Report Number: WUCSE-2004-43

2004-07-18

Discovering Transcriptional Regulatory Rules from Gene Expression and TF-DNA Binding Data by Decision Tree Learning

Jianhua Ruan and Weixiong Zhang

Background: One of the most promising but challenging task in the post-genomic era is to reconstruct the transcriptional regulatory networks. The goal is to reveal, for each gene that responds to a certain biological event, which transcription factors affect its transcription, and how several transcription factors coordinate to accomplish specific regulations. Results: Here we propose a supervised machine learning approach to address these questions. We build decision trees to associate the expression level of a gene with the transcription factor binding data of its promoter. From the decision trees, we extract regulatory rules that specify how the binding of... **Read complete abstract on page 2.**

Discovering Transcriptional Regulatory Rules from Gene Expression and TF-DNA Binding Data by Decision Tree Learning

Jianhua Ruan and Weixiong Zhang

Complete Abstract:

Background: One of the most promising but challenging task in the post-genomic era is to reconstruct the transcriptional regulatory networks. The goal is to reveal, for each gene that responds to a certain biological event, which transcription factors affect its transcription, and how several transcription factors coordinate to accomplish specific regulations. Results: Here we propose a supervised machine learning approach to address these questions. We build decision trees to associate the expression level of a gene with the transcription factor binding data of its promoter. From the decision trees, we extract regulatory rules that specify how the binding of a combination of several transcription factors affects the expression of a gene. Such rules are easy to interpret, and represent experimentally testable hypotheses. We use a decision tree ensemble approach to increase modeling accuracy and robustness. We also propose a novel method to integrate rules learned from several time series that measure the same biological processes. We apply our method to publicly available cell cycle expression data and transcription factor binding data for the budding yeast. Cross-validation experiments show that our method is highly accurate and reliable. The method correctly identifies all major known yeast cell cycle transcription factors, and assigns them into appropriate cell cycle phases. It also explicitly reveals synergetic relationships of transcription factors, most of which agree well with existing literatures, while the rest provide testable biological hypotheses. Conclusions: The high accuracy of our method indicates that our method is valid and that the learned regulatory rules can be used as the basic building elements of a transcriptional regulatory network. As more and more gene expression and TF binding data are available, we believe that our method will be useful for reconstructing large scale transcriptional regulatory networks.

2004-43

Discovering Transcriptional Regulatory Rules from Gene Expression and TF-DNA Binding Data by Decision Tree Learning

Authors: Ruan, Jianhua; Zhang, Weixiong

July 18, 2004

Abstract: Background: One of the most promising but challenging task in the post-genomic era is to reconstruct the transcriptional regulatory networks. The goal is to reveal, for each gene that responds to a certain biological event, which transcription factors affect its transcription, and how several transcription factors coordinate to accomplish specific regulations.
Results: Here we propose a supervised machine learning approach to address these questions. We build decision trees to associate the expression level of a gene with the transcription factor binding data of its promoter. From the decision trees, we extract regulatory rules that specify how the binding of a combination of several transcription factors affects the expression of a gene. Such rules are easy to interpret, and represent experimentally testable hypotheses. We use a decision tree ensemble approach to increase modeling accuracy and robustness. We also propose a novel method to integrate rules learned from several time series that measure the same biological processes. We apply our method to publicly available cell cycle expression data and transcription factor binding data for the budding yeast. Cross-validation experiments show that our method is highly accurate and reliable. The method correctly identifies all major known yeast cell cycle transcription factors, and assigns them into appropriate cellcycle phases. It also explicitly reveals synergetic relationships of transcription factors, most of which agree well with existing literatures, while the rest provide testable biological hypotheses

Discovering Transcriptional Regulatory Rules from Gene Expression and TF-DNA Binding Data by Decision Tree Learning

Jianhua Ruan[1]and Weixiong Zhang*[1,2]

[1]Department of Computer Science and Engineering, Washington University in St Louis, St Louis,MO 63130, USA
[2]Department of Genetics, Washington University in St Louis School of Medicine, St Louis, MO 63130, USA

Email: Jianhua Ruan - jruan@cse.wustl.edu; Weixiong Zhang*- zhang@cse.wustl.edu;

*Corresponding author

Abstract

Background: One of the most promising but challenging task in the post-genomic era is to reconstruct the transcriptional regulatory networks. The goal is to reveal, for each gene that responds to a certain biological event, which transcription factors affect its transcription, and how several transcription factors coordinate to accomplish specific regulations.

Results: Here we propose a supervised machine learning approach to address these questions. We build decision trees to associate the expression level of a gene with the transcription factor binding data of its promoter. From the decision trees, we extract regulatory rules that specify how the binding of a combination of several transcription factors affects the expression of a gene. Such rules are easy to interpret, and represent experimentally testable hypotheses. We use a decision tree ensemble approach to increase modeling accuracy and robustness. We also propose a novel method to integrate rules learned from several time series that measure the same biological processes. We apply our method to publicly available cell cycle expression data and transcription factor binding data for the budding yeast. Cross-validation experiments show that our method is highly accurate and reliable. The method correctly identifies all major known yeast cell cycle transcription factors, and assigns them into appropriate cell cycle phases. It also explicitly reveals synergetic relationships of transcription factors, most of which agree well with existing literatures, while the rest provide testable biological hypotheses.

Conclusions: The high accuracy of our method indicates that our method is valid and that the learned regulatory rules can be used as the basic building elements of a transcriptional regulatory network. As more and more gene expression and TF binding data are available, we believe that our method will be useful for reconstructing large scale transcriptional regulatory networks.

Background

Transcriptional level of gene expression is controlled, to a large extent, by specific interactions between transcription factors (TFs) and the promoter sequences of their target genes. The interactions between TFs and target genes can be many-to-many, i.e., each TF controls many genes, and a gene can be controlled by many TFs. To understand gene functions in different biological processes, it is necessary to reveal this transcriptional regulatory network.

To reveal transcriptional regulatory networks, traditional methods start by clustering genes according to similar expression patterns across multiple conditions [1–3], and then look for statistically over-represented sequence motifs from the promoter regions of genes in the same cluster [4–6]. Such enriched motifs, if identified, are often believed to be the binding motifs of a common TF. These approaches have been successful in small datasets, but are limited by theirs strong assumptions that co-expression means co-regulation and vice versa [7, 8]. Furthermore, in higher eukaryotes, genes are typically regulated by a combination of several TFs, and the TF binding motifs often organize into modular units [9]. Although some progress has been made [10–12], it is still difficult to precisely identify combinatorial motifs. Finally, it may not be easy to map the putative binding motifs to their corresponding TFs.

Another type of approaches formulates the problem in a statistical learning framework [8,13,14]. These approaches assume that each motif makes an additive contribution to the expression of a gene. Therefore, the expression level of a gene can be modeled as a linear function of scores for binding motifs. Given a set of putative binding motifs and gene expression levels, these methods use linear regression and feature selection techniques to find the most significant motifs that can explain the expression levels. These methods have been shown effective for discovering conserved short motifs related to several biological processes in *S. cerevisiae*. However, they are limited by their assumption of linear additivity of binding motifs. Furthermore, the biological meaning of a learned linear function is difficult to interpret, and may be interpreted in multiple ways. For example, if two binding motifs has the same coefficient in a linear function, it may be that the two motifs are both required, but it is also possible that one motif can replace the other.

In this paper, we propose a different approach and formulate the problem in a supervised machine learning framework. In our formulation, we assume that each gene can be in a number of different states, for example, up-regulated, down-regulated and unchanged. Given the states of a set of genes and putative regulatory elements (e.g., binding motifs) in their promoter regions, our goal is to infer a model that associates the expression states of genes with their regulatory elements. This can be considered as a classification problem in machine learning, where the genes are called instances, the regulatory elements are called features, the expression states are called classes, and the model is called a classifier. Many different classification approaches have been studied previously, such as association rules, k-nearest neighbors, support vector machines, and naive bayes [15]. Here we choose decision trees [16], which is a well-studied machine learning technique that has been successfully applied in a variety of application domains [17–19]. For this work, decision trees have several advantages compared to other classification or linear regression approaches. First, decision trees can represent complex logics in transcriptional regulation. Second, decision trees can be easily converted to rules, which are easy to interpret and are experimentally testable. Third, decision trees can handle continuous values in features. Finally, decision tree algorithms have feature selection method built in.

In our work, we also explore the use of classifier ensembles, i.e., we learn multiple decision trees for each data set. Comparing to single classifier approach, the ensemble approach generally results in more accurate and robust classifiers. Furthermore, it provides alternative models that can be studied.

To our knowledge, supervised machine learning approach, in particular decision tree approach, has not been used for learning *transcriptional* regulatory networks, although it has been used for learning regulatory networks [18–21]. In these works, supervised machine learning approaches were used to associate the expression level of a gene with the expression levels of putative TFs. Since there is no evidence of binding, the learned regulation relationships may be indirect interactions.

Besides the difference in problem formulation, our method also differ from previous approaches in that it exploits the genome-wide TF binding data [22]. TF binding data is measured by the chromatin immunoprecipitation (ChIP) DNA chip technology [23], and reflects the relative binding strength of TFs to all promoter regions in a genome. There are several advantages in using TF binding data instead of putative binding motifs. First, the number of TFs is generally much smaller than the number of putative binding motif scores. Using TF binding data therefore significantly reduces the number of irrelevant features that may distract the learning algorithms. Second, with TF binding data, our method directly associates a gene's expression with TFs that regulate it. While knowing binding motifs of TFs is still important, it can be separated from the learning of transcriptional regulatory networks. If there is no TF binding data, however, our method can still use putative binding motifs. In addition, we can also use a mixture of both TF binding data and putative binding motifs to reduce the risk of missing relevant features. Note that TF binding data can also be used by linear regression approaches [8, 13, 14], however some of these works were done before the TF binding data was available.

Microarray experiments are often done in time series (e.g., in [24]). For each time point, our approach learns an ensemble of decision trees. We then extract rules from these decision trees by following each path from the root to a leaf. Such rules are called regulatory rules, which predict for a gene its expression state given its TF binding data. We compute a significance score for each rule based on how many genes it controls. Plotting the significance score of a rule as a function of time therefore reveals the function of TFs.

In some cases, two or more time series related to the same biological process were measured by different researchers, often with different sampling rates. Here we propose a spline interpolation method to combine results from multiple time series. Such an integrated approach can substantially eliminate noises contained in each individual data source and improve modeling accuracy.

To test the validity of our approach, we applied it to three sets of yeast cell cycle gene expression data [24, 25]. We demonstrate that the method is able to identify biologically significant regulatory rules from genome-wide TF binding data and gene expression data. Statistical evaluation indicates that the rules identified are robust and reliable. Many of the transcriptional regulatory rules for yeast cell cycle genes discovered by our approach have been confirmed by published literature, while the testing of other yet-unverified rules may yield additional insights into the biological process.

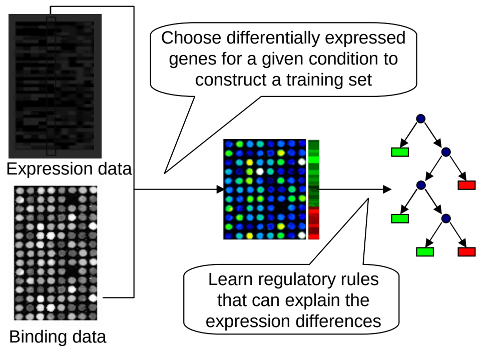
Results

Overview of the Algorithm

Our method takes as input the expression data and TF binding data of a set of genes and proceeds in two stages (Figure 1). In the first stage, we construct a training data set for each experimental condition of the expression data, and learn a set of regulatory rules. In the second stage, we generate profiles for rules, integrate results from multiple data sets, and combine rules into a transcriptional regulatory network.

A training set contains a set of genes (instances), where each gene is represented by a vector. The vector corresponding to the jth gene is defined as $< B_{1j}, B_{2j}, ..., B_{nj}, C_{kj} >$, where B_{ij} is the strength for the ith TF binding to the jth gene, and C_{kj} is the label of expression state for the jth gene under condition k. For simplicity, we considered only binary labels: "up-regulated" and "not-up-regulated", while it can be easily generalized to any number of states. In this paper, we refer to up-regulated and not-up-regulated genes as positive and negative genes, respectively. The labels are determined by fixed thresholds (see Materials and Methods). The strength of a TF binding to a promoter sequence is represented by the negative logarithm of the binding p-value. Note that the binding strength B_{ij} is measured statically and therefore does not change as k changes.

Once we have constructed the training set, we can learn a function that maps the TF binding data of a gene to its expression states. As discussed in the introduction, we choose decision trees to represent this function, because of its advantages over other techniques. A decision tree is a rooted tree consisting of two kinds of nodes: internal nodes and leaf nodes. Each internal node corresponds to a test of the binding of a selected TF to a gene (for example, "can TF A bind to gene g?"), and each leaf is a prediction of the state of that gene (for example, "gene g is up-regulated"). Each internal node has two branches: the right branch is chosen when the test succeeds; and the left branch is chosen when it fails. Therefore, a path from the root to a leaf defines a possible regulatory rule (for example, " if a gene can be bound by TF A and TF B, then it can be up-regulated at time


Expression data
Binding data
Choose differentially expressed genes for a given condition to construct a training set
Learn regulatory rules that can explain the expression differences
(a)

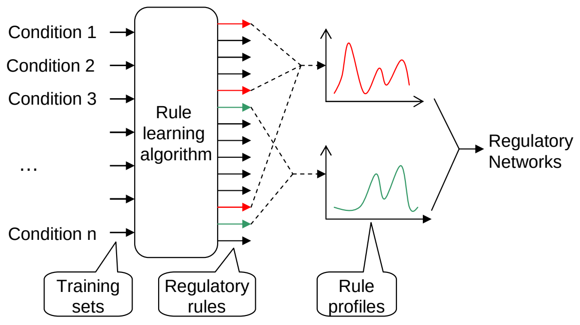


Figure 1: Overview of our approach. a), Constructing the training set and learning a set of regulatory rules for each condition. b), Generating rule profiles and combining them into transcriptional regulatory networks.

t"). We extract regulatory rules from decision trees, and calculate a significance score (p-value) for each rule (see Materials and Methods). Only significant rules (p-values < 0.001) are retained. This provides a single-experiment view of regulatory rules.

A regulatory rule may very often be discovered at multiple conditions (or time points). The negative logarithm of the p-value of a rule under a given condition reflects the significance of the rule. Thus it is informative to plot $-log(p)$ as a function of experimental conditions; such a plot is referred to as a rule profile. This provides a single-rule view across multiple conditions. When two or more microarray time series are available for the same biological process, we can also combine the rule profiles learned from different time series. We approximate each rule profile with a spline interpolation, and add together profiles for the same rule from different time series to give a single profile.

In the last step, we identify the most probable experimental conditions for each rule and the genes it regulates, and organize this information into a transcriptional regulatory network.

Decision trees and regulatory rules relevant to the yeast cell cycle

Learning decision trees and regulatory rules in our method is best illustrated by the yeast cell cycle data. It is known that nine TFs - Mbp1, Swi4, Swi6, Mcm1, Fkh1, Fkh2, Ndd1, Swi5 and Ace2 - regulate a large number of yeast cell cycle dependent genes [26, 27]. Specifically, MBF (a complex of Mbp1 and Swi6) and SBF (a complex of Swi4 and Swi6) control late G1 genes; Mcm1, together with Fkh1 or Fkh2, recruits Ndd1 in late G2 and controls the transcription of G2/M genes; and Swi5 and Ace2 regulate genes at the end of M and early G1. This model was developed using a small set of genes and was recently confirmed by a number of whole-genome analysis combining gene expression and TF binding data [22, 28]. We thus applied our method to the cell cycle data to verify the accuracy of our method. We also aimed to construct a more detailed transcriptional regulation network as well as new, testable hypotheses about yeast cell-cycle regulations.

Gene expression during the yeast cell cycles has been measured with several different synchronization methods. We apply our method to three data sets obtained from the methods of CDC28, CDC15 and α-factor [24, 25], while our discussion below mostly focus on the rules learned from the CDC28 data set. Later we also propose a novel method to combine the results from all three data sets.

We first learned only a decision tree for each time point. Figure 2 shows the decision trees learned from the 20, 40, 70 and 100 minute CDC28 data, corresponding to late G1, S, G2/M and early G1 phases, respectively. The method rediscovered all nine known TFs in appropriate cell cycle phases. As can be seen, Swi4, Swi6 and Mbp1 appeared in 20 and 100 minute. Ndd1, Mcm1, Fkh1 and Fkh2 appeared in 40 and 70 minute. Swi5 and Ace2 appeared in 100 minute.

We then extracted regulatory rules from the trees by a depth-first search from the root node to all leaf nodes labeled as positive. A node was included in a rule only if its right branch was taken by the path. For example, we extracted the following two rules from the 70-minute tree (Figure 2c): (Ndd1 ≥ 2.47) $\cap$ (Mcm1 ≥ 3.82), and (Ndd1 ≥ 2.47) $\cap$ (Fkh1 ≥ 3.44). According to the first rule, genes that can be bound by Ndd1 with a p-value less than $e^{-2.47}$ and by Mcm1 with a p-value less than $e^{-3.82}$ will be up-regulated at 70 minute. For simplicity, we will omit the p-value thresholds of binding data in later discussions, and write the two rules as Ndd1 $\cap$ Mcm1 and Ndd1 $\cap$ Fkh1, respectively. It is worth mentioning, however, that the thresholds are learned automatically and may be different in different rules.

Each rule has some number of supporting genes in the training set, from which a p-value can be calculated. For example, the rule Ndd1 $\cap$ Mcm1 in the 70-minute tree is supported by 18 positive and 1 negative genes out of a total of 41 positive and 416 negative genes. This corresponds to a p-value $\approx 10^{-20}$. (For the detail of calculating the p-value of a rule, see Materials and Methods).

The strongest rule identified for 20-minute time point is Mbp1 $\cap$ Swi6 ($p = 10^{-19}$). The other three significant rules are Swi4 $\cap$ Swi6 ($p = 10^{-9}$), Mbp1 $\cap$ Dot6 ($p = 10^{-5}$) and Mbp1 $\cap$ Ash1 ($p = 10^{-5}$) (Figure 2a). Ash1 was known to accumulate in the daughter cell throughout the G1 phase, inhibiting transcription of the HO endonuclease, thereby preventing mating-type switching [29]. Dot6 has been shown to affect pseudohyphal differentiation [30]. Genes up-regulated at 40 minute are described by three significant rules: Swi4 ($p = 10^{-17}$), Fkh1 $\cap$ Fkh2 ($p = 10^{-6}$), and Met4 $\cap$ Met31 ($p = 10^{-4}$) (Figure 2b). Met4 and Met31 cooperate to regulate the sulfur amino acid pathway [31]. A cluster of genes involved in the biosynthesis of methionine have been previously discovered as cell cycle regulated [24]. Two significant rules were identified for 70-minute time point (Figure 2c), both are well known: Ndd1 $\cap$ Mcm1 ($p = 10^{-20}$) and Ndd1 $\cap$ Fkh1 ($p = 10^{-6}$). Rules identified for 100-minute time point include early G1 phase TFs, Swi5 $\cap$ Ace2 ($p = 10^{-5}$), as well as late G1 phase TFs Mbp1 ($p = 10^{-20}$) and Swi4 ($p = 10^{-8}$).

Ensemble decision trees and regulatory rules learned from all data sets

The above example illustrated the ability of the single decision tree approach in identifying the known TFs and associating them with appropriate cell cycle phases. However, like all methods providing optimal solutions, it may miss suboptimal but meaningful solutions. Even worse, the standard decision tree learning algorithm (for instance, C4.5) is essentially greedy, since the tree structure is learned top-down without backtracking. Therefore, there may be other trees that can explain the training data equally well or even better than the tree reported by the algorithm. A common approach for solving this problem is by learning a set of decision trees for each training data (often referred to as a tree ensemble in machining learning).

Many machine learning approaches have been developed for learning tree ensembles (for review, see [32]), including Bagging [33] and Boosting [34]. One basic idea in these methods is to perturb the original data set many times, and learn a decision tree from each derived data set. Each decision tree stands for an alternative model. To make a prediction, an instance is passed to each individual decision tree and predictions are combined by voting [32]. We adopt the basic idea, but also consider a unique feature of our data set: the number of negative instances is much larger than

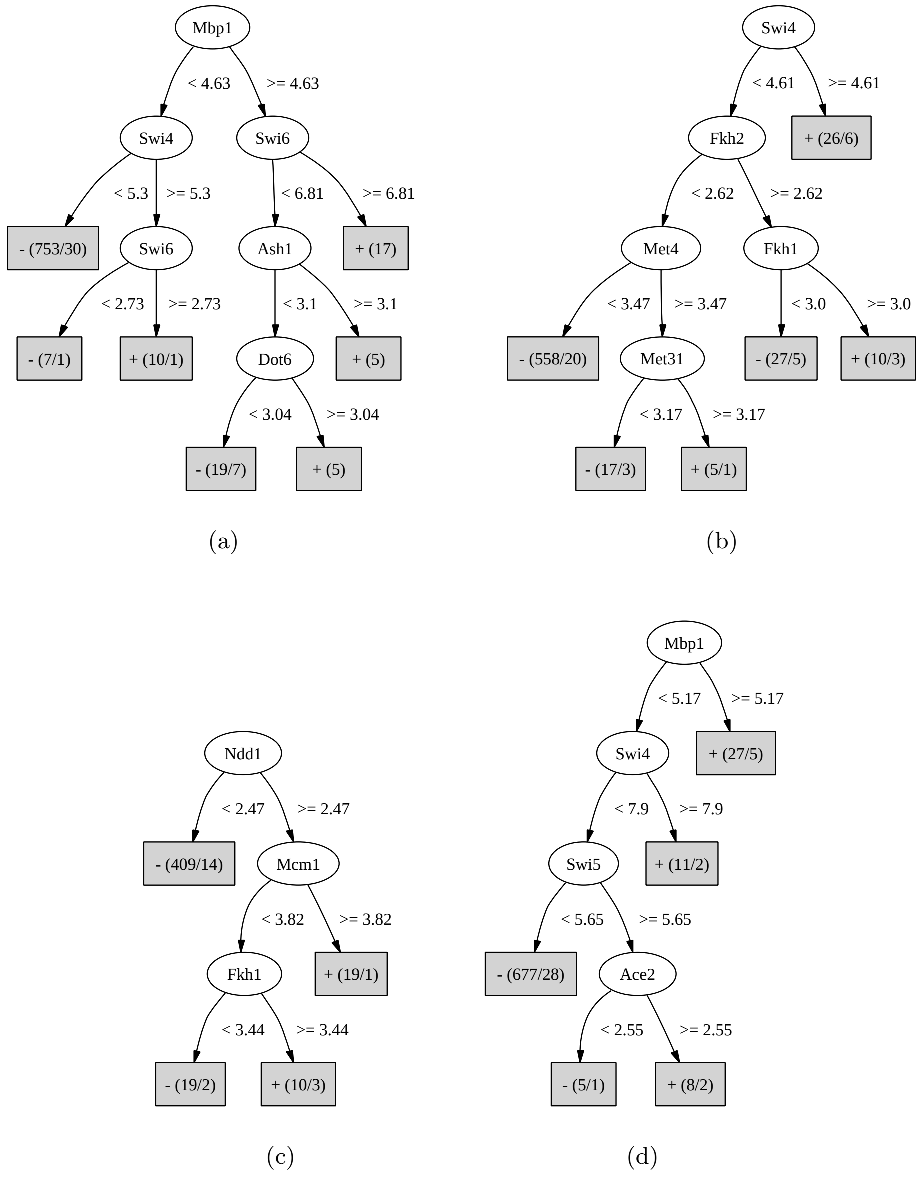


Figure 2: Example decision trees learned from the CDC28 cell cycle data set at four different time points. a), 20 minute. b), 40 minute. c), 70 minute. d), 100 minute. Each oval represents an internal node and each box represents a leaf node. The text inside an internal nodes is a regulator, while the text associated with an edge is a test on a DNA binding p-value. The text inside a leaf node is a prediction of the state of a gene. The number of supporting genes and the number of counter examples are included in parentheses. For example, "+(19/1)" in the 40-minute tree means that the rule will predict positive and there are 19 genes satisfy this rule, of which 18 are true positive and the remaining one is false positive.

Table 1: Rules learned by the splitting approach

20min	40min	70min	100min
Mbp1 (10^{-31})	**Swi4** (10^{-17})	**Mcm1** $\cap$ **Ndd1** (10^{-25})	**Mbp1** (10^{-20})
Mbp1 $\cap$ **Swi6** (10^{-26})	Mth1 $\cap$ Swi4 (10^{-11})	Fkh2 (10^{-21})	**Swi4** (10^{-18})
Stb1 $\cap$ Swi4 (10^{-15})	Fkh2 (10^{-10})	Ndd1 (10^{-17})	Swi4 $\cap$ Swi6 (10^{-14})
Swi4 $\cap$ **Swi6** (10^{-9})	**Fkh1** $\cap$ **Fkh2** (10^{-6})	Fkh2 $\cap$ Ndd1 (10^{-15})	Ste12 $\cap$ Swi4 (10^{-8})
Swi4 (10^{-7})	Met4 (10^{-5})	**Fkh1** $\cap$ **Ndd1** (10^{-6})	Hir2 $\cap$ Swi4 (10^{-5})
Mbp1 $\cap$ Swi4 (10^{-6})	Fkh2 $\cap$ Msn1 (10^{-5})	Fkh1 $\cap$ Fkh2 (10^{-4})	Mbp1 $\cap$ Mss11 (10^{-5})
Dot6 $\cap$ **Mbp1** (10^{-5})	Hsf1 (10^{-5})	Mcm1 (10^{-4})	**Ace2** $\cap$ **Swi5** (10^{-5})
Ash1 $\cap$ **Mbp1** (10^{-5})	**Met4** $\cap$ **Met31** (10^{-5})	Nrg1 $\cap$ Smp1 (10^{-3})	Mbp1 $\cap$ Stb1 (10^{-5})
Ecm22 $\cap$ Mbp1 (10^{-3})	Met4 $\cap$ Cbf1 (10^{-5})		Swi5 (10^{-4})

the number of positive ones. Such a skewed class distribution deteriorates the learning ability of most machine learning algorithms [35], including decision trees. To overcome this difficulty, we split negative instances into smaller subsets and combine each of them with positive instances to form a training set, from which a decision tree is learned (see Materials and Methods). We refer to this method as splitting. By this approach, we effectively adjust the class distribution to a preferred value without losing any information in the original data set. The prominent regulatory rules will likely be present in many trees and stand out when the trees are combined. An idea similar to the splitting approach was proposed to learn decision trees for detecting credit-card frauds [36].

Table 1 shows a selected list of significant rules discovered by the splitting approach when applying to the 20, 40, 70 and 100 minute CDC28 data set. A complete list is included in supplementary table 1. As can be seen, the splitting approach discovered additional synergetic relationships among the known cell cycle TFs, such as Mbp1 $\cap$ Swi4 and Fkh2 $\cap$ Ndd1, which were not identified by the single tree approach. Furthermore, several rules involving cell-cycle related TFs were discovered. For example, Stb1 and Ecm22 were found in 20 minute, Cbf1, Hsf1, Rgm1 and Mth1 in 40 minute, Nrg1 and Smp1 in 70 minute, Ste12, Hir2 and Mss11 in 100 minute. Among them, Stb1 is known to regulate in G1 [37]; Cbf1 binds to centromere and is involved in DNA replication and methionine biosynthesis together with Met4 [31, 38]; Nrg1 and Smp1 were recently found to regulate filamentous growth [39].

We repeated the learning method on the CDC15 and α-factor data sets, and the resulting regulatory rules are listed in Supplementary table 2 and 3, respectively. Not unexpected, most of the significant rules involve at least one of the nine well-known TFs. Two significant rules identified in the α-factor data set involve novel transcription factors: Yap5 ($p = 10^{-10}$ at 14-minute and 10^{-8} at 77-minute) and Gat3 ($p = 10^{-9}$ at 14-minute and 10^{-8} at 77-minute). The role of the two TFs in G1 is still unknown and may deserve further investigation. Later, we will introduce a method for combining the rules learned from the three data sets.

Evaluating the reliability of rules

A critical issue of classification algorithms is generalization - how well a learned model can be applied to data that have not been seen by the learning algorithm? When the number of features is large, a classifier is often over-fitted. In another word, it can achieve very good performance on the data used to learn the classifier, while performs poorly on unseen data. Therefore, it is important to evaluate the accuracy of a classifier on unseen data, which is typically done by a cross-validation procedure (see Materials and Methods). A straightforward measurement of accuracy is the percentage of correctly classified instances (denoted as A). However, A tends to under-estimate the true error, especially when the ratio of positive and negative instances is skewed. For example, if there are 990 negative and 10 positive instances, simply predicting everything as negative will achieve 99% accuracy. Therefore, we compute the kappa statistic K to measure accuracy. K is a better estimation of the true classification accuracy, and is guaranteed to be no greater than A (See Materials and Methods). Furthermore, it has been suggested that $K < 0.4$ indicates a poor classifier, $K > 0.75$ implies an excellent classifier, and $0.4 < K < 0.75$ means a reasonably good classifier [40].

Figure 3 shows the cross-validation kappa statistics of the single decision tree approach and three ensemble approaches (bagging, boosting and splitting) on eight time points of the CDC28 data set. The splitting method has the best K under almost all conditions, with a value at least 0.4 in essentially all time points. Furthermore, when we randomized the training set by randomly exchanging positive and negative labels, the same splitting method yield kappa statistics smaller than 0.02 in all cases (average = -0.002). This confirms that the rules learned are not random.

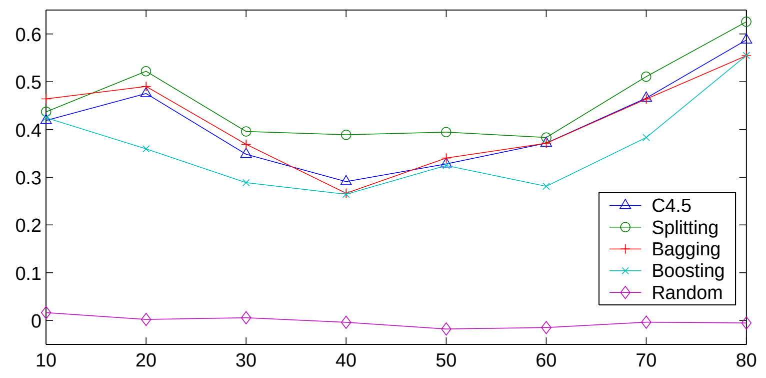
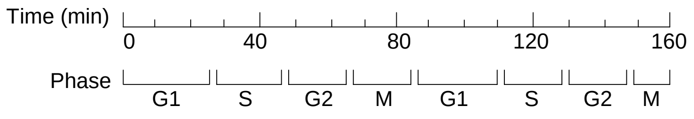


Figure 3: 10-fold Cross-validation accuracy of C4.5 [41], Bagging [33], Boosting [34] and Splitting [36]. Experiments were done on eight different time points of CDC28 data set. Implementation of Bagging, Boosting and C4.5 were obtained from the WEKA package [42]. C4.5 was also used as the base level classifier for Bagging, Boosting and Splitting. Default parameters were used for C4.5, Bagging and Boosting. Splitting were done according to Materials and Methods.

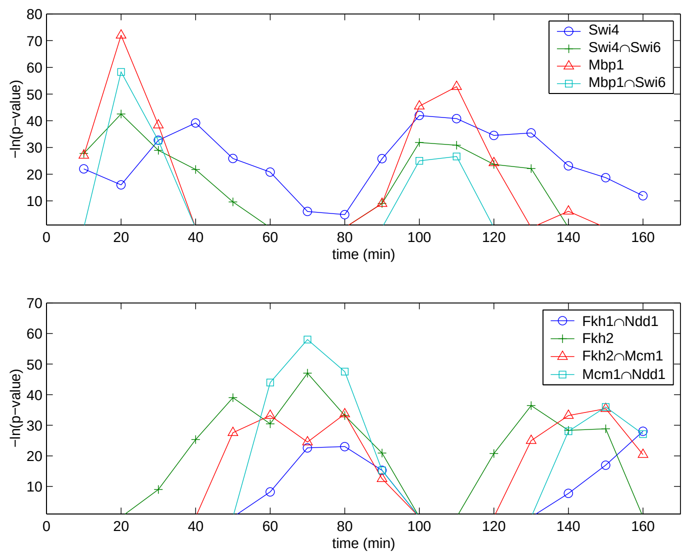
Integrating rule profiles

The negative logarithm of the p-value of a rule under a give condition reflects the significance of the rule. We obtained the profile of each rule by plotting its $-log(p)$ as a function of time. Such a plot can be used for several purposes. First, the wave form shows the change of significance score of a regulatory rule over time. Therefore it reveals the most probable period of time during which the rule regulates. Second, the pattern of rule profiles in a time series reveal certain properties of the biological process (for example, critical time point for a phase transition or length of a cell cycle). Third, comparing the profile of a rule with the expression pattern of the corresponding TFs indicates the direction of the regulation (See Discussion).

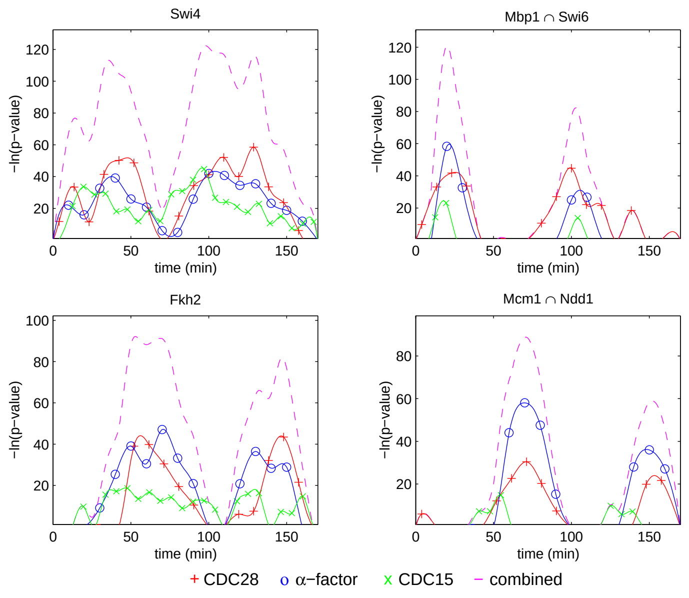
Figure 4b illustrates rule profiles of G1 and G2/M TFs Mbp1, Swi4, Swi6, Ndd1, Mcm1, Fkh1 and Fkh2 obtained from the CDC28 data set. These profiles all showed clear periodicity. Their peaks agree very well to cell cycle phases determined by phenotypes and gene expression data (Figure 4a) [25]: Swi4, Swi6 and Mbp1 peak in G1, and Ndd1, Mcm1, Fkh1 and Fkh2 peak in G2/M. The rule profiles also show that there is a significant lag between the peaks of Mbp1 and Swi4, which was also discovered by previous studies [13, 22, 24]. We also found a lag between the peaks of Fkh2 and Mcm1, which is different from a antagonistic (out-of-phase) relationship suggested by Bussemaker et al. [13], but similar to the results reported by Lee et al. [22]. Our results also show a significant lag between Fkh2 and Fkh1, similar to what was reported previously [22].



(a)



(b)



(c)

Figure 4: Profiles of selected rules. a), Approximate cell cycle phases in CDC28 data set. b), Rule profiles obtained from CDC28 data set alone. b), Integrated rule profiles obtained from CDC28, CDC15 and α-factor data sets.

Since all three data sets, CDC15, CDC28, and α-factor, measured gene expression levels during yeast cell cycle, the gene expression patterns in them should be similar; so should the inferred profiles of regulatory rules. Therefore, it should be possible to combine the rule profiles learned from them. However, the length of a cell cycle and the sampling rates are different in these three data sets, which makes a direct point-to-point addition invalid. Previous studies have shown that it is possible to convert the time scales of the CDC15 and α-factor data sets to the time scale in CDC28 [43]. They found that, after conversion, expression curves in the three data sets can be aligned together very well. We used the same conversion and took the parameters from their results. As we expected, the rule profiles from different data sets can often be aligned together accurately (Figure 4c). We then used spline interpolation in MATLAB (the MathWorks Inc.) to convert rule profiles to continuous curves, which are then added together to obtain a combined profile for each rule. Figure 4c shows the integrated profiles of several rules. As shown, the integrated profiles show prominent cell cycle dependencies (period $\approx$ 85 minutes). Supplementary Figure 2 contains integrated rule profiles with notable cell cycle dependencies, and Supplementary Figure 3 shows integrated rule profiles that do not show clear cell cycle dependencies.

A model for the yeast cell cycle transcriptional regulatory network

From the cell cycle dependent rule profiles in Supplementary Figure 2, we constructed a model of yeast cell cycle transcriptional regulatory network (Figure 5). We first determined for each rule the most probable period of time during which the rule functions, and plotted the rule in the corresponding phase of the cell cycle. We then determined the genes that each rule regulates, and created a link from the rule to a gene if the gene also appears in a regulatory rule (see Materials and Methods). We grouped most rules into two large modules (gray area), where the rules in each module share a lot of common target genes. One module is in G1/S and has Mbp1, Swi4, Swi6 and Stb1 in the rules. The other module is in G2/M and involves Fkh1, Fkh2, Ndd1 and Mcm1.

We found that the rules functioning in one phase of the cell cycle regulate TFs functioning in the next phase (red lines in Figure 5). This result is consistent with previous studies [22,28], although we identified more such relations. We also found that, within each phase, rules that function earlier often regulate TFs that function later (blue lines in Figure 5). For example, we found that the earliest TF in G2, Fkh2, regulates Ndd1 and Fkh1. As to our knowledge, this result has not been reported previously.

In addition, two rules combining G1 and G2 TFs (Fkh2 $\cap$ Swi4 and Fkh2 $\cap$ Swi6) function in S phase and regulate Ndd1. Another such combination, Fkh1 $\cap$ Mbp1, functions in M phase. We also identified several novel TFs for yeast cell cycle: Dot6, Yap5 and Gat3 in G1, and Met4 in S. Yap5 and Gat3 may be suspicious since the rules were only learned in the α-factor data sets, although their profiles show very clear cell-cycle dependencies. Gat3 was found to be regulated by Swi5 in our network.

Discussion

Reconstructing gene regulatory networks from gene expression data is a promising but challenging task for the post-genomic era. Traditional methods use a two approach. The first phase groups genes into clusters according their expression similarities [1–3]. The second phase scan for single or composite motifs that are enriched in the promoter regions of clustered genes [4–6, 10–12]. These methods, however, are limited by their over-reliance on expression similarities. Furthermore, computational motif finding is a difficult task, while the mapping from binding motifs to corresponding TFs is even harder. Statistical learning method consider individual expression experiment separately, and fit a linear model to describe the additive effect of motifs on the expression levels of individual genes [8, 13, 14]. These methods did not, however, explicitly take combinatorial effects into account.

In this paper we proposed a supervised machine learning approach to discover transcriptional regulatory rules from gene expression data and TF-DNA binding data. We used decision trees to model the relationship between the expression level of a gene at particular time points and the TFs that can bind to it, and extracted easy-to-interpret regulatory rules from decision trees. We applied an ensemble learning approach to explore alternative models and increase the modeling accuracy. We also proposed a spline interpolation approach for integrating the regulatory rules learned from multiple time series expression data.

Using the cell cycle data sets as examples, we demonstrated that our method is able to identify biologically significant regulatory rules from genome-wide TF binding data and gene expression

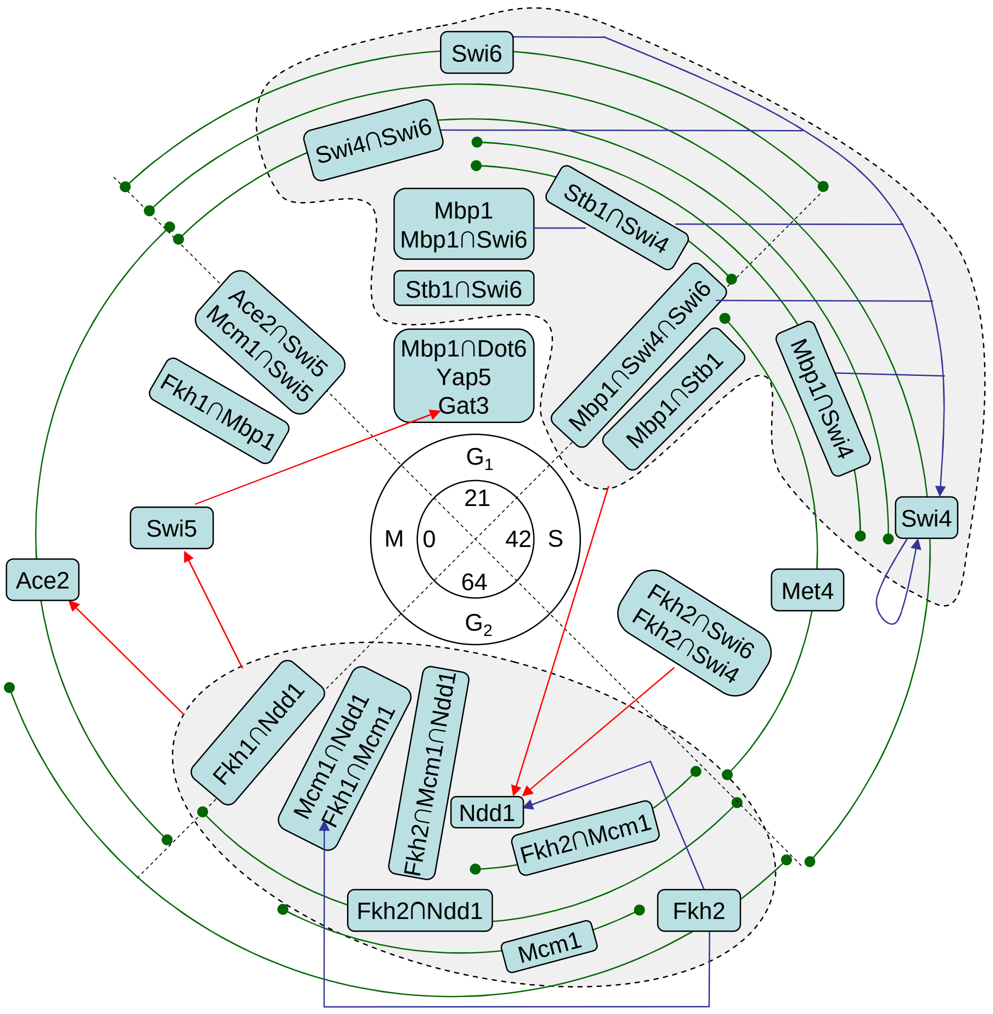


Figure 5: A model for the yeast cell cycle transcriptional regulatory network learned by our method. The text inside each rounded rectangle represents one or more regulatory rules. The position of a box, together with the green arc crossing it if exists, represents the period during which the rules inside are functioning. Two long dashed lines divide the area into G1, S, G2 and M phases approximately. The numbers in the inner circle represent the corresponding middle time point of each phase in the CDC28 data set. The gray area on the top contains the module of rules that regulate late G1/S phase, and the one on the bottom encloses the module of rules that regulate G2/M phase. A red line represents that a set of regulatory rules regulates a regulator outside the module, while a blue line represents that a set of regulatory rules regulates a regulator within the same module.

data. The process of deriving all predictions in our method was unbiased by any computational or experimental knowledge. Without pre-clustering genes based on global similarity of expression patterns, we re-discovered all nine known TFs that are relevant to the yeast cell cycle and assigned them into appropriate cell cycle phases. Most regulatory rules in our results involve two or three TFs, suggesting synergetic relationships for them. For example, we have identified the collaboration of many well known TF pairs, such as Mbp1/Swi6, Swi4/Swi6, Stb1/Swi6, Fkh1/Mcm1, Fkh1/Ndd1, Fkh2/Ndd1, Ace2/Swi5 and Met4/Met31, as well as the recently reported Met4/Cbf1 and Nrg1/Smp1 complexes. The test of other yet unverified rules may yield additional insights to the biological process.

Our method has some limitations. Although statistically significant rules often reflect biological significance, the opposite is not always true. As a result, our method may miss regulatory rules that regulate only a few genes. For example, our method failed to discover Skn7, a TF functioning in S phase, since the number of genes regulated by Skn7 is small in the given data sets to be considered statistically significant. However, this limitation is probably common to most large-scale analysis methods.

Another limitation of our method is that regulatory rules do not specify whether a participating TF contributes inductively or repressively. This is because concentrations of TF proteins are not taken into account. For example, if a rule states that "if gene g can be bound by TF f, then it can be up-regulated at time t", it is possible that g is up-regulated at t due to a reduced concentration of f, which actually implies a repressive role of f. This ambiguity may be resolved by comparing rule profiles with expression patterns of TFs. For example, the rule profile of Swi4 reaches its peak at 40 minute, while expression of Swi4 peaks at about the same time. This suggests that Swi4 is a transcriptional activator. However, the correlation does not always hold, since there may be a lag of time between the expression of a TF and its functioning, and many TFs may be modified post-transcriptionally. For example, the mRNA level of Mbp1 is almost constant during the cell cycle, although its rule profile peaks at 20 minute. We note that the same limitation exists for linear regression approaches [8, 13, 14].

It is also worth noting that there are alternative ways to label genes with expression states. Here we labeled a gene according to its expression level under a single condition relative to an initial condition. Alternatively, we may label a gene according to its expression level relative to the previous time point, or relative to its mean expression level in a time series. It may also be advisable to consider several consecutive time points simultaneously. We have tested some of these ideas, and the conclusion is that all these labeling methods are valid to a certain extent (in terms of cross-validation accuracy), and there is no single method that is the best for all data sets. The labeling method we chose has the best cross-validation accuracy in average. The decision trees learned with different labeling methods are often different. Nevertheless, when the ensemble approach is used, the most significant regulatory rules tends to be stable with respect to labeling methods.

We also tested our method in other data sets that are not time series or do not have TF binding data. For example, we applied the method to the glucose-limited growth of yeast cells [44], and correctly identified Hap4 as a significant TF that regulates the process. For many species, especially in higher eukaryotes, we do not have TF binding data available. To study transcriptional regulations for them, we can utilize information of putative binding motifs. For example, we applied our method to study the transcriptional regulation of stress-responsive genes in Arabidopsis [45], using putative binding motifs from a plant motif database (PLACE) [46]. We successfully identified two motifs (ABRE and CE) that are known to regulate stress-responses in plants. Using putative binding motifs, however, involves some modification of the decision tree learning algorithm, since there are more features and some of them may overlap with each other. The details of these results will be published elsewhere.

Conclusions

We have proposed a decision tree approach for discovering transcriptional regulatory rules. By integrating multiple heterogeneous data sources, we are able to achieve high modeling accuracy. Statistical evaluation and literature validation indicate that the results are robust and reliable. We have also shown that the regulatory rules can be used as the basic building elements of a transcriptional regulatory network. As more and more gene expression data and TF binding data become available, we believe that our method will be useful for reconstructing large-scale transcriptional regulatory networks.

Materials and methods

Gene expression and TF binding data. We used *S. cerevisiae* cell-cycle data synchronized with CDC28 [25], CDC15 [24] and α-factor [24]. For CDC28 data set, we used a 3-fold induction as the threshold for selecting positive genes. That is, a gene is positive at time point t if $E_t/E_0 \geq 3$, where E_t is its expression level at time t and E_0 is its expression level at the starting point of the time series. To have a clear separation of positive and negative genes, we chose a gene as negative only if $E_t/E_0 \leq 1.2$. Since expression levels in CDC15 and α-factor are normalized by a log_2 ratio, we chose positive genes so that $E_t - E_0 \geq log_2 3$ and negative $E_t - E_0 \leq log_2 1.2$. Furthermore, in all three data sets, we required the expression levels of positive genes and negative genes to be greater than and less than their average expression values, respectively. We used genome-wide binding data of 113 *S. cerevisiae* TFs from Lee et al [22]. We used a less stringent threshold ($p < 0.1$) than the suggested threshold ($p < 0.001$) to reduce false negatives, and depended on the learning algorithm to automatically determine an optimal threshold for each TF.

Learning decision trees and tree ensembles. We modified a standard algorithm C4.5 for learning decision trees [41]. The implementation of the algorithm was from the WEKA machine learning package [42]. To learn tree ensembles, we first separate a training set into positive gene set and negative gene set. Instances in the negative set were randomly partitioned into n subsets, where n is chosen so that the size of a negative set is $3 - 4$ times the size of the positive set. This is then repeated 5 times with different random seeds, giving a total of $5n$ negative sets. We combined each negative set with the positive set to learn a decision tree. We also learn a decision tree on the complete original training set. To make predictions, the prediction from individual decision tree is combined by a weighted voting, where the weight is the probability that an instance is predicted as positive.

Cross-validation. A 10-fold cross-validation was used to estimate the accuracy of our method. In other words, we randomly partitioned the training data into 10 subsets of equal size, and then combined 9 subsets for training and the remaining one for testing. The process was repeated 10 times so that each subset was used as a test set once. Furthermore, we repeated the cross-validation procedure 10 times with different random partitioning and calculated the average performance. Denote TP, TN, FP, and FN as the numbers of true positive, true negative, false positive and false negative predictions, respectively. The overall accuracy $A = (TP+TN)/(TP+FP+TN+FN)$ tends to under-estimate the true error when the class distribution is skewed. We thus calculated the kappa static [40], which is defined as

$$K = (A - C)/(1 - C) \tag{1}$$

where C is the expected accuracy that a classifier can achieve by chance and is calculated as

$$C = \frac{(TP + FP)(TP + FN) + (TN + FN)(TN + FP)}{(TP + FP + TN + FN)^2} \tag{2}$$

Extracting significant regulation rules. For each learned decision tree, we extracted rules by following the branches from the root node to leaf nodes labeled as positive. A node was included in a rule only if its right branch was taken to reach the leaf node of the rule. We calculated a p-value for each rule with a hypergeometric distribution, and we considered a rule to be significant if its p-value is smaller than 10^{-3}. If there are totally M positive genes and N negative genes, and a rule is supported by m positive and n negative genes ($m > n$), we calculate the p-value for the rule as the probability that we would select at least m positive genes if we randomly pick $m + n$ gene. This can be calculated as:

$$P(m, n, M, N) = \sum_{m \le x \le \min(m+n, M)} \frac{\begin{pmatrix} M \\ x \end{pmatrix} \begin{pmatrix} N \\ m+n-x \end{pmatrix}}{\begin{pmatrix} M+N \\ m+n \end{pmatrix}} \tag{3}$$

Combining rule profiles. We converted the time scale for the three expression data sets to a common scale. We used a linear function $T(s) = a * s + b$ for the conversion, where s is the actual time in an experiment and $T(s)$ is its converted time. The coefficients a, b were obtained from [43]. Using the cell cycle length of CDC28 as a reference, the coefficients are $a = 0.70$ and $b = -1.58$ for CDC15, and $a = 1.37$ and $b = 5.71$ for α-factor, meaning that the length of a cell cycle in CDC28 is 0.70-fold of the cell cycle length in CDC15 and 1.37-fold of that in α-factor, and the cell cycle in CDC28 starts 1.58 minutes earlier than in CDC15. We then approximated each rule profile with piecewise polynomial functions using the spline function in the Matlab package. An integrated profile was obtained for each rule by summing its three splines from CDC28, CDC15 and α-factor experiments. A rule was considered cell cycle dependent if its integrated profile has two peaks and the distance between the two peaks are approximately 80 - 100 minutes.

Constructing regulatory networks. The rules with notable cell cycle dependency (in Supplementary Figure 2) were used to construct a regulatory network for the yeast cell cycle. By calculating the average distance between two peaks of all the profiles, we estimated the length of a cell cycle to be 85 minutes with CDC28 data set as reference. The period that each rule functions was determined by finding the time points left and right to the peak where the y axis values were two thirds that of the peak. We then plotted the rules in their corresponding functioning phases. Next, a subset of the training data with only the genes that are part of some rules in the network were constructed and passed to the decision tree ensembles. If a gene is predicted to be positive, the rules used for the prediction were extracted, and links were created between the rules and the gene.

Acknowledgements

This research was supported in part by NSF grants IIS-0196057 and EIA-0113618 under the ITR program.

References

1. Eisen M, Spellman P, Brown P, Botstein D: **Cluster analysis and display of genome-wide expression patterns.** *Proc. Natl. Acad. Sci. USA* 1998, **95**:14863–8.
2. Tavazoie S, Hughes J, Campbell M, Cho R, Church G: **Systematic determination of genetic network architecture.** *Nat. Genet.* 1999, **22**:281–5.
3. Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander E, Golub T: **Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation.** *Proc. Natl. Acad. Sci. USA* 1999, **96**:2907–12.
4. Bailey T, Elkan C: **Fitting a mixture model by expectation maximization to discover motifs in biopolymers.** *Proc Int Conf Intell Syst Mol Biol* 1994, **2**:28–36.
5. Lawrence C, Altschul S, Boguski M, Liu J, Neuwald A, Wootton J: **Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment.** *Science* 1993, **262**:208–14.
6. Roth F, Hughes J, Estep P, Church G: **Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation.** *Nat Biotechnol.* 1998, **16**:939–45.
7. Qian J, Dolled-Filhart M, Lin J, Yu H, Gerstein M: **Beyond synexpression relationships: local clustering of time-shifted and inverted gene expression profiles identifies new, biologically relevant interactions.** *J. Mol. Bio.* 2001, **314**:1053–66.
8. Keles S, van der Laan M, Eisen M: **Identification of regulatory elements using a feature selection method.** *Bioinformatics* 2002, **18**:1167–75.
9. Yuh C, Bolouri H, Davidson E: **Genomic cis-regulatory logic: experimental and computational analysis of a sea urchin gene.** *Science* 1998, **279**:1896–902.

10. GuhaThakurta D, Stormo G: **Identifying target sites for cooperatively binding factors.** *Bioinformatics* 2001, **17**:608–21.

11. Pilpel Y, Sudarsanam P, Church G: **Identifying regulatory networks by combinatorial analysis of promoter elements.** *Nat. Genet.* 2001, **29**:153–9.

12. Segal E, Yelensky R, Koller D: **Genome-wide discovery of transcriptional modules from DNA sequence and gene expression.** *Bioinformatics* 2003, **19 Suppl 1**:i273–82.

13. Bussemaker H, Li H, Siggia E: **Regulatory element detection using correlation with expression.** *Nat. Genet.* 2001, **27**:167–71.

14. Conlon E, Liu X, Lieb J, Liu J: **Integrating regulatory motif discovery and genome-wide expression analysis.** *Proc. Natl. Acad. Sci. USA* 2003, **100**:3339–44.

15. Hand DJ, Mannila H, Smyth P: *Principles of Data Mining (Adaptive Computation and Machine Learning).* MIT Press 2001.

16. Murthy SK: **Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey.** *Data Mining and Knowledge Discovery* 1998, **2**(4):345–389.

17. Li J, Liu H, Ng S, Wong L: **Discovery of significant rules for classifying cancer diagnosis data.** *Bioinformatics* 2003, **19 Suppl 2**:II93–II102.

18. Soinov L, Krestyaninova M, Brazma A: **Towards reconstruction of gene networks from expression data by supervised learning.** *Genome Biol.* 2003, **4**:R6.

19. Segal E, Shapira M, Regev A, Pe'er D, Botstein D, Koller D, Friedman N: **Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data.** *Nat. Genet.* 2003, **34**:166–76.

20. Soinov L: **Supervised classification for gene network reconstruction.** *Biochem. Soc. Trans.* 2003, **31**:1497–1502.

21. Qian J, Lin J, Luscombe N, Yu H, Gerstein M: **Prediction of regulatory networks: genome-wide identification of transcription factor targets from gene expression data.** *Bioinformatics* 2003, **19**:1917–26.

22. Lee T, Rinaldi N, Robert F, Odom D, Bar-Joseph Z, Gerber G, Hannett N, Harbison C, Thompson C, Simon I, Zeitlinger J, Jennings E, Murray H, Gordon D, Ren B, Wyrick J, Tagne J, Volkert T, Fraenkel E, Gifford D, Young R: **Transcriptional regulatory networks in Saccharomyces cerevisiae.** *Science* 2002, **298**:799–804.

23. Ren B, Robert F, Wyrick J, Aparicio O, Jennings E, Simon I, Zeitlinger J, Schreiber J, Hannett N, Kanin E, Volkert T, CJ W, Bell S, Young R: **Genome-wide location and function of DNA binding proteins.** *Science* 2000, **290**:2306–9.

24. Spellman P, Sherlock G, Zhang M, Iyer V, Anders K, Eisen M, Brown P, Botstein D, Futcher B: **Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization.** *Mol. Biol. Cell* 1998, **9**:3273–97.

25. Cho R, Campbell M, Winzeler E, Steinmetz L, Conway A, Wodicka L, Wolfsberg T, Gabrielian A, Landsman D, Lockhart D, Davis R: **A genome-wide transcriptional analysis of the mitotic cell cycle.** *Mol. Cell* 1998, **2**:65–73.

26. Breeden L: **Cyclin transcription: Timing is everything.** *Curr. Biol.* 2000, **10**:R586–8.

27. Futcher B: **Transcriptional regulatory networks and the yeast cell cycle.** *Curr. Opin. Cell Biol.* 2002, **14**:676–83.

28. Simon I, Barnett J, Hannett N, Harbison C, Rinaldi N, Volkert T, Wyrick J, Zeitlinger J, Gifford D, Jaakkola T, Young R: **Serial regulation of transcriptional regulators in the yeast cell cycle.** *Cell* 2001, **106**:697–708.

29. Amon A: **Controlling cell cycle and cell fate: common strategies in prokaryotes and eukaryotes.** *Proc. Natl. Acad. Sci. USA* 1998, **95**:85–6.

30. Lorenz M, Heitman J: **Regulators of pseudohyphal differentiation in Saccharomyces cerevisiae identified through multicopy suppressor analysis in ammonium permease mutant strains.** *Genetics* 1998, **150**:1443–57.

31. Blaiseau P, Thomas D: **Multiple transcriptional activation complexes tether the yeast activator Met4 to DNA.** *EMBO. J.* 1998, **17**:6327–36.

32. Bauer E, Kohavi R: **An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants.** *Machine Learning* 1999, **36**:105–139.

33. Breiman L: **Bagging predictors.** *Machine Learning* 1996, **24**:123–40.

34. Freund Y, Schapire R: **Experiments with a New Boosting Algorithm.** In *International Conference on Machine Learning* 1996:148–156.

35. Weiss G, Provost F: **The Effect of Class Distribution on Classifier Learning: An Empirical Study.** Tech. Rep. ML-TR-44, Department of Computer Science, Rutgers University. 2001.

36. Chan PK, Stolfo SJ: **Toward Scalable Learning with Non-Uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection.** In *Knowledge Discovery and Data Mining* 1998:164–168.

37. Ho Y, Costanzo M, Moore L, Kobayashi R, Andrews B: **Regulation of transcription at the Saccharomyces cerevisiae start transition by Stb1, a Swi6-binding protein.** *Mol. Cell Biol.* 1999, **19**:5267–78.

38. Cai M, Davis R: **Yeast centromere binding protein CBF1, of the helix-loop-helix protein family, is required for chromosome stability and methionine prototrophy.** *Cell* 1990, **61**:437–46.

39. Lamb T, Mitchell A: **The transcription factor Rim101p governs ion tolerance and cell differentiation by direct repression of the regulatory genes NRG1 and SMP1 in Saccharomyces cerevisiae.** *Mol. Cell. Biol* 2003, **23**:677–86.

40. Landis J, Koch G: **The measurement of observer agreement for categorical data.** *Biometrics* 1977, **33**:159–74.

41. Quinlan R: *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann 1993.

42. Witten I, Frank E: *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* San Mateo, CA: Morgan Kaufmann 1999.

43. Bar-Joseph Z, Gerber G, Gifford D, Jaakkola T, Simon I: **Continuous representations of time-series gene expression data.** *J. Comput. Biol.* 2003, **10**:341–56.

44. Ferea T, Botstein D, Brown P, Rosenzweig R: **Systematic changes in gene expression patterns following adaptive evolution in yeast.** *Proc Natl Acad Sci U S A* 1999, **96**:9721–6.

45. Seki M, Ishida J, Narusaka M, Fujita M, Nanjo T, Umezawa T, Kamiya A, Nakajima M, Enju A, Sakurai T, Satou M, Akiyama K, Yamaguchi-Shinozaki K, Carninci P, Kawai J, Hayashizaki Y, Shinozaki K: **Monitoring the expression pattern of around 7,000 Arabidopsis genes under ABA treatments using a full-length cDNA microarray.** *Funct Integr Genomics* 2002, **2**:282–91.

46. Higo K, Ugawa Y, Iwamoto M, Korenaga T: **Plant cis-acting regulatory DNA elements (PLACE) database.** *Nucleic Acids Res.* 1999, **27**:297–300.