University of California, Berkeley

U.C. Berkeley Division of Biostatistics Working Paper Series

Year 2004 *Paper* 145

Regulatory Motif Finding by Logic Regression

Sunduz Keles* Mark J. van der Laan†

Chris Vulpe‡

*Division of Biostatistics, School of Public Health, University of California, Berkeley , keles@stat.wisc.edu

†Division of Biostatistics, School of Public Health, University of California, Berkeley, laan@berkeley.edu

‡Nutritional Science & Toxicology, University of California, Berkeley

Regulatory Motif Finding by Logic Regression

Sunduz Keles, Mark J. van der Laan, and Chris Vulpe

Abstract

Multiple transcription factors coordinately control transcriptional regulation of genes in eukaryotes. Although multiple computational methods consider the identification of individual transcription factor binding sites (TFBSs), very few focus on the interactions between these sites. We consider finding transcription factor binding sites and their context specific interactions using microarray gene expression data. We devise a hybrid approach called LogicMotif composed of a TFBS identification method combined with the new regression methodology logic regression of Ruczinski et al. (2003). LogicMotif has two steps: First potential binding sites are identified from transcription control regions of genes of interest. Various available methods can be used in this first step when the genes of interest can be divided into groups such as up and down regulated. For this step, we also develop a simple univariate regression and extension method MFURE to extract candidate TFBSs from a large number of genes in the availability of microarray gene expression data. MFURE provides an alternative method for this step when partitioning of the genes into disjoint groups is not preferred. This first step aims to identify individual sites within gene groups of interest or sites that are correlated with the gene expression outcome. In the second step, logic regression is used to build a predictive model of outcome of interest (either gene expression or up and down regulation) using these potential sites. This two-fold approach creates a rich diverse set of potential binding sites in the first step and builds regression or classification models in the second step using logic regression that is particularly good at identifying complex interactions.

LogicMotif is applied to two publicly available data sets. A genome-wide gene expression data set of Saccharomyces cerevisiae is used for validation. The regression models obtained are interpretable and the biological implications are in agreement with the known resuts. This analysis suggests that LogicMotif provides biologically more reasonable regression models than previous analysis of this data

set with standard linear regression methods. Another data set of Saccharomyces cerevisiae illustrates the use of LogicMotif in classification questions by building a model that discriminates between up and down regulated genes in iron copper deficiency. LogicMotif identified an inductive and two repressor motifs in this data set. The inductive motif matches the binding site of the transcription factor Aft1p that has a key role in regulation of the uptake process. One of the novel repressor sites is highly present in transcription control regions of FeS genes. This site could represent a TFBS for an unknown transcription factor involved in repression of genes encoding FeS proteins in iron deficiency. We established the stability of the method to the type of outcome variable by using both continuous and binary outcome variables for this data set. Our results indicate that logic regression used in combination with cluster/group operating binding site identification methods or with our proposed method MFURE is a powerful and flexible alternative to linear regression based motif finding methods.

1 Introduction

The transcriptional regulatory apparatus is organized in the form of arrays of *transcription factor binding sites (TFBSs)* or *motifs* on DNA. Identifying the components of this array, and the relationships among them is one of the challenging problems of contemporary biology. Transcriptional regulation in eukaryotic organisms requires cooperation of multiple transcription factors. To date, most computational methods focus on identifying single or multiple TFBSs rather than exploring their interdependence in regulation. Such TFBS finding methods can roughly be divided into three: (1) Cluster/group operating methods, (2) Regression based methods using gene expression (3) Dictionary methods. The cluster/group operating methods (Lawrence and Reilly; 1990; Lawrence et al.; 1993; Bailey and Elkan; 1995; Neuwald et al.; 1995; Hertz and Stormo; 1999; Tavazoie et al.; 1999; van Helden et al.; 1998; Tompa; 1999; Sinha and Tompa; 2000; Keleş et al.; 2003) identify potential TFBSs from a set of co-expressed genes by assuming that co-expression implies co-regulation. Regression based methods model gene expression as a function of short oligonucleotides and select the most relevant ones (Bussemaker et al.; 2001; Keleş et al.; 2002) by model selection or hypothesis testing methods. The dictionary based methods (Li et al.; 2000) do not utilize microarray data but rather build a dictionary of oligonucleotide words from the whole genome of a given organism and predict the potential TFBSs among these words. Though useful for many problems, these methods suffer from several drawbacks. Cluster/group operating motif finding methods do not necessarily come up with the most characteristics set of motifs for a given group of genes. Consequently, the source of differences in the regulatory mechanisms between differentially regulated (e.g. up vs down) groups becomes much more difficult to understand. For example, there might be many common motifs identified for the two groups. In addition, the cluster/group operating motif finding algorithms result in a large set of potential motifs and some sort of significance cut-off is required to decide on where to stop on the list. Typically, a score (goodness measure) is attached to each motif and significance levels are assigned based on these scores. Most importantly, these methods do not explore combinatorial relationships among motifs. In contrast, regression based methods try to capture some of the interactions among the motifs but they suffer from the limitations of the motif models (typically short oligonucleotides) considered.

In this paper, we address some of the limitations of existing approaches. We combine cluster/group operating motif finding methods with the regression approach in a hybrid approach that we refer to as `LogicMotif`. `LogicMotif` is especially powerful when the goal is to identify the most discriminating potential sites between groups of genes. It takes advantage of available cluster/group operating motif finding methods to generate candidate motifs (we also refer to motifs as covariates from the regression point of view) and utilize logic regression to build a regression model or a

classifier for the genes of interest. The use of multiple existing motif finding methods provides a rapid way of generating a diverse class of motifs for each group of genes. Subsequently, logic regression identifies the most discriminating or predictive motifs between the two groups and elucidates combinatorial relationships among these motifs. In our approach, a natural way to choose a cut-off on the potential TFBS list is by using cross-validation. For each possible cut-off value, logic regression step can be performed with the binding site list identified by the corresponding cut-off and cross-validated prediction error can be used to determine the amount of reduction in the list. This fine tuning provides a balance between the variance and bias trade-off of the regression models constructed. As an alternative to using cluster/group operating motif finding methods in the first step of `LogicMotif`, we develop a univariate regression and extension method (`MFURE`) for identifying potential sites correlated with microarray gene expression data.

Recently, Conlon et al. (2003) developed a method called `MOTIF REGRESSOR`. This novel approach combines binding site identification using position weight matrices, in particular using `MDSCAN` of Liu et al. (2002), and the linear regression approach to motif finding (Bussemaker et al.; 2001; Keleş et al.; 2002). This two stage approach is similar to our approach in philosophy since it also first identifies potential TFBSs from groups of genes separately and then uses linear regression to model gene expression as a function of these sites. Given the complexity of transcriptional factor binding sites, we suggest that a more flexible approach in both of the steps may be necessary to identify important motifs. Hence we allow motif finding by any method in our approach as long as a binary score can be extracted from the identified candidate binding site. Similarly, logic regression or tree based regression provides a flexible alternative to linear regression. We also propose a simple method based on univariate regression and extension for identifying motifs from a larger group of genes ($\sim$ 200 genes) where cluster/group operating methods might not adequately perform due to high noise levels (significant technical or experimental variability). Time course experiments (i.e., cell cycle regulated genes) in which many genes show differential expression at more then one time point are examples of such settings.

Although most approaches reviewed above do not consider the combinatorial nature of transcriptional regulation, Pilpel et al. (2001) explicitly address identifying motif combinations by calculating an expression coherence measure for the genes that contain all the motifs of interest. This approach is capable of identifying combined effects of a given set of motifs but it lacks the ability to identify and quantify additive effects. In logic regression terminology, this approach only uses "and" operator between motifs but not the "or" operator. As reviewed later in this paper, the logic regression approach is not limited to one type of operator and can generate a series of models from very simple to complex.

In another work, GuhaThakurta and Stormo (2001) address the problem of discovering sites for cooperative binding of two transcription factors by using a likelihood-

based approach that involves modeling of sequence data using two position weight matrices. This approach, which is limited to two interacting binding sites, is different than the approaches above since it does not use microarray data, and it is not suited for identifying context specific coordination of factors.

We applied `LogicMotif` to two data sets of *Saccharomyces cerevisiae*. Since there is a considerable prior information on the regulatory mechanisms of *Saccharomyces cerevisiae*, we were able to confirm the biologic validity of our findings. Our analysis with `Logicmotif` created simple hypotheses for combinatorial interaction of the binding sites and in several cases the resulting models were simple linear regression models of the motifs themselves which agreed with the results of previous regression based methods.

2 Methods

Let Y denote the outcome of interest. Y could be continuous, e.g., representing the log ratio of mRNA abundance in two different samples (referred to here as *(relative) gene expression*), or it could be a binary variable representing the class of genes, e.g., 0 for down-regulated genes and 1 for up-regulated genes. We assume to have N independent and identically distributed observations of random variable Y. For any given potential binding site set of size M, we define a binary covariate vector

$$\overrightarrow{S}_n = (S_{n,1}, \cdots, S_{n,M}),$$

for each gene n. The entries of this vector are defined as

$$S_{n,m} = \begin{cases} 1 & \text{if} \quad \text{gene } n \text{ has at least one copy of motif } m, \\ 0 & \text{o.w.} \end{cases}$$

Given the outcome variable Y and the covariate vector $\overrightarrow{S}$, we are interested in building a predictive model of Y based on $\overrightarrow{S}$. In particular, we are going to look at the regression and classification setting.

Regression problem. We would like to regress the outcome Y on the covariate vector $\overrightarrow{S}$

$$E[Y \mid \overrightarrow{S}] = f(\overrightarrow{S} \mid \beta),$$

where $f(.)$ is a function of the covariate vector $\overrightarrow{S}$ parametrized by β. A simple example of such a regression model is a linear regression model given by

$$E[Y \mid \overrightarrow{S}] = \beta_0 + \beta_1 S_1 + \cdots + \beta_m S_m. \tag{1}$$

If Y is a binary variable, a logistic regression model

$$E\left[\log\left(\frac{P(Y=1 \mid \overrightarrow{S})}{1-P(Y=1 \mid \overrightarrow{S})}\right)\right] = \beta_0 + \beta_1 S_1 + \cdots + \beta_m S_m, \tag{2}$$

might be more appropriate. In both of these models, the β coefficients need to be estimated and the motifs with nonzero coefficients have to be identified. Such motifs represent the "most relevant" motifs, that is, they contribute to the prediction of the outcome variable. The selection of such motifs typically involves applying model selection techniques such as cross-validation. Note that neither of these models are taking into account any combinatorial effects of the motifs. In the next subsection, we consider the extension of these models to incorporate such effects.

Classification problem. When Y is a binary variable, a classical approach is build a classifier rule based on the covariate set S that will classify N observations from the random variable Y into two groups. The goal is, given a set of motif scores for a particular gene, to be able to say whether that gene will be up or down regulated under a given experimental condition.

2.1 `LogicMotif` overview

`LogicMotif` is a systematic combination of the methods we review and propose in the following subsections. In summary, it consists of two steps:

1. *Motif finding:* This step involves the identification of potential motifs from the gene groups of interest. Depending on the nature of the problem at hand, various methods can be used. If the problem involves groups of differentially expressed genes (up and down regulated), off-the-shelf group/cluster operating TFBS finding methods can be used. Let the set of motifs identified from the down group be $\mathcal{M}_d$ and the set of motifs identified from the up group be $\mathcal{M}_u$. The final motif $\mathcal{M}$ set that will be used in the second step of `LogicMotif` is the union of $\mathcal{M}_d$ and $\mathcal{M}_u$. If the genes of interest constitute a large group (genes from time course experiments or a groups of related experiments) and microarray gene expression data is available, `MFURE` method that we propose in subsection 2.2.2 can be used. This method constructs longer oligonucleotides from pentamers. As a result, all pentamers and/or their extensions at all time points or related experiments might be pooled together to form $\mathcal{M}$.
   (a) *Covariate extraction:* For each gene, a binary score vector $\overrightarrow{S}$ representing the occurrence of the motifs in that gene's transcription control region, is computed using all the motifs in the motif set $\mathcal{M}$.

2. *Regression/Classification:* This step is an application of logic regression with an appropriate model, e.g., linear regression, logistic regression or classification model to build a predictive model of the outcome variable Y.

We now describe these two steps in details.

2.2 Methods for step I of `LogicMotif`

If the set of all possible binding sites were known to us, then the task at hand would be to build a predictor for gene expression that includes the most predictive motifs. However, there is not yet a comprehensive set of motifs representing all TFBSs. Hence we first have to identify a set of potential sites. One of the popular approaches is to use a set different length oligonucleotides (Bussemaker et al.; 2001; Keleş et al.; 2002). Enumeration of all possible oligonucleotides up to a certain length allowing degeneracy is computationally prohibitive and similarly it is not possible to allow flexible motif structures such as gapped motifs. For this reason, we take advantage of the available cluster/group operating TFBS finding methods. These are utilized when the genes of interest are divided into groups. In our analysis that involved such groups of genes, we used van Helden et al. (1998) enumerative motif finding method `rsa-tools`. We review this method in subsection 2.2.1. For the cases when the gene group of interest is large and partitioning into smaller disjoint groups is not desirable or possible, we propose to use a simple univariate regression and extension method. This method is described in subsection 2.2.2.

Our combined approach allows us to first identify potential motifs and then select among these by using a regression/classification approach. Presumably any method for binding site identification can be used with the caveat that downstream analysis will depend on the quality of the obtained set of motifs. The only restriction is that a binary score has to be calculated for each motif representation. For example, one could use a method that identifies potential sites by position weight matrices and then reduce them to consensus sequences to calculate binary scores.

2.2.1 Motif finding by `rsa-tools`

van Helden et al. (1998)'s `rsa-tools` is based on oligonucleotide frequencies in a given set of co-expressed genes. It assigns a statistical significance value to each of the oligonucleotides that occur in the data based on a binomial model for the count data. The algorithm developed (available at `http://rsat.ulb.ac.be/rsat`) is very fast and allows a maximum oligonucleotide length of 8. In a later work, van Helden et al. (2000) extended the set of oligonucleotides to dyads (two piece of oligonucleotides of length 3 with a variable length in between). Although we used `rsa-tools` in some

of our analysis, other methods could have been used as well or motifs obtained by different methods can be pooled together to generate a richer set.

2.2.2 Motif finding by univariate regression and extension (MFURE)

We devised a simple motif finding method based on univariate regression adopting the extension procedure of Keleş et al. (2002). This approach, referred to here as MFURE, is especially useful for the cases where one has a large set of genes, e.g. > 200, that are differentially expressed at various time points of a time course experiment and the partitioning of the genes into non-overlapping sets is not possible or desirable. This method essentially uses all pentamers as seeds and fits a univariate linear regression model of the type

$$Y = \beta_0 + \beta_1 S_m + \epsilon,$$

where S_m represents the number of counts of pentamer m in a given transcription control region and Y is the gene expression. Each seed pentamer is extended by adding nucleotides to the right and/or left. Each extended motif is assessed by using the average residual sum of squares to determine if it represents a better motif than the seed motif. This procedure uses IUPAC nucleotide symbols at the extension step hence allowing discovery of degenerate motifs with a conserved core. Furthermore, it can be used with binary outcomes by using a univariate logistic regression model instead of a linear regression model. When the gene expression is measured over a time course, univariate regression and the extension procedure is applied at each time point using the gene expression from that time point as outcome.

2.3 Method for step II of Logicmotif

2.3.1 Logic regression

The logic regression methodology is proposed and studied extensively in Ruczinski et al. (2003). Here, we use this method in the context of binding site identification. Assume that there are a few interacting transcription factors for our experiment of interest and these require binding to different sites on the transcription control regions. We will assume that the interaction of these transcription factors, equivalently binding sites, can be reduced to a boolean expression. For instance, the transcription process might require that a gene should have binding sites for factor B and C or binding site for factor A in order to be regulated. This is represented in the tree structure of Figure 1(a). This tree returns an outcome of 1 if binding sites B and C or binding site A is present for a gene, otherwise it returns an outcome of 0. Similarly, the requirement for transcriptional regulation might be having sites A and B and D but

not C. The corresponding logic tree for this boolean expression is displayed in Figure 1(b).

[Figure 1 about here.]

We denote this new binary variable which is a boolean expression constructed from motif scores by L. Then the linear regression model given in (1) can be extended to allow combinatorial effects as

$$Y = \beta_0 + \beta_1 L_1 + \beta_2 L_2 \epsilon,$$

where L_1 and L_2 are boolean expressions obtained from the covariate vector $\overrightarrow{S}$.

The logic regression methodology identifies Boolean combinations of a given set of predictors (typically high dimensional) that are associated with an outcome. This method handles a variety of problems including linear regression, logistic regression and classification and can be extended to other problems by defining an appropriate score function. In the linear regression setting, the score function is the residual sum of squares and in the classification setting the scoring function is the misclassification rate. The logic regression algorithm implemented by Ruczinski et al. (2003) as a freely available `R` function uses simulated annealing to search through the high dimensional covariate space with a well defined move set and uses cross-validation and randomization base hypothesis testing to choose among different model sizes.

Step I of `LogicMotif` can be tuned further. Note that the TFBS finding procedures used in this step are likely to produce large sets of candidate motifs. If one wants to subset these set of motifs a priori to logic regression step (covariate reduction), a natural way to do so is by cross-validation. For each potential cut-off on the motif list, step 2 can be repeated with the set of motifs identified by the cutoff. Then average prediction or classification error of the logic regression models over the validation samples can be reported. The best cut-off is the one that is minimizing this cross-validated criteria.

3 Results

3.1 Performance on simulated data sets

We firstly assess the performance of our approach on simulated data sets that try to mimic the real life data sets. For this purpose we generated data in the following fashion: Firstly, $n_1 = 50$ and $n_2 = 50$ sequences of length 600bps were generated from a 0th order markov chain to represent the regulatory regions of up and down

regulated genes, respectively. Having generated these regulatory regions, we then created transcription regulation scenarios using the TFBSs available in the promoter database of *Saccharomyces cerevisiae* (Zhu and Zhang; 1999). Based on these transcription regulation scenarios which are in the form of boolean expressions we then generated gene expression for up and down regulated genes based on the model

$$Y = \beta_0 + \beta_1 L_1 + \beta_2 L_2 + \epsilon$$

where L_1 and L_2 represent boolean expressions of the transcription regulation mechanisms and ϵ is the error term generated from a normal distribution with mean 0 and standard deviation σ. Three simulations with different boolean expressions for transcriptional regulation were considered and the consensus sequences of the TFBSs used in these are given in Table 1.

- Simulation I: L_1 is set to `I(GCR1 or (GCN4 and CPF1))` and L_2 is empty. Transcriptional regulation requires either GCR1 or both of GCN4 and CPF1.
- Simulation II: L_1 is set to `I(PHO4 and ACE2 and (CuRE or RAP1`^{c}`))` and L_2 is empty. Transcriptional regulation requires having PHO4 and ACE2 and either having CuRE or not having RAP1.
- Simualtion III: L_1 is set to `I((SFF or PDR3) and ATF`^{c}`)`, and L_2 is set to `I(GCR1 and GCN4)`. Transcriptional regulation is an additive model of two terms.

In 90% of the up regulated genes, we implanted the corresponding TFBSs of the boolean expressions so that the evaluation of expression will return 1 indicating up-regulation. Similarly, to increase noise, we implanted in 10% of the down-regulated sequences the TFBSs from the boolean expressions. This mimics the scenario where not all of the co-expressed genes share common regulatory motifs. For the implantation of the motifs, if available, their corresponding position weight matrices are used otherwise an instance of the consensus is used. We also used two different values of σ to control the noise level in the generated microarray gene expression outcome Y. The results of these three simulated cases are reported in Table 2. In all of the cases, 5-fold cross-validation is used to select the number of logic trees and leaves. The covariate set used in logic regression included all the 50 consensus sequences in SCPD. Note that we included all SCPD TFBSs because running `rsa-tools` on these set of genes already identified the correct set of sites hence including all SCPD TFBSs extends this set. The results indicate that with a small noise level of $\sigma = 0.1$, logic regression identifies the correct boolean expressions in all of the cases. As the noise level increases, typically boolean expressions with smaller number of TFBS are selected. In the first two simulations, the identified boolean expression contains a subset of the true set of TFBSs. In the third simulation, two trees representing the

two additive boolean expressions were selected. One of the TFBS in the first identified boolean expression is not included in the corresponding true boolean expression however the PHO4 site which is replacing the ATF site of the true boolean expression has a consensus (CACGTK where K represents a G or a T) that highly overlaps with the consensus of the ATF site (ACGTCA). These limited simulations point out that, depending on the noise level, if the correct set of TFBSs are among the covariates of logic regression, logic regression is quite successful at identifying them. However, as the noise level increases, typically smaller models (boolean expressions with small number of TFBSs) are selected and finally highly correlated TFBSs can be can be substituted for each other. We also noticed that when the noise level is high, different runs of logic regression could obtain slightly different results. This is due to the stochastic nature of the simulated annealing algorithm used by logic regression. In our simulations, we ran logic regression three times for each data set and chose the model with the smallest cross-validation error.

[Table 1 about here.]

[Table 2 about here.]

3.2 Biological data sets

We have analyzed two different data sets using `LogicMotif`. For all data sets, 800bp upstream regions of the genes were used as regulatory regions and 5-fold cross-validation is employed in logic regression. Brief descriptions of these data sets are as follows:

α factor-based synchronized cell-cycle progression (Spellman et al.; 1998). Spellman et al. (1998) identified $\sim$ 800 yeast genes whose transcript levels vary periodically within the cell cycle. These genes are expressed in one or many phases of the cell cycle: early G1, G1, S, G2, M/G1. In our analysis we used 569 of these genes after filtering the ones that have overlapping transcription control regions with the other genes in the genome. The relative expression levels of these genes over α-factor time course experiments were used as outcomes. There are a total of 18 time points in the interval $[0 - 119]$ minutes and the difference between any two time points is 7 minutes. These time points cover two cycles of the cell cycle. Time points 0 to 56 mins correspond to the first cycle and 56 to 119 mins correspond to the second cycle.

Copper and iron deficiency data set of Freitas et al. (2004). Freitas et al. (2004) identified a set of 46 up regulated and 22 down regulated genes involved in

iron metabolism in yeast by combining their microarray gene expression data set with the publicly available data set by Rosetta Inpharmatics (Hughes et al.; 2000). Our analysis included these 68 yeast genes and we have used both their gene expression and binary class information (up/down) as outcomes.

3.3 Results for Spellman et al. (1998) data set

In the analysis of this time course data set, `MFURE` of Section 2.2.2 is first used to identify sets of potential binding sites at each time points using the gene expression as outcome. This method successfully identifies consensus sequences for the well known cell cycle regulators MCB (`ACGCG`), SCB (`CGCGAAA, CACGAAA`), SFF (`GTAAACAA`), STE12 (`TGAAACA`), ACE2 (`ACCAGC`) and partial matches to MCM1 (`TTTCCTAA,ATTTCC`). After pooling all the motifs generated at all time points (this provided a total of 631 motifs as a result of using `MFURE` with 512 pentamers), we use logic regression to build logic trees for all 18 time points using 631 binary predictors each of which corresponds to a motif. In all of our analysis, we treat a binding site and its reverse complement as identical.

The tree size, i.e., the total number of motifs in each tree, and the number of trees, i.e., the number of boolean expressions, are selected with 5-fold cross-validation. We allowed a maximum of 8 motifs distributed over a maximum of 3 trees. Evolution of cross-validation criteria (average residual sum of squares over the validation sample) indicated that it was not necessary to search for higher tree sizes. At all time points but 7, 49, 91 and 119 minutes a single tree was selected as the best tree. At time points 7, 49, and 91 three trees were selected whereas at time point 119 two trees were seleceted. For the time points with single trees, we compared the gene expression distribution among the groups with 0 ($L = 0$) and 1 ($L = 1$) boolean expression. Figure 2 displays box-plots of gene expression over these 14 time points for the $L = 0$ and $L = 1$ group.

[Figure 2 about here.]

[Figure 3 about here.]

These box-plots show that, in general, the mean gene expression in the $L = 0$ group is located around zero (except the time points 0 and 14 minutes) and the $L = 1$ groups has a positive mean gene expression across different time points. Moreover, Figure 3 displays box-plots of gene expression within all genes, $L = 0$ genes, and $L = 1$ genes at all time points separately. We performed a Wilcoxon rank sum test to test the hypothesis that the difference in the mean gene expression of the two groups ($L = 0$ and $L = 1$) is 0. The corresponding p-values are given at the title of each plot. All time

points had a significant p-value at the stringent threshold (0.05/14) obtained with Bonferroni correction. Most stable logic trees, in the sense that the trees generated are similar for the two cycles, were obtained for the time points that corresponded to the G1 phase. In particular, for the time points 14 and 77 minutes the selected logic tree corresponded to the boolean expression `I(ACGCG or (CGCGAAA or CACGAAA))` reflecting that MCB or SCB motif is sufficient for transcriptional regulation in G1 phase. More explicitly, this model states that

$$E[Y \mid \overrightarrow{S}] = \beta_0 + \beta_1(\texttt{I(ACGCG or (CGCGAAA or CACGAAA))}). \tag{3}$$

We note that this model is different from the following additive model

$$E[Y \mid \overrightarrow{S}] = \beta_0 + \beta_1 \texttt{I(ACGCG)} + \beta_2 \texttt{I(C\{G,A\}CGAAA)}. \tag{4}$$

Model (4) suggests that the expected gene expression for the genes which have both MCB and SCB motifs are higher than the expression of genes which have only SCB or MCB motif. Additionally, the "and" operator in model (3) successfully brings `CGCGAAA` and `CACGAAA`, the two possibilities of the SCB site, together.

[Figure 4 about here.]

[Figure 5 about here.]

Since the alpha factor-based synchronization consisted of two cell cycles, we would expect to discover these two cycles in the box-plots of the $L = 1$ genes. As seen in Figure 2, these two cycles are roughly covered. To explore this periodicity further, we plot box-plots of $L = 1$ genes of time point 14 minutes at all times points. This time point corresponds to G1 phase and the expression peak occurs at all G1 phases (time points 14, 21, 77, 84 minutes) as displayed in Figure 4. However, the periodicity signal can be lower for the other phases of the cell cycle. For instance, the same type of plot produced for $L = 1$ genes of time point 42 minutes (Figure 5) corresponding to G2 phase shows almost no signal of periodicity. The main reason for this is that the genes identified as regulated at this time point do not show a uniform behavior across other time points.

Among the time points with more than one logic trees, 3 of them are additive models of 3 single motifs and one of them is an additive model of 2 single motifs. Time point 7 minutes had an additive model of the motifs `GTCAACAA` (matches SFF consensus `GTMAACAA`), `CCAGAAAGGA` (partial match to MCM1), and `AGGGG` (matches STRE). MCM1 and SFF are known to promote gene expression at M/G1 phases thus our findings are consistent with the known results. The third motif that is contained in transcription control regions of many genes is also predicted to have inductive effect right after cell cycle arrest due to a stress respond. These four additive models are

given in Table 3 and these results mostly agree with the additive models obtained by Bussemaker et al. (2001) even though we are focusing on a smaller subset of genes by using cell cycle regulated genes in this analysis. The main difference is that Bussemaker et al. (2001) obtained larger additive models for these time points. In particular, they report 6, 8, 5, and 4 motifs for time points 7, 49, 91, 119 minutes, respectively. Some among these motifs are too short (3 base pairs) to represent a real biological site. One other reason for this discrepancy between the two methods might be due to the model selection criteria used by them. Bussemaker et al. (2001)' model sizes are based on p-values calculated from an extreme value distribution, and such an approach, in general, is likely to produce false positives if the multiple testing issues are not handled with caution. We use cross-validation for model selection and hence multiple testing is not an issue. In summary, the analysis of this data set revealed that logic regression is capable of identifying most relevant motifs from a given set of motifs as well as linear regression methods. Additionally, it is flexible enough to generate predictive models of gene expression with combinatorial interaction of binding sites.

[Table 3 about here.]

3.4 Results for Freitas et al. (2004) data set

This data set consists of 46 up regulated and 22 down regulated genes. Both binary class variable (indicator of up or down regulation) and continuous gene expression levels are available to use as outcome in the logic regression step. van Helden et al. (1998)' `rsa-tools` was used to extract potential binding sites. This resulted in a total of 74 motifs with widths between 6 and 8bps. For this data set, both the class variable and the continuous gene expression measurement were used as outcome. For both type of outcomes, model selection was performed by 5-fold cross-validation. The 5-fold cross-validation scores (average residual sum of squares over the validation sample) with continuous and binary outcomes are given in Figure 6.

[Figure 6 about here.]

As seen in Figure 6(a), the best model for the continuous outcome is of size 3 with 2 trees. Figure 7 displays these two logic trees and Table 4 provides the details of this model. This model is a linear regression model with two variables. The first variable is a single motif and the second variable is a boolean expression of two motifs. The first motif identified matches the Aft1p binding site identified by Yamaguchi-Iwai et al. (1996) and it has an inductive effect on the gene expression, i.e. positive regression coefficient. The transcriptional factor, Aft1p, plays a key

role in regulation of the uptake process. (Casas et al.; 1997). In iron deficiency, Aft1p induces transcription of multiple genes involved in iron uptake, intra-cellular transport, mobilization and recycling of heme iron (Casas et al.; 1997; Yamaguchi-Iwai et al.; 2002). The second variable is a boolean expression which identifies repressive effects of two motifs. Interestingly, one motif, `CCGCAA` is present in transcription control region of eleven genes: *YBR147w, Glt1, Cyc7,Met10, Leu1,YGL117w, Bio2, Ecm17,MSN4, Aco1*, and *YOR356w*. Of these, seven are known or predicted to encode FeS cluster proteins (see Table 6). Hence, this motif could represent a TFBS for an unknown TF involved in repression of genes encoding FeS proteins in iron deficiency. As suggested by an anonymous referee, we compared the fitted values of the logic regression model with the actual means of the four groups obtained by altering the two covariates in the model. Table 5 summarizes the observed and fitted values for these four cells. This comparison supports the additive effect. Moreover, logic regression fit of a logistic regression model with the binary outcome variable identifies an additive model that is almost the same as the linear regression model of Table 4 (except that `TGCACCC` is identified instead of `TGCACCSW`).

[Figure 7 about here.]

[Table 4 about here.]

[Table 5 about here.]

[Table 6 about here.]

The best tree using binary outcome variable, hence treating the problem as a classification problem, is a tree of size 3 (Figure 6(b)). When dealing with classification problems, the maximum tree size is 1 (single classification rule). The corresponding tree is given in Figure 8 and it is composed of the same motifs as the regression trees of Figure 7, with `TGCACCSW` having an inductive effect and the combination of `ACGTCG` and `CCGCAA` having a repressive effect. We use this resulting tree to classify 68 genes in the data set. Note that we would expect a good classification rate since the trees themselves are built using the same data. The real indicator of the predictive power is obtained from the cross-validation test scores. Since we se 5-fold cross-validation the lowest score is about ~ 3.5 and the average misclassification rate is 25% $(3.5/(68/5) \times 100)$. Classification results using the logic tree of Figure 8 and only using Aft1p site `TGCACCSW` are given Tables 7 and 8.

[Table 7 about here.]

[Figure 8 about here.]

We have compared the results of `LogicMotif` on this data set to the linear regression based method of Keleş et al. (2002). The final model obtained by this method is given in Table 9. The first motif selected matches the Aft1p site and it has an inductive effect as in the logic regression model. Similarly, the forth motif selected is `CCGCAA` and it has a repressive effect. However, the second repressive motif identified by logic regression is not identified by this approach and we could not find any exact matches to this motif in SCPD.

[Table 8 about here.]

4 Conclusion

We have presented an application of the newly developed logic regression methodology to the problem of binding site identification. In particular, we devised a systematic analysis method that we refer as `LogicMotif`. `LogicMotif` consists of two steps. The first step uses any available potential binding site identification tool or our method of univariate regression and extension (`MFURE`) and the second step builds regression or classification models using logic regression. The success of linear regression methods in motif finding has been illustrated by previous studies (Bussemaker et al.; 2001; Keleş et al.; 2002; Conlon et al.; 2003). The main strength of `LogicMotif` depends on the adaptability of the logic regression methodology since it is capable of creating more complex variables to include in a regression or classification model. Moreover, the first step is also flexible since it allows pooling of the motifs identified by various motif detection methods hence creating a richer covariate set. So far, we have used logic regression with binary covariates, however extension of other type of variables is straight forward since logic regression deals with categorical or continuous variables by creating dummy variables. In particular, since the reduction of position weight matrices into consensus sequences will typically reduce the amount of information contained in the position weight matrix, using continuous covariates with logic regression might be beneficial.

`LogicMotif` can directly be used to analyze microarray data from a single experiment by first applying the motif finding with univariate regression and extension method that we describe here. It is also suitable for pre-processed microarray data where genes are classified into two groups according to up and down regulation. Additionally, chromatin immunoprecipitation-microarray (ChIP-array) experiments (Ren et al.; 2000) are another type of data set where this method can be useful for identifying binding sites with complex structures. Clearly, the success of the entire method relies on the binding site detection method used in the first step. For this reason, pooling of the binding sites obtained by different methods is useful for generating a rich class of binding sites.

For the data sets that we have considered, the logic trees obtained were in general simple hence generating simple hypothesis for experimental testing. Moreover, there were cases where the selected models turned out to be linear regression models without any interactions. Our analysis suggest that this systematic approach provides a powerful and flexible method by combining cluster/group operating motif finding methods and the adaptive logic regression methodology.

Recently, there have been many interesting research on the topic of identifying *regulatory modules*, which are groups of TFBSs clustered together in the regulatory regions of the genomes. Some of the novel approaches that focus on this problem are by Bailey and Noble (2003); Sinha et al. (2003); Aerts et al. (2003). These methods, using only raw sequence data as input (and sometimes the actual position weight matrices of the TFBSs), aim to identify individual TFBSs and their closely spaced occurrences in the regulatory regions. We would like to point out that the problem we considered in this paper is slightly different. We are not focusing on regulatory modules but instead on the context dependent interactions of TFBSs. Module searching methods typically operate on only sequence data, our approach requires as input sequence data and class index such as up and down regulation or actual microarray gene expression outcome corresponding to two or more groups of genes. However, our framework could easily replace the cluster/group operating TFBS search method used in the first step by a module searching method or a combination of these, and then the question at hand would be identifying which modules or combinations of modules explain the outcome variable of interest the best.

Acknowledgements

The authors would like to thank two anonymous referees for their careful comments which improved the context and the presentation of the paper.

References

Aerts, S., van Loo, P., Thjs, G., Moreau, Y. and Moor, B. D. (2003). Computational detection of *cis*-regulatory modules, *Bioinformatics* **19**: ii5–ii14.

Bailey, T. and Elkan, C. (1995). Unsupervised learning of multiple motifs in biopolymers using expectation maximization, *Machine Learning* **21**(1-2): 51–80.

Bailey, T. L. and Noble, W. S. (2003). Searching for statistically significant regulatory modules, *Bioinformatics* **19**: ii16–ii25.

Bussemaker, H., Li, H. and Siggia, E. (2001). Regulatory element detection using correlation with expression, *Nature Genetics* **27**: 167–171.

Casas, C., Aldea, M., Espinet, C., Gallego, C., Gil, R. and Herrero, E. (1997). The AFT1 transcriptional factor is differentially required for expression of high-affinity iron uptake genes in *Saccharomyces cerevisiae*, *Yeast* **13**(7): 621–637.

Conlon, E., Liu, X., Lieb, J. and Liu, J. (2003). Integrating regulatory motif discovery and genome-wide expression analysis, *Proceedings of the National Academy of Sciences* **100**: 3339–3344.

Freitas, J. D., Kim, J., Poynton, H., Su, T., Wintz, H., Fox, T., Holman, P., Loguinov, A., Keleş, S., van der Laan, M. J. and Vulpe, C. (2004). Exploratory and confirmatory gene expression profiling of *mac1*, *Journal of Biological Chemistry* **2**(6): 4450–4458.

GuhaThakurta, D. and Stormo, G. (2001). Identifying target sites for cooperatively binding factors, *Bioinformatics* **17**(7): 608–621.

Hertz, G. and Stormo, G. (1999). Identifying dna and protein patterns with statistically significant alignments of multiple sequences, *Bioinformatics* **15**(7-8): 563–577.

Hughes, T., Marton, M., Jones, A., Roberts, C., Stoughton, R., Armour, C., Bennett, H., Coffey, E., Dai, H., He, Y., Kidd, M., King, A., Meyer, M., Slade, D., Lum, P., Stepaniants, S., Shoemaker, D., Gachotte, D., Chakraburtty, K., Simon, J., Bard, M. and Friend, S. (2000). Functional discovery via a compendium of expression profiles, *Cell* **102**: 109–126.

Keleş, S., van der Laan, M. and Eisen, M. (2002). Identification of regulatory elements using a feature selection method, *Bioinformatics* **18**(9): 1167–1175.

Keleş, S., van der Laan, M. J., Dudoit, S., Xing, B. and Eisen, M. B. (2003). Supervised detection of regulatory motifs in dna sequences, *Statistical Applications in Genetics and Molecular Biology* **2**(1). Article 5.

Lawrence, C., Altschul, S., Boguski, M., J.S. Liu, A. N. and Wootton, J. (1993). Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment, *Science* **262**: 208–214.

Lawrence, C. and Reilly, A. (1990). An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences, *Proteins: Structure, Function and Genetics* **7**: 41–51.

Li, H., Bussemaker, H. and Siggia, E. (2000). Building a dictionary for genomes: Identification of presumptive regulatory sites by statistical analysis, *Proceedings of the National Academy of Sciences USA* **97**(18): 10096–10100.

Liu, X., Brutlag, D. and Liu, J. (2002). An algorithm for finding protein-DNA binding sites with applications to chromatin immunoprecipitation microarray experiments, *Nature Biotechnology* **20**(8): 835–839.

Neuwald, A., Liu, J. and Lawrence, C. (1995). Gibbs motif sampling: Detection of bacterial outer membrane protein repeats, *Protein Science* **4**: 1618–1632.

Pilpel, Y., Sudarsanam, P. and Church, G. (2001). Identifying regulatory networks by combinatorial analysis of promoter elements, *Nature Genetics* **29**: 153–159.

Ren, B., Robert, F., Wyrick, J., Aparicio, O., Jennings, E., Simon, I., Zeitlinger, J., Schreiber, J., Hannett, N., Kanin, E., Volkert, T., Wilson, C., Bell, S. and Young, R. (2000). Genome-wide location and function of DNA binding proteins, *Science* **290**: 2306–2309.

Ruczinski, I., C., K. and M.L., L. (2003). Logic regression, *Journal of Computational and Graphical Statistics* **12**(3): 475–511.

Sinha, S. and Tompa, M. (2000). A statistical method for finding transcription factor binding sites, *Eighth International Conference on Intelligent Systems for Molecular Biology*, AAAI, San Diego, CA, pp. 344–354.

Sinha, S., van Nimwegen, E. and Siggia, E. (2003). A probabilistics method to detect regulatory modules, *Bioinformatics* **1**(1): 1–10.

Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D. and Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization, *Molecular Biology of the Cell* **9**(12): 3273–3297.

Tavazoie, S., Hughes, J., Campbell, M., Cho, R. and Church, G. (1999). Systematic determination of genetic network architecture, *Nature Genetics* **22**: 281–285.

Tompa, M. (1999). An exact method for finding short motifs in sequences, with application to the ribosome binding site problem, *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, AAAI, Heidelberg, Germany, pp. 262–271.

van Helden, J., Andre, B. and Collado-Vides, J. (1998). Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies, *Journal of Molecular Bilogy* **281**: 827–842.

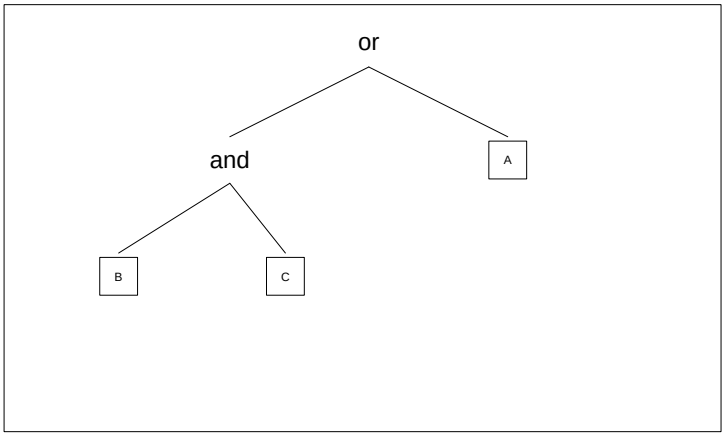
van Helden, J., Rios, A. F. and Collado-Vides, J. (2000). Discovering regulatory elements in intergenic sequences by analysis of spaced dyads, *Nucleic Acids Research* **28**(8): 1801–1818.

Yamaguchi-Iwai, Y., Stearman, R., Dancis, A. and Klausner, R. (1996). Iron-regulated DNA binding by the AFT1 protein controls the iron regulon in yeast, *The EMBO Journal* **15**: 3377–3384.

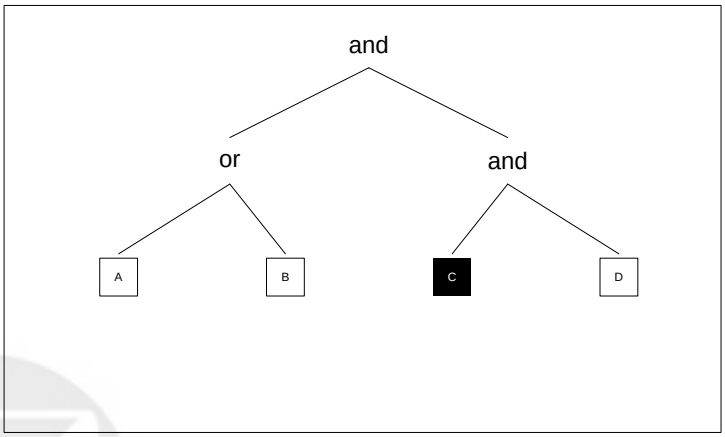
Yamaguchi-Iwai, Y., Ueta, R., Fukunaka, A. and Sasaki, R. (2002). Subcellular localization of AFT1 transcription factor responds to iron status in *Saccharomyces cerevisiae*, *Journal of biological chemistry* **227**(21): 18914–18918.

Zhu, J. and Zhang, M. (1999). Scpd: A promoter database of yeast saccharomyces cerevisiae, *Bioinformatics* **15**: 607–611.

List of Figures



(a) I(A **or** (B **and** C))



(b) I((A **or** B) **and** (C^c **and** D))

Figure 1: *Examples of logic trees.* C^c represents the complement of C, i.e. if the score for C is 1 then the score for C^c is 0. The black boxes are used to represent complements, i.e., "not" operator. $I(.)$ represents the indicator function that returns 1 if the expression evaluated is true and 0 otherwise.

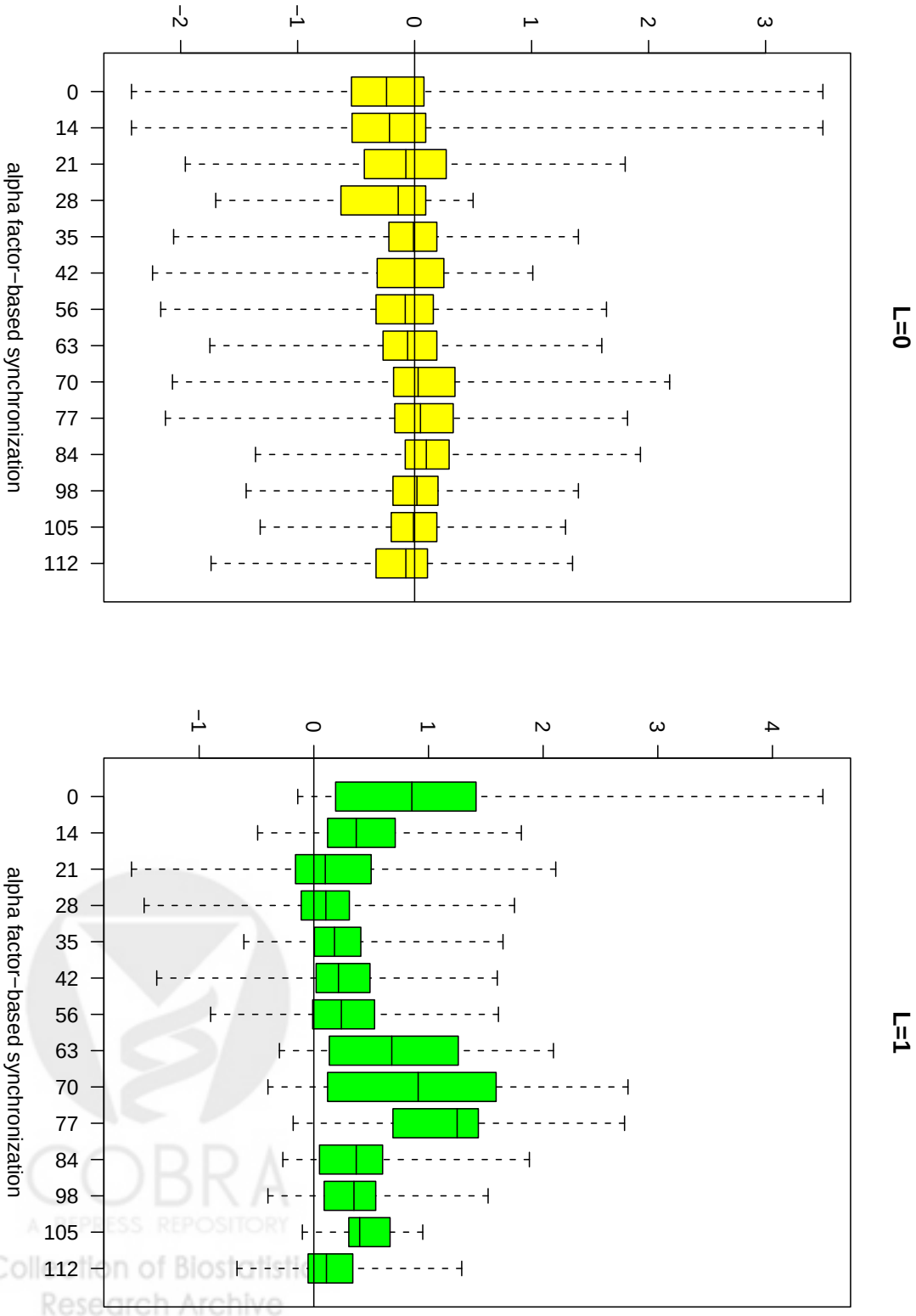


Figure 2: *Box-plots over two cell cycles.* Box-plots of the gene expression within groups with $L = 1$ and $L = 0$, respectively. Purple line is the 0 gene expression level.

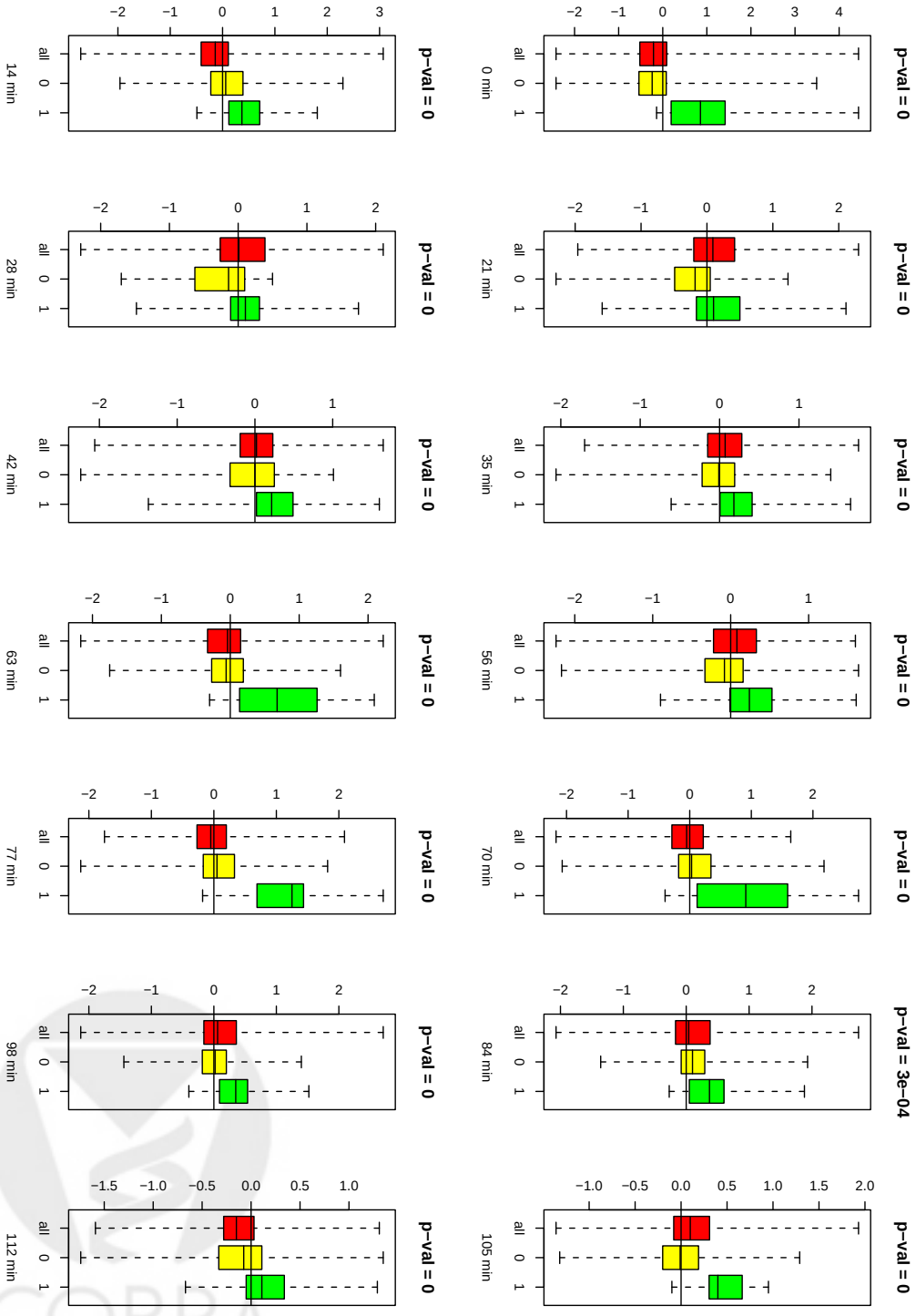


Figure 3: *Summary of the logic trees.* Box-plots of the gene expression within $L = 0$ group and $L = 1$ group at different time points. p-values are computed using Wilcoxon rank sum test (p-values smaller than 1e-4 are rounded to 0). Purple line represents the 0 gene expression level.

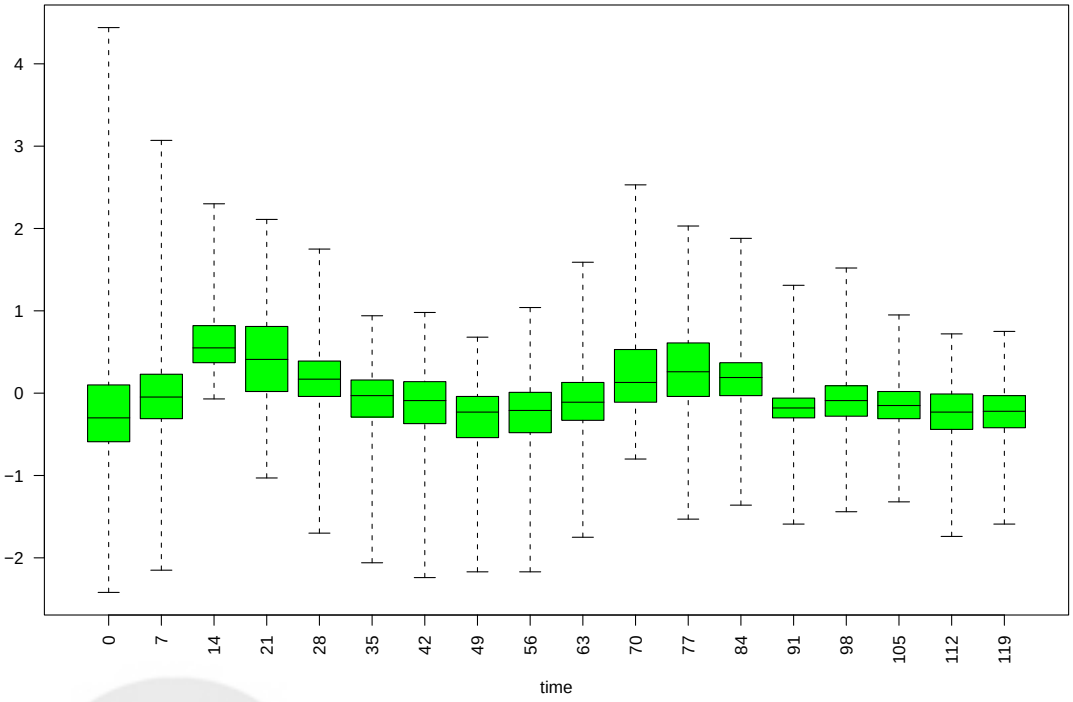


Figure 4: *Box-plots of the gene expression of $L = 1$ genes for the time point 14 minutes.*

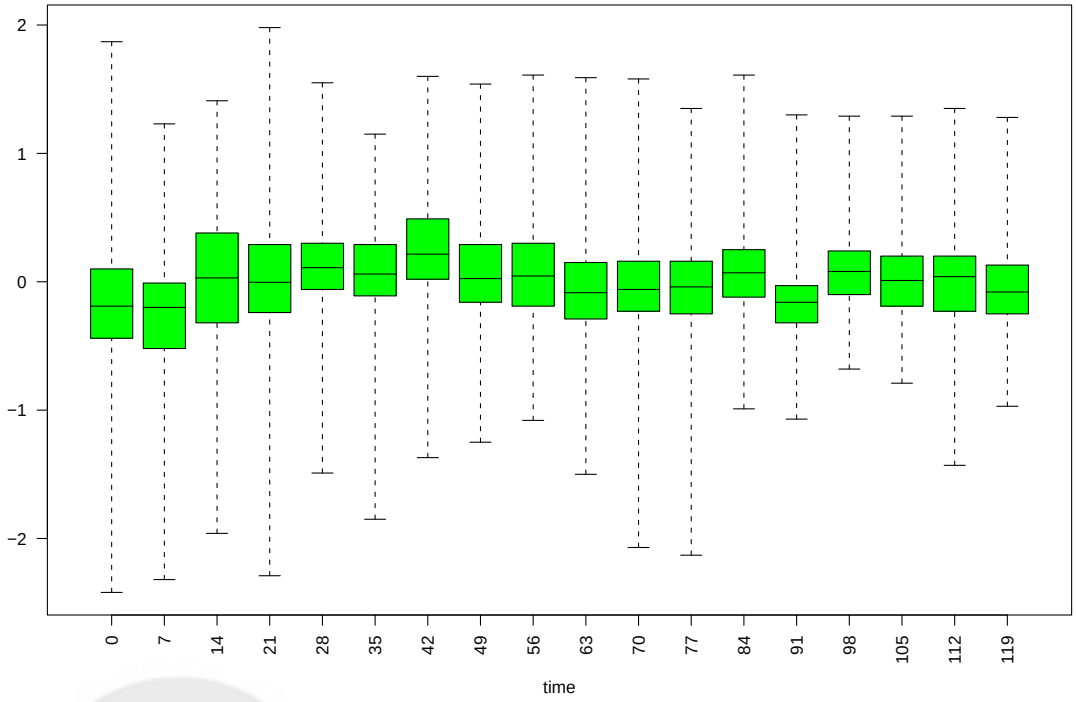
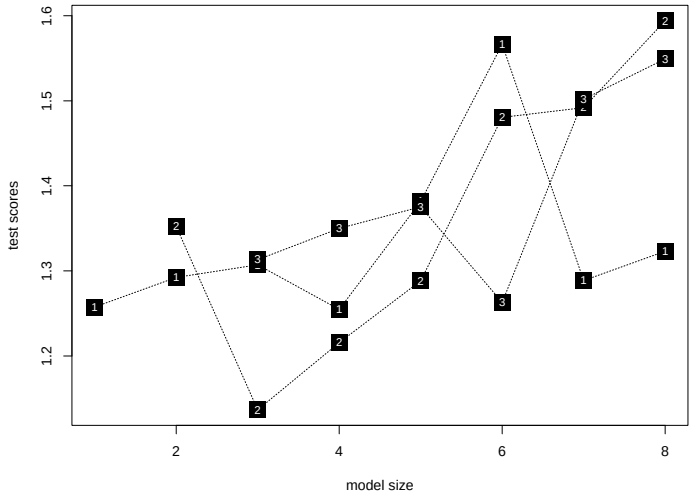
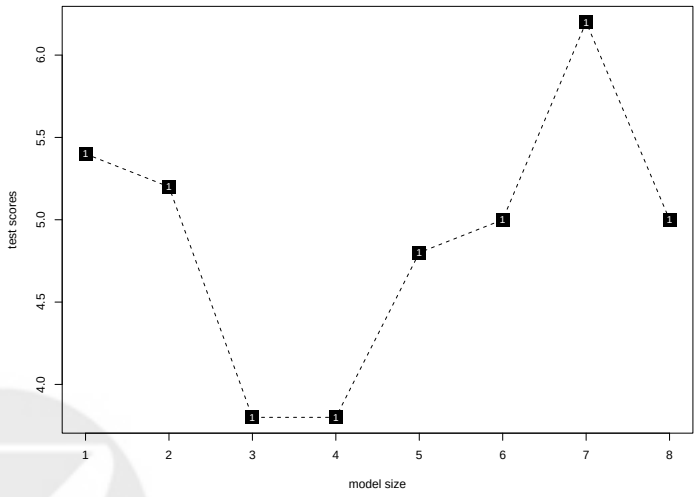


Figure 5: *Box-plots of the gene expression of $L = 1$ genes for the time point 42 minutes.*



(a) 5-fold cross-validation with continuous outcome



(b) 5-fold cross-validation scores binary outcome

Figure 6: *Model selection with continuous and binary outcome using 5-fold cross-validation:* Numbers in the boxes represent the number of trees. A single tree is considered with the binary outcome in the classification setting. Model size refers to the total number of motifs in all of the trees considered.

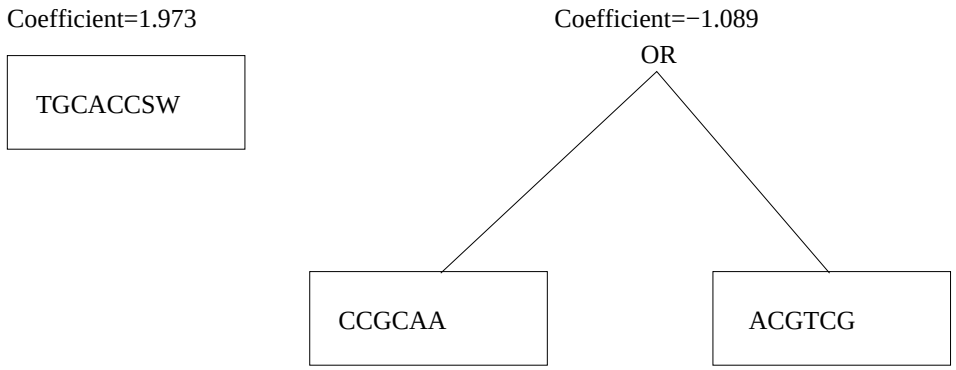


Figure 7: *Best logic tree with continuous outcome.* Logic regression model of size 3 with 2 logic trees.

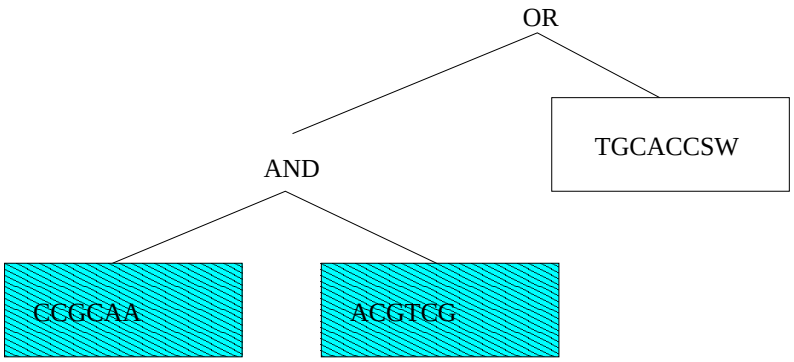


Figure 8: *Best logic tree with binary outcome.* Best logic tree is of size 3. This logic tree corresponds to the boolean expression I(`TGCACCSW` **or** (not `ACGTCG` **and** not `CCGCAA`)). The shaded boxes indicate a score of 0 for the corresponding motif.

List of Tables

TFBS	Consensus	Sampling method
GCR1	CWTCC	m
GCN4	TGANTN	m
CPF1	TCACGTG	c
PHO4	CACGTK	m
ACE2	GCTGGT	c
CuRE	GAGCAAA	c
RAP1	RMACCCA	m
SFF	GTMAACAA	c
PDR3	TCCGYGGA	m
ATF	ACGTCA	c

Table 1: *TFBSs from SCPD that are used in the simulation studies.* m: position weight matrix is used to sample an instance of the motif from the corresponding TFBS; c: consensus sequence is used for sampling.

Simulation I: $\beta_0 = 0.5, \beta_1 = 1$.

L_1	I(GCR1 or (GCN4 and CPF1))
$\hat{L_1}(0.1)$	I(GCR1 or (GCN4 and CPF1))
$\hat{L_1}(1)$	I(GCR1 or CPF1)

Simulation II: $\beta_0 = 0.5, \beta_1 = 1$.

L_1	I(PHO4 and ACE2 and (CuRE or RAP1^c))
$\hat{L_1}(0.1)$	I(PHO4 and ACE2 and (CuRE or RAP1^c))
$\hat{L_1}(1)$	I(PHO4 and ACE2 and CuRE)

Simulation III: $\beta_0 = 0.5, \beta_1 = 0.8, \beta_2 = 1$.

L_1	I((SFF or PDR3) and ATF^c)
$\hat{L_1}(0.1)$	I((SFF or PDR3) and ATF^c)
$\hat{L_1}(1)$	I((SFF or PDR3) and PHO4^c)
L_2	I(GCR1 and GCN4)
$\hat{L_2}(0.1)$	I(GCR1 and GCN4)
$\hat{L_2}(1)$	I(GCR1 and GCN4)

Table 2: *Logic regression results on different transcription regulation scenarios with simulated data.* $L_i, i = 1, 2$: True logic term; $\hat{L}_i(0.1)$: Estimated logic term when $\sigma = 0.1$; $\hat{L}_i(1)$: Estimated logic term when $\sigma = 1$

Table 3: *Additive models of individual motifs at time points 7, 49, 91 and 119 minutes.* These four additive models mostly agree with the additive models obtained by Bussemaker et al. (2001). The only discrepancy is that Bussemaker et al. (2001) in general have larger models for these time points.

Time point	Motifs in the additive model
7 min	`GTCAACAA` (SFF), `CCGAATTAGG` (MCM1), `AGGGG` (STRE)
49 min	`ACGCG` (MCB), `ACCAGC` (SWI5), `TTTCCTAATTA` (MCM1)
91 min	`ACCAGC` (SWI5), `ACGCGT` (MCB), `CGCGAAA(SCB)`
119 min	`ACGCG` (MCB), `CGCGAAA` (SCB)

Table 4: *Best logic regression with continuous outcome.* This model corresponds to the logic trees displayed in Figure 7 and has an R^2 of ~ 0.60.

Motif	Coef	Std Error	p-value
(Intercept)	0.5602	0.1530	5.05e-4
(TGCACCSW/WSGGTGCA)	1.9729	0.2432	1.86e-11
(CCGCAA/TTGCGG or ACGTCG/CGACGT)	-1.0893	0.2208	5.92e-06

Table 5: *Mean observed values versus fitted values corresponding to all outcome combinations of the two logic trees in logic regression model of Figure 7.* Fitted values are obtained using Table 4.

	Fitted	Observed
$L_1 = 1,\ L_2 = 1$	1.4438	1.48
$L_1 = 0,\ L_2 = 1$	-0.5290	-0.5392
$L_1 = 1,\ L_2 = 0$	2.5331	2.515
$L_1 = 0,\ L_2 = 0$	0.5602	0.5675

Table 6: *Genes containing the repressor motif* `CCGCAA`*:* * :has CDD 16482 and CDD10514 = FeS protein

ORF/Gene name	type/function
ybr147w	unknown
ydl171c GLt1	FeS protein
yel039c Cyc7	heme
yfr030w met10	FeS protein
ygl009c leu1	FeS protein
ygl117w	unknown
ygr282c Bio2	FeS protein
yjr137c Ecm17	FeS protein
ykl062w MSn4	unknown
ylr304c aco1	FeS protein
yor356w*	unknown

Table 7: Classification using the logic tree of Figure 8 with odds ratio 28.

	up	down
up	42	4
down	6	16

Table 8: Classification using `TGCACCSW/WSGGTGCA`.

	up	down
up	18	28
down	0	22

Table 9: *Linear regression model obtained by the method of Keleş et al. (2002).* The # of splits performed in Monte carlo cross-validation is 100.

Motif	Coef	Std Error	p-value
Intercept)	-0.4669	0.2142	0.0331
GCACCC/GGGTGC	0.7759	0.1068	7.39e-10
CAACC/GGTTG	0.3009	0.0975	0.0030
GTGCAA/TTGCAC	0.4646	0.1196	0.0002
CCGCAA/TTGCGC	-0.9217	0.1925	1.09e-05
AGGTGTA/TACACCT	1.1438	0.2855	0.0002