].

5. Data Preprocessing:

For computational efficiency and debugging purposes, we separate the gene expression data into a set of regulators and a set of targets. Since the two sets overlap, during the LASSO training we exclude the gene in the regulator set if it is the target gene.

6. Evaluation Metrics:

We consider the following metrics to evaluate the performance of our method and in comparison with other methods:

6.1. Gold standard:

To evaluate the gene network analysis performance, we used a gold-standard expression dataset composed of expression profiles in which their matched TF and the genes regulated by the TF are already annotated. There are three gold standard networks available for the same GRN, as shown in Table 3, [2].

6.2. AUROC and AUPR:

AUROC and AUPR were used to compare how much overlap between the predicted network and the gold standard networks. We calculate the AUROC in two different ways: the first way includes all the possible connection between genes. For example, if the gold network is A -> B,

B -> C and the predicted network is A -> B, we count both the edges that appear in either of the network and edges that do not appear in any of them, so the true positive rate will be 2 /3; In the second calculation, we only count edges that appear in either of the network and in the previous example, the true positive rate will be ½ instead. Since our graph is sparse, we found the second score a more realistic metrics than the first one, which often gives inflated score. For the AUPR, we stick to the first way to include all the edges.

AUROC are computed by a two step process: first we plot the ROC curve with the false positive rate as x-axis and recall score as y-axis at various thresholds; second, the area under the ROC curve along the x-axis is computed to produce the score. AUPR use precision as the x-axis and computed in the area the same way as AUPR. Results show that the baseline method and the LASSO + Prior method are able to detect more edges compared to the random network generated.

6.3. Number of identifiable genes:

We also evaluated our system using the number of genes that are successfully "identified" by the model. We define a target gene to be identifiable if the AUPR score for its predicted TFs is higher than 0.9; similarly, we define a regulator to be identifiable if the AUPR score for its targets is higher than 0.9.

6.4. Cross-validation with expression prediction:

The cross-validation score computes how well the model can predict an unseen gene expression based on the predicted GRN. The predictive power can be scored using either the correlation between the predicted expression and the true expression or the sum-square error.

6.5. Comparison with random networks:

We can generate a few random networks and compare their performance with the proposed models. Define the edge density to be the ratio between the number of edges in the GRN and the maximum number of possible edges for the GRN. The random networks can be generated by connecting a TF and a gene with probability equal to the edge density of the network. Then we can use either AUROC/AUPR or cross-validation for comparison.

7. Experimental Results:

7.1. Module-free LASSO:

The main parameter we need to tune for the baseline LASSO is the regularization weight, alpha. The tables below show how alpha affects the various evaluation metrics. From the Table 7.1.1 and 7.1.2, the model becomes less predictive of the unseen expression data and achieves lower

R2 and PCC scores as alpha becomes larger, possibly because larger alpha makes the model more and more biased; smaller alphas also achieve higher AUPR and AUROC score than larger alphas, possibly because it preserves more useful information for identifying regulatory relations. In Table. 1(b), AUROC over all edges is higher than the AUROC over only positive edges since the former score is inflated with edges between regulators and targets, many of which are counted as correct for free. So we focused on AUROC with positive edges in the later discussion. We also observed a slight improvements in the number of identifiable genes and AUPR when the expression profiles of each gene are normalized to zero-mean and unit variance.

alpha	0.001	0.005	1.0
AUC (completely wrong: 0.50)	0.50046	0.5026	0.5
AUPR (completely wrong: 0.0)	0.0250	0.025	0.027
Avg. R2 (random=0.3)	1	0.73	0.21
Avg. PCC (random=0.6)	0.99	0.9	0.5

Table. 7.1.1 Performance of LASSO with different regularization parameters without knowledge of the regulators on NatVar

log alpha	-5	-4	-2	0
AUC over all edges	0.970	0.97	0.82	0.501
AUC over positive edges (completely wrong: 0.5)	0.552	0.552	0.552	0.501
AUPR (completely wrong: 0.0)	0.0253	0.254	0.0202	0.00146
Avg. R2 (random=0.47)	0.6	0.4	0.4	0.0

Avg. PCC (random=0.71)	0.8	0.8	0.8	0.2

Table. 7.1.2 Performance of LASSO with different regularization parameters given the set of regulators on NatVar + Stress + KO

7.2. Prior LASSO and Module-based LASSO

In Table. 7.2.2, we compare the two extensions of LASSO with the baseline and random network prediction. We can see that for all models, GRN inference with combined data tends to give better results in the number of identifiable genes as well as the AUPR score, except the prior-based approach. Further, both the prior-based and module-based LASSO performs consistently better than the baseline and the random network across all metrics.

For the prior-based LASSO, we tuned the base alpha parameter and found that alpha = 500 / 1881, where 1881 is equal to the number of targets for consistent definition, works best. We did not experiment with the confidence power score gamma but found gamma = 3 generally works well. The model achieves a fold enrichment of 10. By comparing with the performance of using prior network without any expression data, which achieves a fold enrichment of about 15, we find that the success of the prior-based LASSO can be mostly attributed to the quality of the prior network. We leave it as future work to find models that can outperform the prior network by harvesting additional information from gene expression data.

For the module-based LASSO, we found that it performs better than the LASSO without using any prior information, which may be an evidence suggesting that the modular assumption indeed holds for the GRN. We have experimented with the number of modules to be 60, 100 and 300 but found no significant difference in performance.

alpha	10 / 1881	100 / 1881	500 / 1881
AUC over all edges	0.858	0.717	0.662
AUC over positive edges (completely wrong: 0.5)	0.594	0.668	0.652
AUPR (completely wrong: 0.0)	0.0305	0.0659	0.106

Table. 7.2.1 Performance of LASSO+prior with different regularization parameters given the set of regulators on NatVar + Stress + KO

	NatVar only	NatVar + Stress + KO
LASSO	0.0217	0.0253 (0.258)
LASSO + Prior	0.132	0.106
Prior Network	0.22	0.22
Random (p_{edge} = 0.2)	0.0176	0.0175
Module LASSO (K=100)	0.0219	0.0264 (0.224)

Table 7.2.2 AUPR scores of models given the set of regulators, where the number inside the bracket for LASSO is the result after proper data normalization

	NatVar only		NatVar + Stress + KO	
	Number of TFs	Number of Targets	Number of TFs	Number of Targets
LASSO	0	19	0	20 (27)
LASSO + Prior	1	171	1	151
Prior Network	7	404	7	404
Random	0	0	0	0
Group LASSO (K=100)	0	22	0	23 (32)

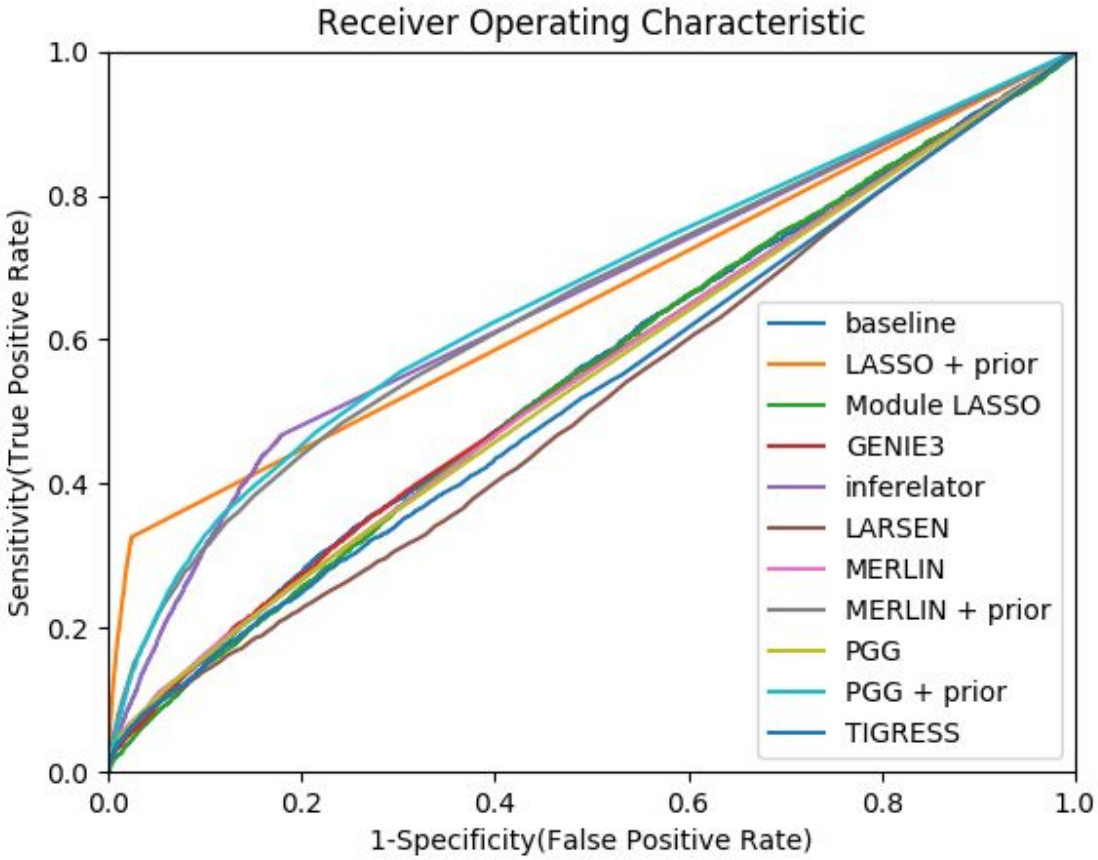
Table 7.2.3 Number of identifiable regulators and targets of the models given the set of regulators, where the numbers inside the brackets are the result after proper data normalization

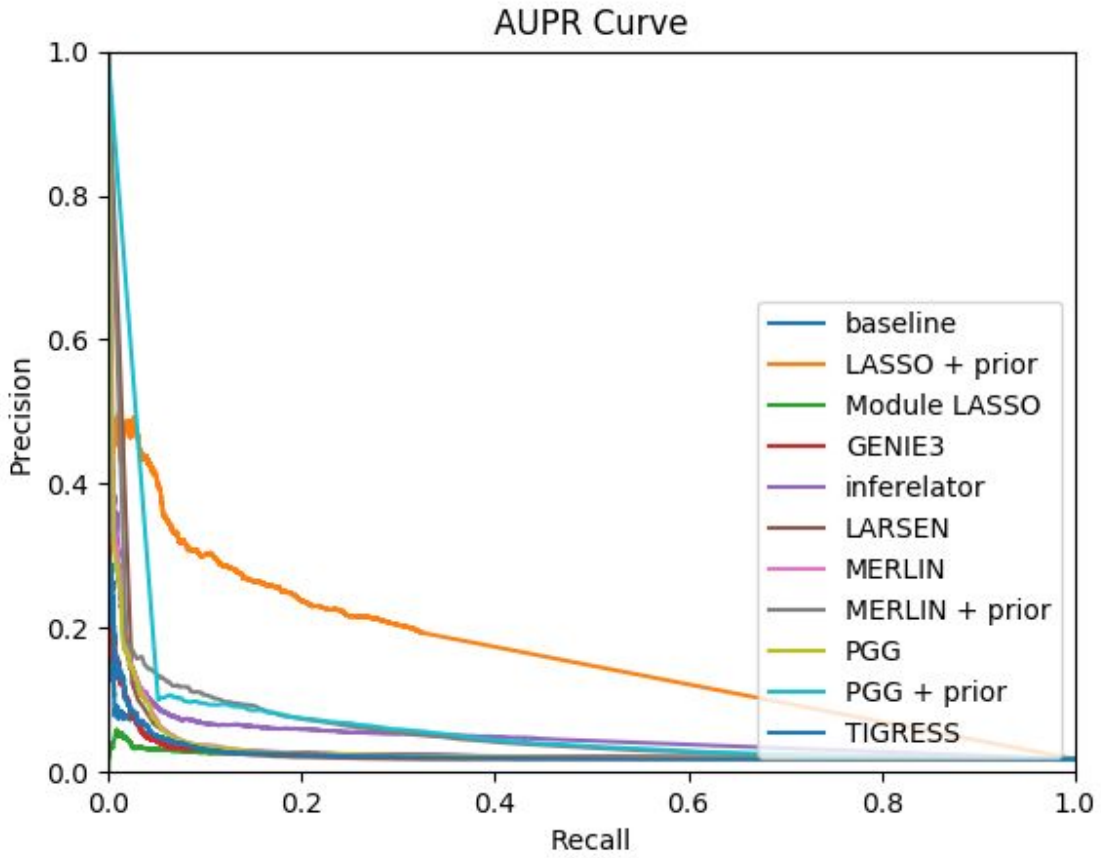
7.3. Results of performance in comparison with other software

#code is developed will run when the LASSO + prior is developed

Model Name	AUPR	AUC
GENIE 3	0.0254	0.548
Inferelator	0.0432	0.651
LARSEN	0.0241	0.511
MERLIN	0.0296	0.546
MERLIN + Prior	0.0480	0.649
PGG	0.0288	0.542
PGG + Prior	0.0452	0.658
TIGRESS	0.0253	0.526
Lasso	0.0253	0.552
Lasso + Prior	0.106	0.652
Module Lasso	0.0264	0.546

Table 7.3.1 Comparison of AUPR and AUROC over positive edges for various models


Receiver Operating Characteristic
Sensitivity(True Positive Rate)
1-Specificity(False Positive Rate)
0.0
0.2
0.4
0.6
0.8
1.0
baseline
LASSO + prior
Module LASSO
GENIE3
inferelator
LARSEN
MERLIN
MERLIN + prior
PGG
PGG + prior
TIGRESS



	Experimental Data + ChIP prior α = .05	Experimental Data + ChIP prior α = .25	Only prior network
AUC	0.668	0.652	0.775
Average precision	0.066	0.106	0.221
Number of identifiable TFs	1	2	7
Number of identifiable targets	151	252	404

7.4 Simulated Dataset

We simulated gene expression and double knock-out data using DREAM4, a Bioconductor package available in R. We evaluated our modular approach using the simulated data and adjusted the α to better understand the limitations of our model.

	Simulated data α = .005	Simulated data α = .0005	Simulated data α = .005
AUC	0.648	0.644	0.560
Average precision	.128	0.123	.081
Number of identifiable TFs	0.0	0.0	0.0
Number of identifiable targets	10.0	12.0	4.0

Table 7.4.1 Number of identifiable regulators and targets of the models given the set of regulators of simulated data generated from DREAM4, a bioconductor package.

7.5 Effect of Group Norm:

We plotted the weight coefficients for module-based and module-free LASSO regression and observed that the module-based regression has denser coefficients than LASSO. While there are some consistent trends in weight vectors from the same module, which are plotted in adjacent rows for visualization purpose, we do not observe a clear modular pattern using the module-based LASSO.

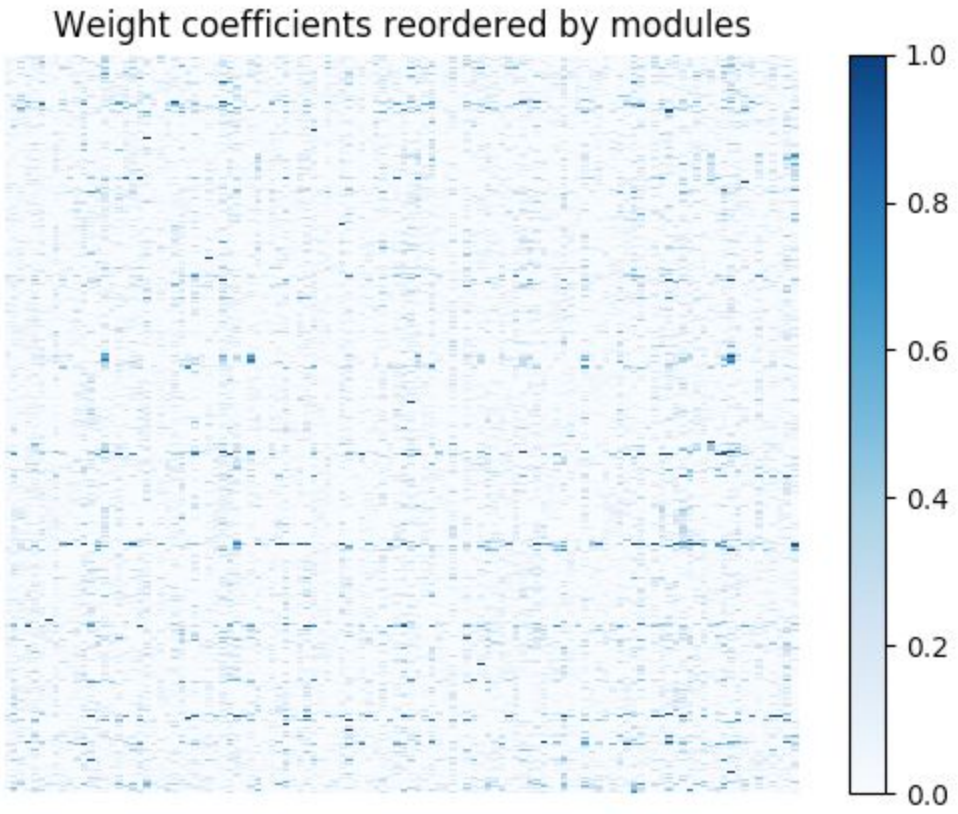


Fig. 7.5.1 Weight coefficients (number of targets by number of TFs) for module-based LASSO regression with K = 60

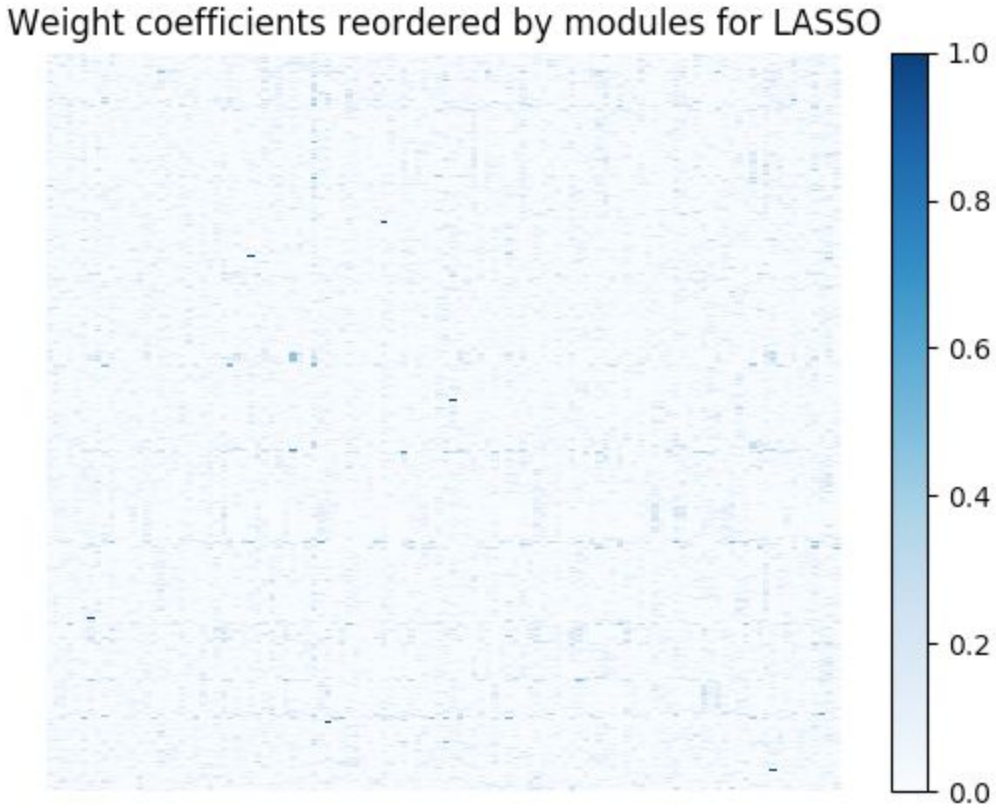


Fig. 7.5.2 Weight coefficients (number of targets by number of TFs) for module-free LASSO regression

8. Discussion

In our model, we displayed how the use of prior information can increase the accuracy and precision of the model. Through the manipulation of the datasets available to us, we were able to identify 1 TFs and 151 target genes.

8.1 Limitation on the data input type

Gene regulatory networks can be redundant and genes are frequently co-regulated, which cause spurious relationships between transcription factors and genes. Specifically, in a regression based method, it is difficult to discriminate true and false interactions. To decrease the amount of false

interactions, it is essential to provide the model with a better framework to prevent bias model selection. In our model we incorporated prior experimental data (i.e. CHIP-seq, knockout gene expression, motif prediction) to decrease the amount of false interactions in the model. Incorporating prior knowledge increased the accuracy of inferred models in our study.

8.2 Limitations of the penalty parameter of LASSO

The LASSO regression method in the model implements a penalty parameter (λ) to identify the relationships between TF and target genes. Lasso regression is one of the most simplistic techniques resulting in the reduction of model complexity and prevents over-fitting which is often the case in simple linear regression.When the λ learned from the model is large this results in a very sparse gene network. When the λ of the model increases, most of the edges tend to accumulate to a few genes, which are also as hub genes eluding to the propensity of the model to only capture hub genes with other datasets could increase [10]. These hub genes are influential in many regulatory networks, and represent the decreased ability of the model to capture unique relationships when the sparsity parameter is set too stringent. Our best model exhibited this phenomena and 2 TFs and 252 target genes were identified. We assume that the model with an increased sparsity parameter identified two hub genes that play a major role in the organism. Therefore, LASSO is more robust than most linear methods; however, LASSO can be biased by the regularization constant λ, as part of the bias-various tradeoff. Another problem with the LASSO model with a high sparsity parameter is that ultimately would be unable to identify unique interaction.

8.3 Limitations of LASSO + prior

The prior information can provide the of GRN structure of the underlying network, but can contain many incorrect or irrelevant interactions. Incorporating structure priors into expression-based GRN inference presents several algorithmic challenges but has the potential to precisely identify gene interactions. In Figure 7.2.3, our results show that reconstruction of the GRN using only prior network identified 7 TFs and 404 regulated genes implementing our modular approach. From this result we identified that the prior network information provides better accuracy and precision than the use of gene expression data. Therefore, the availability of more prior information could increase the accuracy of our model.

8.4 Limitations of the amount of data

The amount of data available is a broad range depending upon the organism being studied. With this knowledge, we evaluated our model with a data set of 100 genes to evaluate its robustness on relatively small dataset. We used simulated data generated from DREAM4, a Bioconductor package available in R to evaluate how the limitation of data may affect our model. Our results, in Table 5, indicate that the use of linear regression may be more applicable to large networks

opposed to small networks. Compared the well annotated yeast network, the simulated network was unable to identify TFs, and < 15 genes were identified.

9. Conclusion

In this work, we presented two methods: 1) labelled as baseline which incorporated LASSO regression and 2) we labelled LASSO + prior incorporated additional prior information on to add precision to the structure of the network. In the analysis of the methods, we focused on parameter choice and robustness minimizing false positives. We conclude that of the two methods the inclusion of prior knowledge significantly improved the quality of inferred networks without damaging our ability to identify new regulatory genes.

10. Reference:

[1] Roy S, Lagree S, Hou Z, Thomson JA, Stewart R, et al. (2013) Integrated Module and Gene-Specific Regulatory Inference Implicates Upstream Signaling Networks. PLoS Comput Biol 9(10): e1003252. doi:10.1371/journal.pcbi.1003252

[2] Siahpirani A and Roy S. (2017) A Prior-based Integrative Framework for Functional Transcriptional Regulatory Network Inference. Nucleic Acids Research, 2017, Vol. 45, No. 4 e21. doi: 10.1093/nar/gkw963.

[3] Huynh-Thu VA, Irrthum A, Wehenkel L and Geurts P. Inferring Regulatory Networks from Expression Data Using Tree-Based Methods. PLoS ONE 5(9): e12776.doi:10.1371/journal.pone.0012776
5(9):e12776.doi:10.1371/journal.pone.0012776

[4] MacNeil, L. T., & Walhout, A. J. M. (2011). Gene regulatory networks and the role of robustness and stochasticity in the control of gene expression. Genome Research, 21(5), 645–657. https://doi.org/10.1101/gr.097378.109

[5] McCall, M. N. (2013). Estimation of Gene Regulatory Networks. *Postdoc Journal : A Journal of Postdoctoral Research and Postdoctoral Affairs*, *1*(1), 60–69.

[6] Ay, A., & Arnosti, D. N. (2011). Mathematical modeling of gene expression: A guide for the perplexed biologist. *Critical Reviews in Biochemistry and Molecular Biology*, *46*(2), 137–151. https://doi.org/10.3109/10409238.2011.556597

[7] Delgado, F. M., & Gómez-Vela, F. (2019). Computational methods for Gene Regulatory Networks reconstruction and analysis: A review. *Artificial Intelligence in Medicine*, *95*, 133–145. https://doi.org/10.1016/j.artmed.2018.10.006

[8] Margolin, A. A., Wang, K., Lim, W. K., Kustagi, M., Nemenman, I., & Califano, A. (2006). Reverse engineering cellular networks. *Nature Protocols*, *1*(2), 662–671. https://doi.org/10.1038/nprot.2006.106
[9] Santra, T. (2014). A Bayesian Framework That Integrates Heterogeneous Data for Inferring Gene Regulatory Networks. *Frontiers in Bioengineering and Biotechnology*, *2*. https://doi.org/10.3389/fbioe.2014.00013.
[10] Hao, T., Ma, H., & Zhao, X. (2012). [Progress in automatic reconstruction and analysis tools of genome-scale metabolic network]. *Sheng Wu Gong Cheng Xue Bao = Chinese Journal of Biotechnology*, *28*(6), 661–670.
[11] Greenfield, A., Hafemeister C and Bonneau R. (2013), Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. Bioinformatics. Apr 15;29(8):1060-7. doi: 10.1093/bioinformatics/btt099.
[12] Yuan, M. & Lin, Y. (2007), 'Model selection and estimation in regression with grouped variables', Journal of the Royal Statistical Society, Series B 68(1), 49–67.
[13] Jerome Friedman, Trevor Hastie and Robert Tibshirani. (2010), A note on the group lasso and a sparse group lasso, arXiv:1001.0736.
[14] Seyoung Kim and Eric P. Xing (2012), 'Tree-guided Group LASSO For multi-response regression with structured sparsity, with an application to eQTL mapping.' The Annals of Applied Statistics. 2012, Vol. 6, No. 3, 1095–1117. DOI: 10.1214/12-AOAS549