BIOINFORMATICS

Vol. 20 no. 5 2004, pages 750–757
DOI: 10.1093/bioinformatics/btg480

Regression trees for regulatory element identification

Tu Minh Phuong[1,2], Doheon Lee[1,] and Kwang Hyung Lee[1]*

[1]Department of BioSystems, Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong Yuseong-gu, Daejeon 305-701, Korea and [2]Faculty of Information Technology, Posts & Telecommunications Institute of Technology, Km 10 Nguyen Trai Road, Hatay, Vietnam

Received on July 17, 2003; revised on October 9, 2003; accepted on October 10, 2003
Advance Access publication January 29, 2004

ABSTRACT

Motivation: The transcription of a gene is largely determined by short sequence motifs that serve as binding sites for transcription factors. Recent findings suggest direct relationships between the motifs and gene expression levels. In this work, we present a method for identifying regulatory motifs. Our method makes use of tree-based techniques for recovering the relationships between motifs and gene expression levels.

Results: We treat regulatory motifs and gene expression levels as predictor variables and responses, respectively, and use a regression tree model to identify the structural relationships between them. The regression tree methodology is extended to handle responses from multiple experiments by modifying the split function. The significance of regulatory elements is determined by analyzing tree structures and using a variable importance measure. When applied to two data sets of the yeast *Saccharomyces cerevisiae*, the method successfully identifies most of the regulatory motifs that are known to control gene transcription under the given experimental conditions, and suggests several new putative motifs. Analysis of the tree structures also reconfirms several pairs of motifs that are known to regulate gene transcription in combination.

Availability: http://if.kaist.ac.kr/~phuong/RegTree

Contact: doheon@kaist.ac.kr

1 INTRODUCTION

Living cells respond to changing environmental conditions by regulating the expression of specific genes. This regulation occurs at several levels, one of which is transcriptional regulation. The transcription of a gene is controlled by diverse regulatory proteins called transcriptional factors (TFs), which bind to specific DNA sequences in the promoter region of the gene. Each TF recognizes a unique family of binding sites based on sequence binding preferences that arise through the energetic interactions between the atoms of the TF and those of the DNA sequence. The binding sites are short sequences (motifs) that average 5–20 bp in length. How a collection of TFs regulates the transcription of a gene depends to a large extent on the binding sites found in the gene's promoter. Hence, identifying and characterizing regulatory motifs that serve as TF binding sites is important for our understanding of the complex regulation of gene expression. Because experimental identification of regulatory motifs is difficult and time-consuming, researchers have long looked for computational approaches to this problem.

The main sources of data for studying regulatory elements are genome sequencing projects and DNA microarray data on the expression levels of many or all genes in a genome. A popular strategy is to look for conserved motifs upstream of genes that are believed to be co-regulated [reviewed in (Ohler and Niemann, 2001)]. First, genes with similar expression patterns across experimental conditions are grouped together by applying clustering analysis to genome-wide expression data sets (Eisen *et al*., 1998). The assumption here is that co-expressed genes are also co-regulated. Then, a motif discovery algorithm is used to search the sequences upstream of genes within each cluster for motifs that are common to them. These sequence motifs are plausible candidates for binding sites implicated in transcriptional regulation. There are many algorithms and methods that can be used to search for conserved sequences (Lawrence *et al*., 1993; Bailey and Elkan, 1995; van Helden *et al*., 1998; Sinha and Tompa, 2002). Bussemaker *et al*. (2001) noted that despite the success in the identification of many motifs, this strategy has a drawback: there are genes in the cluster without the motif, and many genes with the motif do not respond. To overcome this shortcoming, Holmes and Bruno (2000) suggested that researchers should cluster genes based on both gene expression patterns and promoter sequences. However, although their approach might theoretically overcome this limitation, no effective algorithm has yet been implemented to demonstrate its advantage in real data.

In a recent paper, Segal *et al*. (2003) described a probabilistic method for identifying modules of co-regulated genes together with their regulators. This method takes as input a set

*To whom correspondence should be addressed.

of candidate regulatory genes and a gene expression data set. Based on the assumption that the regulators are themselves transcriptionally regulated, the method uses the Expectation Maximization algorithm to search simultaneously for a partition of genes into modules and for a regulation program for each module. A Bayesian score is used to evaluate how well a regulation program can explain the expression behavior of the genes in the module as a function of the expression level of a small set of regulators. A motif finder then searches for conserved motifs upstream of the genes in each module.

Another approach to the identification of regulatory motifs is based on the association between gene expression values and the abundance of motifs (Bussemaker *et al*., 2001; Keles *et al*., 2002). This approach models expression levels for a single experiment as linear functions of sequence motifs. A linear regression procedure is used to fit the model and select the motifs that contribute most heavily to the model. The underlying assumption is that if a motif represents a functional binding site for an active TF, then the presence of the motif contributes additively to the expression level of the gene under the given experimental condition. Conlon *et al*. (2003) proposed a modification to this method by applying a sequence analysis algorithm for choosing only statistically over-represented motifs as inputs for the linear regression procedure.

In this paper, we formulate the problem in the regression framework and present a method for identifying regulatory motifs using tree-based regression models. The tree-based regression paradigm was introduced by Breiman *et al*. (1984) for dealing with a single response, and was later extended for use in handling multiple responses (Segal, 1992). When used for multiple responses, it is called a multivariate regression tree [some authors use the term 'multivariate tree' to refer to the classification trees whose splits are based on testing more than one variable (Broadley and Utgoff, 1995; Quinlan, 1993)]. Multivariate regression trees are useful when the goal is to identify strata with common covariate values and homogeneous multiple outcomes (Segal, 1992). In this work, we treat motif occurrences by the number of times motifs appear in gene promoters as predictor variables and the expression levels across different experimental conditions as multiple responses; then, we construct a multivariate regression tree that fits the data. The motifs that are important for fitting are then considered plausible regulatory motifs. We evaluate the importance of the motifs by analyzing the structure of the tree as well as using a technique based on surrogate splits.

2 METHODS

2.1 Regression tree methodology for a single response

In this section we give a brief introduction to the tree-based models. We refer the reader to Breiman *et al*. (1984) for further details. Suppose that there are p predictor variables $X_1, X_2, \ldots, X_p$ and a response Y. The values of X_j and Y are observed for n learning cases. In the context of motif identification, p variables X_i are the motifs, response Y is the expression level for a single time point and n learning cases are the genes.

A regression tree is a binary tree constructed by repeatedly splitting (sub)sets of learning cases into two descendant subsets. Each node of a tree contains a subset of cases. A node that does not have descendant nodes is a terminal node. The root node comprises the entire learning sample. The left and right child nodes contain disjoint subsets of the parent content and are defined by splitting the parent node. To construct a regression tree, the following must be specified:

(1) The rule to split a node.

(2) The method to determine the tree size.

Splitting is a critical step in tree-based techniques. Briefly, suppose that a predictor X_i is an ordered variable. Two subgroups result from answering the question 'is $x_i \leq c$?'. Cases for which the answer is 'yes' go to the left node, and those for which the answer is 'no' go to the right node. The cutoff value c is in the range of observed values of x_i. For a given number of predictor variables and a given set of cutoff values, there may be many allowable splits. A tree-growing algorithm chooses the best split for each node based on a split function $\phi(s, g)$ that can be evaluated for each split s in each node g. A split is chosen so as to get the distributions of responses in the child nodes are most homogeneous. Two such split functions are discussed by Breiman *et al*. (1984): least squares (LS) and least absolute deviation (LAD). Here we will focus on LS.

Let us assume that g is a node containing a subgroup of cases $\{(\mathbf{x}_i, y_i)\}$, where $\mathbf{x}_i^{\mathrm{t}} = (x_{i1}, \ldots, x_{ip})$, ($\mathbf{a}^{\mathrm{t}}$ is the transpose of vector $\mathbf{a}$), and that n_g is the total number of cases in g. The within node sum-of-squares is given by

$$SS(g) = \sum_{i \in g} (y_i - \bar{y}(g))^2 \qquad (1)$$

where

$$\bar{y}(g) = \frac{1}{n_g} \sum_{i \in g} y_i$$

For a split s that partitions g into left and right child nodes g_{L} and g_{R}, the LS split function is

$$\phi(s, g) = SS(g) - (SS(g_{\mathrm{L}}) + SS(g_{\mathrm{R}})) \qquad (2)$$

The best split s^* is that which maximizes $\phi(s, g)$:

$$s^* = \underset{s \in S}{\operatorname{argmax}}(\phi(s, g))$$

where S is the set of all allowable splits. The non-negativity of the split function ensures that recursive splitting will create smaller nodes with increased homogeneity. The algorithm proceeds recursively until some stop criterion is met. Typically, a minimum node size is specified or splitting

stops when $SS(g)$ drops below a certain level, e.g. 1% of the sum-of-square of the root node.

2.2 Multiple responses

Now consider a situation in which more than one response is observed. In such a multivariate regression setting, each learning case has both a vector of predictor variables and a vector of responses $\mathbf{y}_i^{\mathrm{t}} = (y_{i1}, \ldots, y_{iq})$, e.g. a vector of expression levels from multiple experiments. How then is the split function in Equation (2) to be generalized to this situation?

An obvious generalization is to rewrite the within-node sum-of-squares in Equation (1) as follows (Segal, 1992):

$$SS(g) = \sum_{i \in g} (\mathbf{y}_i - \bar{\mathbf{y}}(g))^{\mathrm{t}} \mathbf{V}^{-1} (\mathbf{y}_i - \bar{\mathbf{y}}(g)) \qquad (3)$$

where $\mathbf{V}$ is the covariance matrix of $\mathbf{y}_i$ in the root node and $\bar{\mathbf{y}}(g)$ is the average of $\mathbf{y}_i$ within node g. Then, the split function remains the same as in Equation (2). Segal (1992) noted that taking $\mathbf{V}$ to be the pooled sample covariance matrix of the responses in the root node corresponds to a two-sample Hotelling's T^2 statistic. He also noted that any two-sample statistic provides a split function that optimizes between-node separation rather than within-node homogeneity. With the new generalized split functions, the recursive algorithm proceeds as in the case with a single response.

2.3 Determining the tree size

A crucial aspect of tree construction is avoiding overfitting and underfitting, i.e. avoiding trees with too many nodes or too few nodes. Breiman *et al.* (1984) proposed a pruning algorithm that determines the tree size as follows: (a) initially, grow a large tree; (b) starting with the initial tree, define a nested sequence of its subtrees using cost-complexity; (c) select an optimal subtree from this sequence by cross-validation.

For step (b), a tree cost-complexity must be defined. Let $R(g)$ be the cost of a node g. Then, the cost of a tree G is defined as $R(G) = \sum_{g \in \tilde{G}} R(g)$, where $\tilde{G}$ is the set of terminal nodes of G. Further, define the complexity of G as the number of terminal nodes $|\tilde{G}|$. Then, the cost-complexity of G is defined as

$$R_\alpha(G) = R(G) + \alpha|\tilde{G}|, \quad \alpha > 0, \qquad (4)$$

where α is the complexity parameter that penalizes the number of the terminal nodes.

For the case of multiple responses, Zhang (1998) introduced the following cost function for $\phi(s, g)$

$$R(G) = \sum_{g \in \tilde{G}} \sum_{i \in g} (\mathbf{y}_i - \bar{\mathbf{y}}(g))^{\mathrm{t}} \mathbf{V}^{-1} (\mathbf{y}_i - \bar{\mathbf{y}}(g)) \qquad (5)$$

where $\mathbf{V}$ and $\bar{\mathbf{y}}(g)$ are estimated from the learning sample (based on which the initial tree is grown). Once $R(G)$ is defined, steps (b) and (c) are carried out as described in (Breiman *et al.*, 1984).

2.4 Surrogate splits and variable importance

In general, regression analysis can have two purposes: (1) to predict the values of the response variables in the future; (2) to understand the structural relationships between the responses and the measured variables. Our main purpose is not to predict but to discover those predictor variables (motifs) that are most relevant to the responses. Hence, a question of interest is: which variables are the most important? In other words, how do we rank those variables that, while not giving the best split of a node and thus do not appear in the tree structure, may give the second-best or third-best split. By using such variables to split the node we can obtain a tree that is almost as accurate as the original tree.

Breiman *et al.* (1984) proposed a measure of variable importance based on *surrogate splits*. For a given node g, suppose that s^* is the optimal split. The surrogate split $\tilde{s}_m$ on variable X_m for s^* is defined as the split that best predicts the results of s^* in comparison with other splits on X_m. That is, the child nodes that result from splitting g by $\tilde{s}_m$ show the greatest intersection with those that result from splitting g by s^*. Then, the measure of importance of variable X_m is defined as

$$M(X_m) = \sum_{g \in G} \phi(\tilde{s}_m, g) \qquad (6)$$

where ϕ is the split function. If there is more than one surrogate split on X_m at a node, we take the one with the larger ϕ.

3 IMPLEMENTATION AND RESULTS

3.1 Regression trees and regulatory motif discovery

When a transcription factor binds to an appropriate motif, it regulates the expression level of the respective gene. Thus, motifs can be considered as predictors that explain changes in expression levels. For a gene i, we introduce a vector of predictor variables $\mathbf{x}_i^{\mathrm{t}} = (x_{i1}, \ldots, x_{ip})$, where x_{ij} is the number of times motif j appears in the promoter of i. The vector of responses for i is $\mathbf{y}_i^{\mathrm{t}} = (y_{i1}, \ldots, y_{iq})$, where y_{ij} is the logarithm base 2 of the expression level of gene i in sample point j.

To build the tree model, first we need to decide which sequence motifs will be motif candidates and therefore will be predictor variables. The simplest way is to enumerate all the sequences in the genes' promoters as potential motifs (Bussemaker *et al.*, 2001; Keles *et al.*, 2002). However, this method has several drawbacks. First, because the number of all possible sequences is very large, only short sequences (up to 8 bp) can be considered. Second, most of considered sequences are not present in the final models while they can make noise that affect the model selection process.

Here, we take a more reasonable approach to selecting candidate motifs. Our method is similar to that proposed by Conlon *et al.* (2003). In particular, we take only sequences

that are conserved over genes. These motifs can be found using a motif finding algorithm such as AlignACE (Hughes *et al*., 2000) or MEME (Bailey and Elkan, 1995). The details of motif selection are given below.

3.2 Data sets

3.2.1 Candidate motifs For experiments, we used the set of 356 motifs compiled by Pilpel *et al*. (2001) as candidate motifs. The motif matrices are derived by applying the motif finding program AlignACE to the upstream regions of genes in the MIPS (Mewes *et al*., 2000) functional categories. Of the 356 motifs, 25 are known motifs described in the biological literature. The details of motif collection and the list of all 356 motifs with motif matrices can be found at http://genetics.med.harvard.edu/~tpilpel/MotComb.html

For each motif, the number of times the motif appears in the promoter regions of genes is counted by applying program ScanACE (Hughes *et al*., 2000). This program takes a matrix of motifs and scans a set of target sequences (promoters of the genes in this case) for similar motifs. When applying ScanACE to the promoters of the genes in *Saccharomyces cerevisiae*, Pilpel *et al*. (2001) found 4483 promoter regions that contain motifs from the above set. We counted x_{ij} from these data.

3.2.2 Microarray data We tested the tree model on two sets of microarray data for the yeast *S.cerevisiae*, specifically, the cell cycle data set of Cho *et al*. (1998) and the sporulation data set of Chu *et al*. (1998).

Cho *et al*. (1998) collected gene expression level data at 17 time points separated by 10 min intervals across two full cell cycles. Following Tavazoie *et al*. (1999), we discarded two time points (90 and 100 min) due to less efficient labeling of their mRNA during the hybridization. From the total 6220 genes, the 3000 most-variable genes were selected. In this selection, the metric of variation was the ratio between the standard variation and mean of the expression levels of each gene across the time points. Of the 3000 genes retained, only 2584 genes contain motifs from our set of candidate motifs.

We selected only genes that have observed data for at least 80% of sample points. The missing values were replaces with the mean of the observed data over sample points. In both experimental data sets, the percentage of missing elements is very low and it is therefore unlikely that our conclusions are affected by missing data. The expression data were then normalized so that the sum of squared values of expression levels for each gene was equal to unity [see e.g. Tavazoie *et al*. (1999) for details].

The second data set was taken from http://cmgm.stanford.edu/pbrown/sporulation. This data set consists of the expression levels of about 6200 genes over 10 sample points during meiosis and spore formation. Following Eisen *et al*. (1998), we selected the 2473 most-variable genes and applied the selection and transformation procedures described above to this data matrix.

3.3 Tree construction

During the growing phase we do not partition any node with less than 60 genes. In addition, we do not consider splits that result in nodes with less than 30 genes (i.e. about 1% of all genes). That ensures a reasonable number of genes in every node to make conclusion about motifs. After growing initial trees by using $\phi(s, g)$ as the split function, we pruned the trees with the cost function $R(G)$. We used 10-fold cross-validation to estimate the subtree costs and standard errors.

3.4 Results

The final trees of the cell cycle and sporulation data sets are shown in Figures 1 and 2, respectively. In these figures, the numbers inside circles are the node indices. The split used for a node is shown below the circle in the form '*motif i name* $\Leftarrow n$', which means that genes with more than n instances of motif i in their promoters go to the right node, and other genes go to the left node. For putative motifs with long names, we show only the motif number. The names of these motifs are given in Table 3.[1]

The trees tend to grow leftward, with less genes going to the right child node at each split. This is normal for our application, in which the known functional regulatory motifs are present in only a small number of gene promoters. For example, of the 2584 genes in the first data set, only 124 genes have motif MCB in their promoters. The remaining 2460 genes, therefore, go to the left node 1.

Among the motifs in the tree structure, MCB, SCB, ECB and MCM1 were mentioned previously by Cho *et al*. (1998). In particular, motifs MCB and ECB have a strong effect on transcription in the late G1 phase, SCB is active in the early G1 phase, and MCM1 plays a regulatory role during the G2 and M phases. Although the tree model does not give direct information about when a motif is active, it can be inferred by analyzing the mean temporal profile (Tavazoie *et al*., 1999) of the node containing the motif (the right child node after splitting a node using the motif). The motif AAAANGTAAACAA that has a high score in Cho *et al*. (1998) is very similar to the motif SFF (GTAAACAAA). Other motifs—RAP1, MET31-32 and mRRPE(M3A)—were found by Tavazoie *et al*. (1999), who applied clustering analysis to the same data set. Our tree model failed to identify motif STRE, which was found by these authors.

We ranked all the motifs using the variable importance measure based on surrogate splits. The top 20 motifs for the cell cycle data set are listed in Table 1 with their normalized importance measures M. The motivation for analyzing this table is to identify additional motifs that do not appear in

[1]Motif names are taken from Pilpel *et al*. (2001), see our website for sequences.

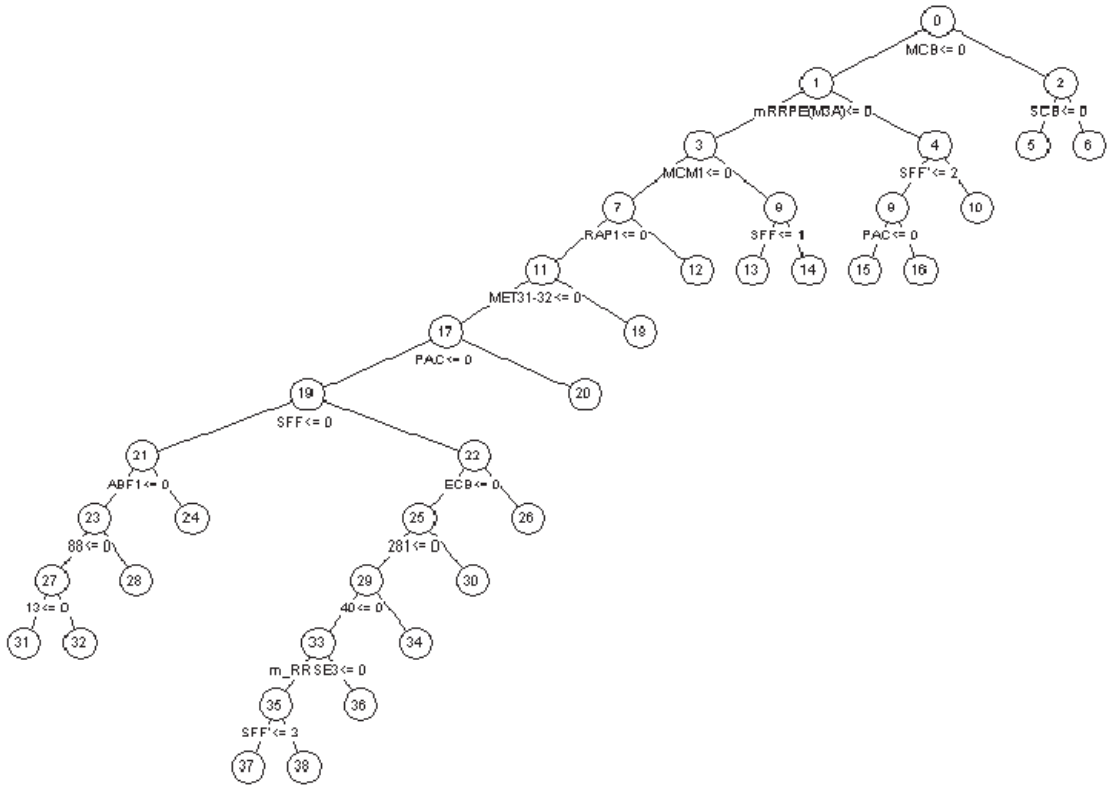

Fig. 1. Regression tree for the cell cycle data set.

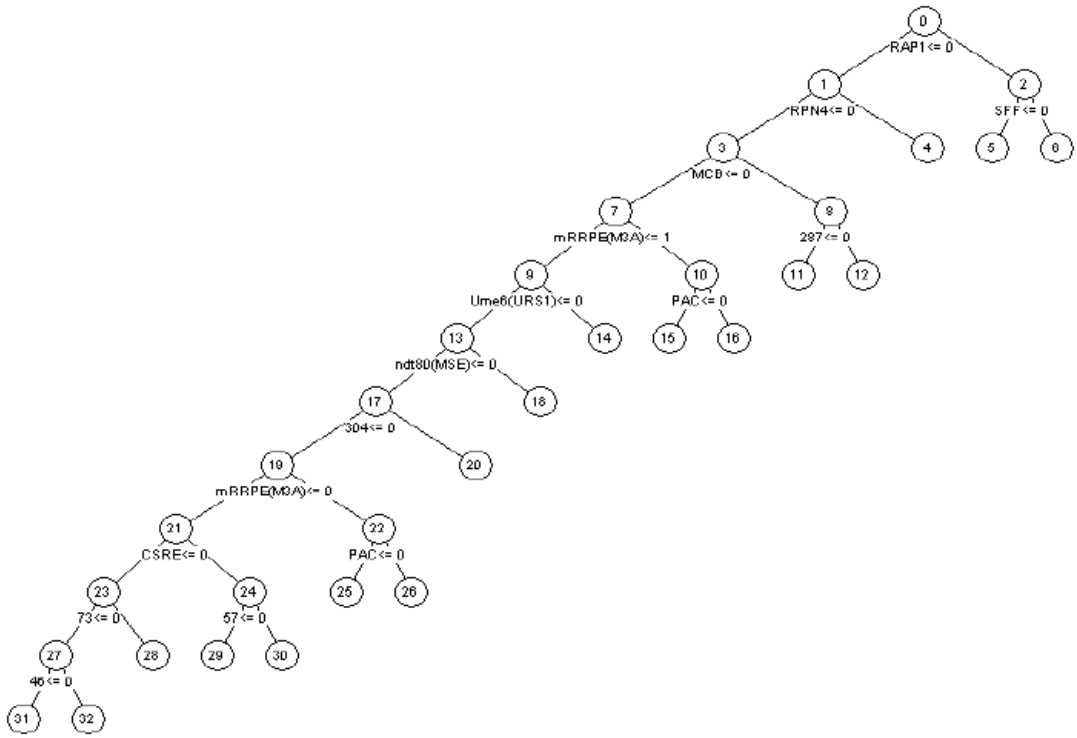

Fig. 2. Regression tree for the sporulation data set.

Table 1. Top ranked motifs for the cell cycle data set using surrogate splits: M is the importance measure and putative motifs are given by the motif numbers

Motif	*M*	Motif	*M*
MCB	1	ABF1	0.172
SFF	0.40	13	0.161
mRRPE(M3A)	0.379	246	0.152
PAC	0.363	MET31-32	0.150
MCM1	0.357	SCB	0.137
SFF	0.330	96	0.133
102	0.197	MCM1	0.125
ECB	0.193	184	0.116
m_RRSE3	0.183	88	0.114
RAP1	0.175	221	0.112

Table 2. Top ranked motifs for the sporulation data set

Motif	*M*	Motif	*M*
RAP1	1	154	0.176
mRRPE(M3A)	0.782	m_RPE58	0.163
RPN4	0.761	176	0.162
MCB	0.641	PAC	0.161
m_RPE72	0.355	SFF	0.161
m_RPE52	0.269	31	0.157
HSE	0.261	m_RPE8	0.153
Ume6(URS1)	0.248	57	0.148
304	0.184	141	0.142
ndt80(MSE)	0.178	49	0.139

the tree structure but that may be relevant to the expression data. We found that most motifs that appear in the tree have high ranking in Table 1. There are several additional putative motifs, which are ranked lower.

For the sporulation data set, the most important motifs identified by the regression tree are RPN4, RAP1, MCB, URS1, Ntd80(MSE), SFF, mRRPE(M3A) and PAC. Regulatory elements MSE and URS1 are known to play active roles in the regulation of gene expression during sporulation (Chu *et al*., 1998). Another motif, MCB, is one of the highly scored motifs found in (Bussemaker *et al*., 2001). The top 20 ranked variables for the sporulation data set are shown in Table 2.

The trees obtained for the cell cycle and sporulation data sets also suggest some previously unknown motifs: numbers 13, 40, 88 and 281 in Figure 1, and numbers 46, 57, 73, 287 in Figure 2 (names are given in Table 3). Some of these motifs possibly have role in transcription regulation.

In Figure 1, we can see several nodes that are defined by paths containing combinations of motifs. For instance, node 6 results from the path containing MCB and SCB, node 16 corresponds to a combination of mRRPE(M3A) and PAC, node 14 corresponds to MCM1 and SFF, and node 26 corresponds to SFF and ECB. These combinations of motifs are consistent with those reported by Pilpel *et al*. (2001). From the tree in Figure 2, we find one motif combination—mRRPE(M3A) and PAC—which is among the findings of Pilpel *et al*. (2001).

As regression trees split genes into homogeneous subgroups, it is interesting to examine the groups of genes within the terminal nodes. For each terminal node we plotted expression profiles of the genes in the node and calculated deviation (not shown here due to lack of space). We found that terminal nodes located closer to the root are more homogeneous than those located further from the root. The analysis also shows that while most genes within a terminal node have similar expression profiles, the expression profiles of some genes deviate from the mean. This tendency is stronger for terminal nodes that are more distant from the root, which may reflect the fact that splits closer to the root are more important and that their respective motifs have a stronger effect in transcriptional regulation.

4 DISCUSSION

In this study we have developed a new approach to the identification of regulatory elements based on regression trees. Tree-based methods are more suitable than parametric methods when the data set is large both in terms of the number of observations and the number of variables, which is the norm for genetics data. To handle expression data from multiple experiments, we adopted the approach proposed by Segal (1992). This approach extends the traditional tree model to cases with multiple responses by introducing generalized split functions. As a special case, our method can be applied to data from a single experimental condition; in that case, it is more similar to the approach of Bussemaker *et al*. (2001).

Our approach is similar to that used in previous studies (Bussemaker *et al*., 2001; Keles *et al*., 2002; Conlon *et al*., 2003) in that it considers motifs as predictors, gene expression levels as responses and then uses regression fitting to extract most relevant predictors. However, unlike the prior works, the tree model does not require the linearity assumption or any assumption about the relationships between variables. More importantly, by using the multivariate regression tree model, our approach can simultaneously handle expression data collected over multiple experiments. As noted by Bussemaker *et al*. (2001), using expression data from multiple experiments reduces the negative effect of noise and missing data. Another characteristic of the model is that it clusters genes into homogeneous groups when constructs trees. However, clustering here is different from that used by Eisen *et al*., (1999) in that it considers simultaneously expression data and motifs; clustering is integrated directly with motif identification.

Experiments in which the proposed approach was applied to two-data sets of the yeast *S.cerevisiae* demonstrated the ability of our method to identify biologically verified regulatory motifs. For the cell cycle and sporulation data sets, we identified most of the motifs known to be active in the given experimental conditions and suggest several new putative

Table 3. Putative motifs from tree structures and importance ranking

	Motif		Motif
13	m_ion_transporters_orfnum2SD_n7	46	m_breakdn_of_lipids_fatty_acids_and_isoprenoids_orfnum2SD_n8
31	m_pentose-phosphate_pathway_orfnum2SD_n7	49	m_regulation_of_amino-acid_metabol_orfnum2SD_n11
40	m_biogenesis_of_cytoskeleton_orfnum2SD_n5	88	m_regulat_of_lipid_fatty-acid_isoprenoid_biosynth_orfnum2SD_n8
73	m_metal_ion_transporters_orfnum2SD_n10	96	m_other_proteolytic_degradation_orfnum2SD_n2
57	m_organization_of_cytoplasm_orfnum2SD_n27	102	m_other_transcription_activities_orfnum2SD_n5
141	m_glyoxylate_cycle_orfnum2SD_n19	176	m_organization_of_cell_wall_orfnum2SD_n20
221	m_anion_transporters_orfnum2SD_n19	184	m_pheromone_response_generation_orfnum2SD_n12
287	m_g-proteins_orfnum2SD_n11	246	m_regulation_of_amino-acid_metabolism_orfnum2SD_n15
154	m_peroxisomal_organization_orfnum2SD_n28	281	m_other_energy_generation_activities_orfnum2SD_n4

regulatory elements. The tree models additionally revealed several motif combinations, which are known to have combinatorial effects on transcriptional regulation. Our approach of filtering motif candidates prior to tree construction reduced the computational complexity while simultaneously increasing the specificity of the results.

Despite these successes, our method fails to identify certain motifs, e.g. STRE from the cell cycle data set. A possible reason is that the hierarchical nature of tree construction makes it unable to capture transcriptional responses that are a superposition of independent processes. If several motifs participate independently in the same regulatory event, the tree model generally picks up only one representative.

Although in the experiments we constructed trees from the putative/known motifs compiled by Pilpel *et al*. (2001), our approach will probably be able to take as input any set of candidate motifs that are statistically significant over genes. In combination with a method that can prepare such a set of motifs, our approach can identify the motifs (including *de novo* ones) that are significant for given microarray experiments. Due to similarity with the clustering approach in splitting genes into homogeneous groups our approach will succeed in collections of microarray experiments that allow differentiating clusters of genes based on their expression profiles.

ACKNOWLEDGEMENTS

We would like to thank Jong H. Park for his comments on an earlier version of the paper, and the IBM SUR program for providing computing facilities for this research. T.M.P. was also supported by the Korea Foundation for Advanced Studies. This work was supported by the Korean Systems Biology Research Grant (M10309020000-03B5002-00000) from the Ministry of Science and Technology.

REFERENCES

Bailey,T.L. and Elkan,C. (1995) Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Mach. Learning*, **21**, 51–80.

Breiman,L., Friedman,J.H., Olshen,R.A. and Stone,C.J. (1984) *Classification and Regression Trees*. Wadsworth, Belmont, CA.

Broadley,C.A. and Utgoff,P.E. (1995) Multivariate decision trees. *Mach. Learning*, **19**, 45–77.

Bussemaker,H.M., Li,H. and Siggia,E.R. (2001) Regulatory element detection using correlation with expression. *Nat. Genet.*, **27**, 167–171.

Cho,R.J., Campbell,M.J., Winzeler,E.A., Steinmetz,L., Conway,A., Wodicka,L., Wolfsberg,T.G., Gabrielian,A.E., Landsman,D., Lockhart,D.J. and Davis,R.W. (1998) A genome-wide transcription analysis of the mitotic cell cycle. *Mol. Cell*, **2**, 65–73.

Chu,S., DeRisi,J., Eisen,M., Mulholland,J., Botstein,D., Brown,P.O. and Herskowitz,I. (1998) The transcriptional program of sporulation in budding yeast. *Science*, **282**, 699–705.

Conlon,E.M., Liu,X.S., Lieb,J.D. and Liu,J.S. (2003) Integrating regulatory motif discovery and genome-wide expression analysis. *Proc. Natl Acad. Sci. USA*, **100**, 3339–3344.

Eisen,M.B., Spellman,P.T., Brown,P.O. and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, **95**, 14863–14868.

Holmes,I. and Bruno,W.J. (2000) Finding regulatory elements using joint likelihoods for sequence and expression profile data. *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*. AAAI, New York.

Hughes,J.D., Estep,P.W., Tavazoie,S. and Church,G.M. (2000) Computational identification of *cis*-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *J. Mol. Biol.*, **296**, 1205–1214.

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

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

Mewes,H.W., Frishman,D., Gruber,C., Geier,B., Haase,D., Kaps,A., Lemcke,K., Mannhaupt,G., Pfeiffer,F., Schuller,C., Stocker,S. and Weil,B. (2000) MIPS: a database for genomes and protein sequences. *Nucleic Acids Res.*, **28**, 37–40.

Ohler,U. and Niemann,H. (2001) Identification and analysis of eukaryotic promoters: recent computational approaches. *Trends Genet.*, **17**, 56–60.

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

Quinlan,J.R. (1993) C4.5 *Programs for Machine Learning*. Morgan Kaufman, San Mateo, CA.

Segal,E., Shapira,M., Regev,A., Pe'er,D., Botstein,D., Koller,D. and Friedman,N. (2003) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet.*, **34**, 166–176.

Segal,M.R. (1992) Tree-structured methods for longitudinal data. *J. Am. Stat. Assoc.*, **87**, 407–418.

Sinha,S. and Tompa,M. (2002) Discovery of novel transcription factor binding sites by statistical overrespresentation. *Nucleic Acids Res.*, **30**, 5549–5560.

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

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

Zhang,H. (1998) Classification trees for multiple binary responses. *J. Am. Stat. Assoc.*, **93**, 441, 180–193.