Applying Linear Models to Learn Regulation Programs in a Transcription Regulatory Module Network

Jianlong Qi[1], Tom Michoel[2], and Gregory Butler[1]

[1] Department of Computer Science, Concordia University, Montreal, Quebec, Canada
[2] Freiburg Institute for Advanced Studies, School of Life Sciences - LifeNet, Freiburg, Germany

Abstract. The module network method has been widely used to infer transcriptional regulatory network from gene expression data. A common strategy of module network learning algorithms is to apply regression trees to infer the regulation program of a module. In this work we propose to apply linear models to fulfill this task. The novelty of our method is to extract the contrast in which a module's genes are most significantly differentially expressed. Consequently, the process of learning the regulation program for the module becomes one of identifying transcription factors that are also differentially expressed in this contrast. The effectiveness of our algorithm is demonstrated by the experiments in a yeast benchmark dataset.

1 Introduction

There is a complex mechanism in cells that controls which genes are expressed. Generally, this mechanism consists of two levels of controls: post-transcriptional regulation, and transcriptional regulation. The former controls protein synthesis after synthesis of RNA has begun, while the latter controls which genes are transcribed into mRNA. A major part of transcriptional regulation is fulfilled by transcription factors which can influence the expression levels of other genes by binding to their upstreams or downstreams.

Gene expression data have been widely used to infer transcriptional regulatory relationships between genes and their transcription factors. Many methods have been applied for this task such as information-theoretic approaches [5], Bayesian networks [6], and clustering algorithms [4]. In particular, the module network method [14], a special type of Bayesian networks, has shown promising results [15,11]. The models inferred by standard Bayesian networks often overfit data, because the number of parameters to be learned is enormous compared to the number of samples (experimental conditions) in a typical gene expression dataset. In contrast, the module network method, groups genes with similar expression profiles into regulatory modules, and consequently reduces the number of parameters to be learned.

Module network learning consists of two tasks: clustering genes into modules, and inferring a regulation program for each module. Segal $et\ al.$ [15] applied the expectation maximization (EM) algorithm [3] to alternate between these two tasks. That is, genes are grouped into modules in E-steps, and a regulation program for each module

C. Pizzuti, M.D. Ritchie, and M. Giacobini (Eds.): EvoBIO 2011, LNCS 6623, pp. 37–47, 2011.

is learned in M-steps. In [8], the authors enhanced the learning procedure by separating the two tasks: They first group genes into modules using a two-way clustering algorithm [9], and then apply a logistic regression (LR) model to infer the regulation program of each module. Moreover, instead of the LR model, in [13] the authors applied a Gibbs sampler-based learning algorithm to infer regulation programs.

A common strategy of the above module network learning algorithms [15,8,13] is that the regulation program of a module is represented by a regression tree. Each internal node of the tree is associated with a transcription factor and a set of conditions (i.e., a condition cluster), while each leaf node is only associated with a condition cluster. In this way, each internal node represents a contrast between the conditions covered by its left-child and right-child nodes. The confidence of assigning a transcription factor to a particular node is evaluated by the degree of differential expression that that transcription factor manifests in the contrast represented by the node. Accordingly, the overall confidence (i.e., the regulatory score) for assigning a transcription factor to a module is calculated by summing individual confidences for that transcription factor in all internal nodes of the module's tree.

In this work we apply linear models to learn the regulation program of a module. Given a condition clustering of the module, instead of building a regression tree, the proposed method extracts the contrast in which the module's genes are most significantly differentially expressed, called the *critical contrast*. The differential expression under the contrast represents an important characteristic of the expression profile of the genes, so the process of learning the regulation program for the module becomes one of identifying transcription factors whose expression profiles are also associated with the characteristic. The effectiveness of the proposed method is demonstrated by applying it to a real biological dataset.

The remainder of this paper is organized as follows: Section 2 describes how to apply linear models to infer regulatory relationships in module networks; Section 3 presents experimental results; Section 4 summarizes the main results and points to future work.

2 Inferring Regulatory Relationships in Module Networks by Linear Models

Given a condition clustering of a gene module, the proposed method consists of two tasks: extracting the critical contrast of the condition clustering, and inferring transcription factors based on the contrast. We use linear models to accomplish both tasks. The following subsections describe the details of each task.

2.1 Extracting the Critical Contrast of a Condition Clustering

The purpose of identifying the critical contrast of a condition clustering is to find, between which two condition clusters, the module's genes are most significantly differentially expressed. Consequently, we define the critical contrast as consisting of two condition clusters: the *extraordinary cluster*, in which the genes show extraordinary behaviors (i.e., extremely high or low expression values); and the *ordinary cluster*, in which the genes show ordinary behaviors.

We measure the differential expression of genes between condition clusters with the linear model described below. Suppose that in a dataset we identified a gene module M in which conditions are partitioned into two clusters: c_1 and c_2. The expression values of genes in M under the condition i can then be represented by the linear model [10]:

$$y_i = \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i \tag{1}$$

where β_1 and β_2 denote regression coefficients, and ε_i is normally distributed with mean 0 and variance σ^2. X_{i1} and X_{i2} are indicator variables defined as follows:

$$X_{i1} = \begin{cases} 1 : i \in c_1 \\ 0 : i \notin c_1 \end{cases},$$

$$X_{i2} = \begin{cases} 1 : i \in c_2 \\ 0 : i \notin c_2 \end{cases}.$$

In this way, the degree of differential expression of the genes in M between c_1 and c_2 can be determined by the ordinary t statistic, which is defined as:

$$t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}\sqrt{\frac{n_1+n_2}{n_1 n_2}}} \tag{2}$$

where μ_1 and μ_2 are the means of the expression values of the genes in c_1 and c_2, respectively; s_1 and s_2 are the standard deviations in c_1 and c_2, respectively; n_1 and n_2 denote the numbers of conditions in c_1 and c_2, respectively.

We then apply the following searching strategy to identify the critical contrast of a given condition clustering c of M. Suppose that c consists of N condition clusters. First we sort these N condition clusters into an ordered list according to the means of the expression values in the clusters. Then, we calculate the ordinary t-statistic for the contrast between the unions of the first k condition clusters ($k = 1, 2, ..., N-1$) and remaining $N-k$ condition clusters in the ordered list. Finally, the contrast with the maximum t-statistic among the $N-1$ contrasts is chosen as the critical contrast of c. Consequently, its associated union of condition clusters with the higher absolute mean of expression values becomes the extraordinary cluster c_e, while the other union becomes the ordinary cluster c_o.

2.2 Using Moderated t-Statistics to Select Differentially Expressed Transcription Factors

Since the genes in M show different behaviors between c_e and c_o, the task of learning the regulation program of M can be accomplished by identifying transcription factors that are also dramatically differentially expressed between the same two clusters. We may apply ordinary t statistics as defined in Eq. 2 to do the work, but inferences based on the statistics might not be stable when the number of expression values in c_e or c_o is small. This is a likely situation, because we only evaluate the expression values of an individual transcription factor instead of a set of genes.

To cope with the instability, we use a moderated t-statistic [16,17], based on a Bayesian hierarchical model, to select differentially expressed transcription factors.

Given a transcription factor r, the hierarchical model assumes a prior distribution for the variance of r (σ_r^2), which is defined as:

$$\frac{1}{\sigma_r^2} \sim \frac{1}{d_0 s_0^2} \chi^2_{d_0} \tag{3}$$

where d_0 and s_0 are estimated by an empirical Bayes approach, and $\chi^2_{d_0}$ represents a chi-square distribution with d_0 degree of freedom. It can be shown that the posterior mean of σ_r^{-2} is:

$$\tilde{s_r}^2 = \frac{d_0 s_0^2 + (n_e - 1)s_{re}^2 + (n_o - 1)s_{ro}^2}{d_0 + n_e + n_o - 2} \tag{4}$$

where s_{re} and s_{ro} are the standard deviations of the expression values of r in c_e and c_o. The moderated t-statistic is defined by replacing the pooled variance in Eq. 2 by $\tilde{s_r}$:

$$\tilde{t_r} = \frac{\mu_{re} - \mu_{ro}}{\tilde{s_r}\sqrt{\frac{n_e + n_o}{n_e n_o}}} \tag{5}$$

where μ_{re} and μ_{ro} are the means of the expression values of r in c_e and c_o, respectively; and n_e and n_o denote the numbers of conditions in c_e and c_o, respectively. The $\tilde{t_r}$ provides more stable inference when the number of conditions is small [16], because it borrows extra information from the ensemble of genes in the dataset by using d_0 and s_0. Furthermore, in order to make $\tilde{t_r}$ comparable with moderated t-statistics based on other condition clusterings, we can normalize $\tilde{t_r}$ by:

$$\tilde{t}_{r\_standardized} = \frac{\tilde{t_r} - \mu_{\tilde{t}}}{s_{\tilde{t}}} \tag{6}$$

where $\mu_{\tilde{t}}$ and $s_{\tilde{t}}$ are the mean and standard deviation of the moderated t-statistics of all candidate transcription factors based on c.

2.3 The Regulatory Score for Assigning a Transcription Factor to a Module

If the expression values of genes in M can be partitioned into multiple equiprobable condition clusterings, then the overall confidence (i.e., the regulatory score) of the assignment of r may be calculated by summing the individual confidences that r shows in all condition clusterings. Hence, the regulatory score for assigning r to M over a set of condition clusterings C is defined as:

$$Z(r) = \sum_{c \in C} \tilde{t}_{cr\_standardized} \tag{7}$$

where $\tilde{t}_{cr\_standardized}$ is the standardized moderated t-statistic of r, based on a condition clustering c. We can rank all candidate transcription factors according to their regulatory scores as defined in Equation 7. The higher its ranking, the more likely a candidate transcription factor regulates M.

3 Experimental Results and Discussion

3.1 Dataset and Validation Reference Database

We applied the proposed method to a yeast dataset which measures yeast's response to various stresses, and consists of 173 experimental conditions [7]. In [9] 2355 differentially expressed genes in this dataset were clustered into 69 gene modules. We sampled 10 condition clusterings for each gene module using a Gibbs sampler [8]. Then, given the list of 321 candidate transcription factors prepared in [15], we calculated the regulatory score for assigning a transcription factor to a particular module as defined in Eq. 7. Furthermore, the regulatory relationships between 185 transcription factors and 6297 genes recorded in YEASTRACT [12] (released on Apr 27, 2009) were used as the reference database to evaluate predictions given by the linear model.

3.2 Results for Regulation of Nitrogen Utilization

In the yeast stress dataset, a module for nitrogen utilization was obtained in [8]. This module consists of 47 genes mostly involved in two pathways: the methionine pathway (regulated by MET28 and MET32), and the nitrogen catabolit regression (NCR) system (regulated by GLN3, GZF3, DAL80 and GAT1). Both pathways relate to the process by which yeast use the best available nitrogen source in the environment [1,2].

In this module, we sampled a condition clustering with 18 clusters that were ordered descendingly by their means of expression values. As shown in Fig. 1, we obtained the maximum ordinary t-statistic (38.98) when we compared the union of the first 3 condition clusters with the remaining clusters in the ordered list. This indicates that the extraordinary cluster of the clustering's critical contrast includes those conditions under nitrogen depletion and amino-acid starvation where using non-preferred nitrogen sources is crucial, while the ordinary cluster consists of the remaining conditions. Accordingly, the critical contrast represents the comparison of the genes' behaviors under preferred and non-preferred nitrogen sources.

Figure 2 shows that the module's genes are dramatically differentially expressed in the contrast. That is, they are only highly expressed under non-preferred nitrogen sources (i.e., the extraordinary cluster). Similar results were obtained for the critical contrasts of the other nine condition clusterings of the module.

We then ordered candidate transcription factors according to their regulatory scores as defined in Eq. 7. Table 1 shows the top ten regulators as ranked by the linear model, which includes most known transcription factors of the NCR process and the methionine pathway.

3.3 Linear Model versus LeMoNe in the NCR Process

In this subsection we compare the predictions for the module studied in the Section 3.2 given by the linear model and by the LeMoNe regression tree-based method [8]. As shown in Table 1, both methods identified most known transcription factors of the module, but they ranked the transcription factors of the NCR process (denoted with *) differently. DAL80, GLN3, GZF3, and GAT1 are the first, fifth, eighth, and ninth regulators in the rank by the linear model. However, LeMoNe ranks GAT1, DAL80,

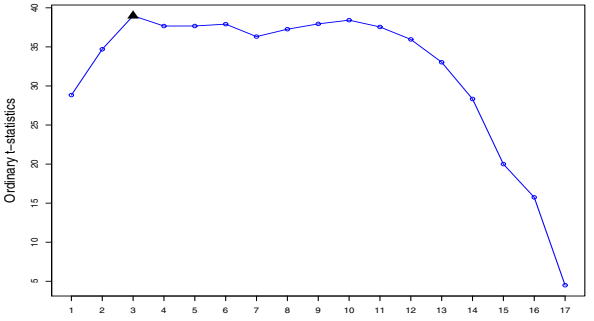


Fig. 1. Ordinary t-statistic for the contrast between the union of the first k condition clusters ($i = 1, 2, ..., 17$) and the remaining $18 - k$ clusters. The horizontal axis gives the values of k. The colored triangle represents the largest ordinary t-statistic.

Table 1. Transcription factors for the regulation of nitrogen utilization as inferred by the linear model and LeMoNe

Top 10 regulators as ranked by the linear model and the numbers of genes in the module they regulate											
rank	1	2	3	4	5	6	7	8	9	10
regulator	**DAL80***	MET32	UGA3	LYS14	**GLN3***	YAP5	MET28	**GZF3***	**GAT1***	DAL82
#Gene regulated	10	13	1	1	18	3	8	6	7	9
Top 10 regulators as ranked by LeMoNe and the numbers of genes in the module they regulate										
rank	1	2	3	4	5	6	7	8	9	10
regulator	**GAT1***	MET28	MET32	**DAL80***	UGA3	THI2	YAP5	CMP2	GCN20	INO2
#Gene regulated	7	8	13	10	1	0	3	0	0	1

* regulators are known transcription factors of NCR process.

GZF3 as the first, fourth and fourteenth regulators, and most strikingly, GLN3 is out of the top 100. We next investigate why their assigned confidences vary so widely between the two methods.

Given the condition clustering we studied in the previous subsection, LeMoNe built a regression tree as shown in Fig. 3, in which we focus on three condition clusters: cluster1; cluster2, mainly consisting of conditions in stationary phase; and cluster3, including the conditions under nitrogen depletion and amino acid starvation (i.e., the extraordinary cluster identified by the liner model).

As seen in Fig. 2, genes in the module are not expressed in cluster1, and slightly expressed in cluster2, but significantly expressed in cluster3. We speculate that the conditions in cluster2 represent a transition from utilizing preferred nitrogen sources to non-preferred nitrogen sources. During the transition, preferred nitrogen sources become less and less available, such that NCR related genes are expressed to some degree, but the expressed amount is much less than that under non-preferred nitrogen sources (e.g., conditions in cluster3).

The confidence of assigning a transcription factor to the module by LeMoNe is mainly determined by the degree of the transcription factor's differential expression in the contrast between cluster1, and the union of cluster2 and cluster3 (i.e., the contrast represented by the root node of the regression tree). For example, GAT1 is significantly more differentially expressed in the contrast (p-value $<$ 2.2e-16 for two sample t-test)

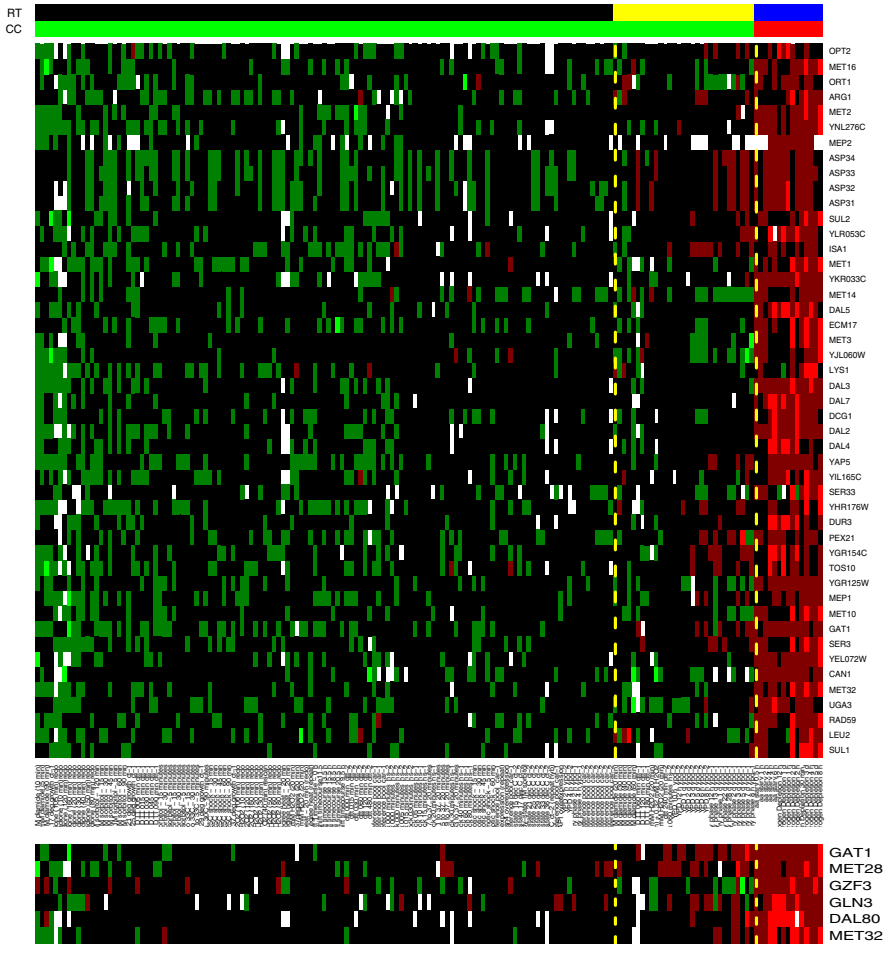


Fig. 2. Heatmaps of expression values of genes in the module (top), and known transcription factors of the module (bottom). In track CC (critical contrast) conditions assigned to the extraordinary and ordinary clusters are colored by red and green, respectively. In track RT (regression tree) conditions assigned to cluster1, cluster2, and cluster3 (detailed in Fig. 3) are colored by black, yellow, and blue, respectively.

than GLN3 (p-value = 9.043e-06 for two sample t-test), and consequently LeMoNe ranks GAT1 much higher than GLN3.

On the other hand, as explained in the previous subsection, the linear model searches for transcription factors that are differentially expressed in the contrast between cluster3, and the union of cluster1 and cluster2 (i.e., between conditions under non-preferred nitrogen sources and the other conditions). Since genes in the module are involved in the process by which the yeast uses the best available nitrogen source, differential expression in this contrast is the most important property of the genes' expression profile.

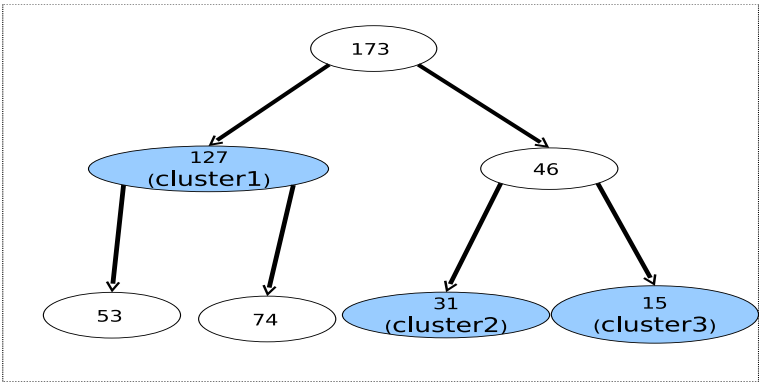


Fig. 3. Top 3 levels of the LeMoNe's regression tree built on a condition clustering of the module for nitrogen utilization. A condition cluster is represented by a circle containing its number of conditions. Three of the clusters have been given labels for easy reference in the text.

But as the contrast can not be directly represented by LeMoNe's regression tree, it gives low confidence for the assignment of two known regulators (GLN3 and GZF3) to the module.

The above results might indicate a limitation of regression tree-based algorithms. That is, tree structures can represent a contrast between two condition clusters only if they are assigned to the left-child and right-child of a same internal node. Hence, regression tree-based learning may miss some biologically meaningful contrasts.

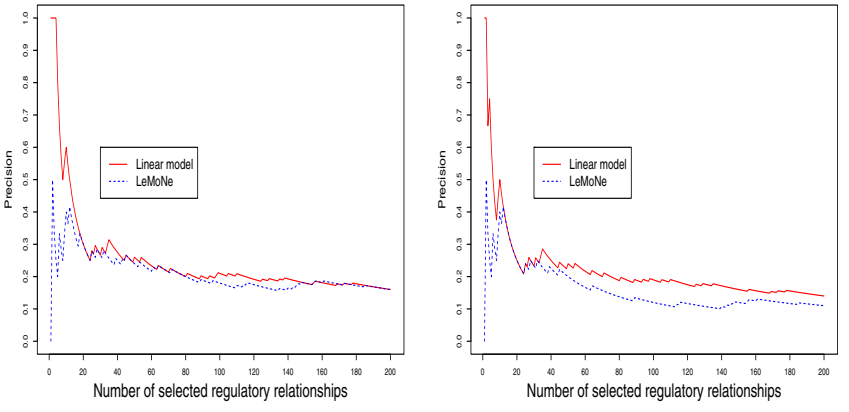
3.4 Results over the Entire Yeast Stress Dataset

In this subsection we compare the performance of the linear model and LeMoNe over the entire yeast stress dataset. We apply each method to the dataset to calculate the regulatory score for assigning a regulator to a module. Then we order all of the method's regulatory scores between 321 candidate transcription factors and 69 modules in descending order. This leads to a ranked list of 22149 regulator-module interactions for the method.

For each regulator-module interaction, we use the the hypergeometric distribution—based on the number of genes regulated by the regulator in the dataset, the number of genes regulated by the regulator in the module, and the number of genes in the module—to calculate the p-value of the regulator *module-wise* prediction. In Figs. 4(a) and 4(b), we show the precisions of the top i predictions ($i = 1, 2, ..., 200$) in rankings of LeMoNe and the linear model, at threshold p-values of 0.01 and 0.005. For a given threshold, the precision of the top i regulator module-wise predictions in a ranking is defined as:

$$precision_{module-wise}(i) = \frac{TP(i)}{i},$$

where $TP(i)$ represents the number of predictions with p-values less than the threshold in the top i predictions. Generally, the linear model obtains slightly better precisions than LeMoNe at both thresholds.



(a) p-value=0.01 (b) p-value=0.005

Fig. 4. Precisions of the linear model and LeMoNe in the yeast stress dataset at the threshold p-values of 0.01 and 0.005

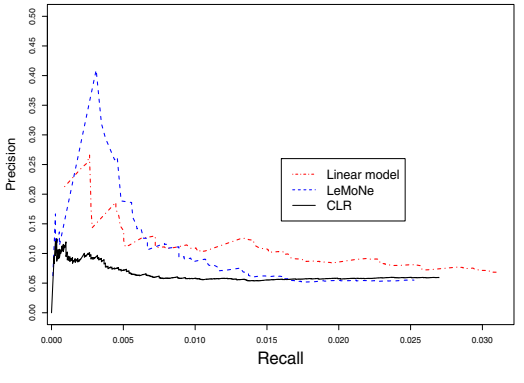


Fig. 5. Precision versus recall curves for the linear model, LeMoNe and CLR in the yeast stress dataset

Looking deeper, we compare the regulator *gene-wise* performance of the proposed method, LeMoNe, and the CLR (Context Likelihood of Relatedness) algorithm [5] which directly infers regulatory relationships between transcription factors and genes. In order to convert regulator module-wise predictions, as given by the first two methods, into regulator gene-wise predictions, we make the simplifying assumption that the regulator of each module-wise prediction regulates all genes in the module. Following this strategy, the top 200 regulator module-wise predictions from the linear model yield 4993 regulator gene-wise predictions. The closest number of gene-wise predictions yielded by LeMoNe (5021) are produced by its top 191 module-wise predictions. Taking these gene-wise predictions from LeMoNe and the linear model with the top 4993 predictions from CLR, we get Fig. 5 showing the precision versus recall curves

for these three methods. The precision and recall of the top i regulator gene-wise predictions from a method is defined as:

$$precision_{gene-wise}(i) = \frac{TP(i)}{i} \quad , \quad recall_{gene-wise}(i) = \frac{TP(i)}{P},$$

where $TP(i)$ represents the number of regulator-gene interactions recorded in YEASTRACT in the top i predictions, and P gives the total number of interactions recorded in YEASTRACT. LeMoNe and the linear model obtain similar results (with areas under the curves of 0.0028 versus 0.0033), and they both outperform CLR. This demonstrates the effectiveness of module-based learning algorithms.

In general, LeMoNe and the proposed method achieve comparable performance in the dataset, but we observed that they retrieve very different parts of the transcriptional regulatory networks in the yeast. For example, in Table 2 which shows the top 10 predictions given by two methods, the only overlapped true positive is the assignment of DAL80 to module 51. In LeMoNe it is the second prediction, while in the linear model it is the fourth prediction. The difference is because LeMoNe and the linear model depend on distinct contrasts to infer regulators of modules (i.e., select differentially expressed transcription factors). The difference also suggests that combining the predictions given by these two methods might be a promising direction.

Table 2. Inferred regulatory relationships by the linear model and LeMoNe in the yeast stress dataset

Top 10 predictions given by the linear model										
rank	1	2	3	4	5	6	7	8	9	10
regulator	**DAL80***	**MET32***	**PHD1***	**DAL80***	DAL82	UGA3	ACA1	DAL80	**LYS14***	**GLN3***
module	11	11	36	51	48	11	48	40	11	11
Top 10 predictions given by LeMoNe										
rank	1	2	3	4	5	6	7	8	9	10
regulator	PDR3	**DAL80***	USV1	HAP4	IME4	**HAP4***	XBP1	TOS8	**GAT1***	GAL80
module	13	51	28	30	46	7	10	24	11	41

* records represent true positives at the threshold p-value =0.01

4 Conclusion and Future Works

In this paper, we proposed to apply a linear model, rather than regression trees, to infer regulators in transcriptional module networks. Experiments in a yeast dataset show that the simple linear model can achieve comparable results with LeMoNe, a well known regression tree-based algorithm. In the future work, we will focus on integrating results from multiple complementary regulation program learning algorithms.

References

1. Nitrogen regulation in saccharomyces cerevisiae. Gene 290(1-2), 1–18 (2002)
2. Cunningham, T.S., Rai, R., Cooper, T.G.: The Level of DAL80 Expression Down-Regulates GATA Factor-Mediated Transcription in Saccharomyces cerevisiae. J. Bacteriol. 182(23), 6584–6591 (2000)
3. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)

4. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences of the United States of America 95(25), 14863–14868 (1998)
5. Faith, J.J., Hayete, B., Thaden, J.T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J.J., Gardner, T.S.: Large-Scale Mapping and Validation of Escherichia coli Transcriptional Regulation from a Compendium of Expression Profiles. PLoS Biology 5(1), 54–66 (2007)
6. Friedman, N.: Inferring Cellular Networks Using Probabilistic Graphical Models. Science 303(5659), 799–805 (2004)
7. Gasch, A.P., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D., Brown, P.O.: Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes. Mol. Biol. Cell 11(12), 4241–4257 (2000)
8. Joshi, A., De Smet, R., Marchal, K., Van de Peer, Y., Michoel, T.: Module networks revisited: computational assessment and prioritization of model predictions. Bioinformatics 25(4), 490–496 (2009)
9. Joshi, A., Van de Peer, Y., Michoel, T.: Analysis of a Gibbs sampler method for model-based clustering of gene expression data. Bioinformatics 24(2), 176–183 (2008)
10. Kutner, M.H., Neter, J., Nachtsheim, C.J., Li, W.: Applied Linear Statistical Models. McGraw-Hill Irwin, New York (2005)
11. Li, J., Liu, Z.J., Pan, Y.C., Liu, Q., Fu, X., Cooper, N.G., Li, Y., Qiu, M., Shi, T.: Regulatory module network of basic/helix-loop-helix transcription factors in mouse brain. Genome Biol. 8(11), R244 (2007)
12. Monteiro, P.T., Mendes, N.D., Teixeira, M.C., d'Orey, S., Tenreiro, S., Mira, N.P., Pais, H., Francisco, A.P., Carvalho, A.M., Lourenco, A.B., Sa-Correia, I., Oliveira, A.L., Freitas, A.T.: YEASTRACT-DISCOVERER: new tools to improve the analysis of transcriptional regulatory associations in Saccharomyces cerevisiae. Nucl. Acids Res. 36(suppl. 1), D132–D136 (2008)
13. Qi, J., Michoel, T., Butler, G.: A regression tree-based gibbs sampler to learn the regulation programs in a transcription regulatory module network. In: Proceedings of 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 1–8 (2010)
14. Segal, E., Pe'er, D., Regev, A., Koller, D., Friedman, N.: Learning module networks. Journal of Machine Learning Research 6, 557–588 (2005)
15. Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., Friedman, N.: Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nature Genetics 34(2), 166–176 (2003)
16. Smyth, G.K.: Linear models and empirical bayes methods for assessing differential expression in microarray experiments. Stat. Appl. Genet. Mol. Biol. 3 (2004)
17. Smyth, G.K.: Bioinformatics and Computational Biology Solutions using R and Bioconductor, pp. 397–420. Springer, New York (2005)