JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 13, Number 2, 2006

Pp. 463–480

Modeling the Combinatorial Functions of Multiple Transcription Factors

CHEN-HSIANG YEANG[1] and TOMMI JAAKKOLA[2]

ABSTRACT

A considerable fraction of gene promoters are bound by multiple transcription factors. It is therefore important to understand how such factors interact in regulating the genes. In this paper, we propose a computational method to identify groups of co-regulated genes and the corresponding regulatory programs of multiple transcription factors from protein-DNA binding and gene expression data. The key concept is to characterize a regulatory program in terms of two properties of individual transcription factors: the function of a regulator as an activator or a repressor, and its direction of effectiveness as necessary or sufficient. We apply a greedy algorithm to find the regulatory models which best explain the available data. Empirical analysis indicates that the inferred regulatory models agree with known combinatorial interactions between regulators and are robust against various parameter choices.

Key words: combinatorial function, gene regulation.

1. INTRODUCTION

THE COMBINATORIAL INTERACTIONS OF MULTIPLE TRANSCRIPTION FACTORS play an essential role in transcriptional regulation. For instance, many genes are regulated by protein complexes comprised of multiple transcription factors (McNabb *et al.*, 1995). To model the combinatorial interactions of transcription factors, it is necessary to relate the activity states of transcription factors to the expression levels of regulated genes. Finding this relation—a regulatory program—between regulators and regulated genes is a challenging problem since the number of possible regulatory programs grows rapidly with the number of transcription factors involved. The set of possible regulatory programs has to be simplified so as to be able to infer reasonable candidate programs from the available data.

In this paper, we present a computational method that identifies the regulatory programs of multiple transcription factors and the genes they regulate from both protein-DNA binding and gene expression data. The results are regulatory models; each contains a set of transcription factors, genes putatively regulated by these factors, and the regulatory program specifying the relation between regulators and regulated gene expressions. We simplify a regulatory program by characterizing it in terms of the functions and

[1]Center for Biomolecular Science and Engineering, University of California Santa Cruz, Santa Cruz, CA 95064.
[2]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139.

directions of effectiveness of individual regulators. This characterization gives a simple interpretation of the mechanisms underlying a regulatory program and greatly reduces the model complexity.

Modeling the transcriptional regulation of multiple transcription factors has been addressed in a considerable number of previous works. Most Bayesian network models of gene expression analysis (e.g., Friedman *et al.* [2000], Hartemink *et al.* [2001], and Segal *et al.* [2002]) focused only on the structure of a regulatory model and did not directly infer the regulatory program. Some authors considered the effects of single regulators separately and avoided identifying the combinatorial interactions of multiple regulators (e.g., Bar-Joseph *et al.* [2003]). Some works limited the scope to synergistic or complementary effects of regulator pairs, for example, Pilpel *et al.* (2001) and Tong *et al.* (2004). Others attacked the combinatorial functions of multiple regulators with different computational models, such as Boolean networks (Tanay *et al.*, 2001), regression trees (Segal *et al.*, 2003), system identification (Gardner *et al.*, 2003), and many others. However, since these models targeted only functional relations in the data, the resulting models can be sometimes difficult to interpret in terms of the underlying mechanisms. Some authors modeled the dynamic, physical-chemical processes of gene regulation with differential equations (e.g., Shea *et al.* [1985]). Although physical mechanisms were explicitly addressed in these models, it is difficult to apply them to large-scale systems due to the complexity of models and the lack of sufficient dynamic data. Another approach of modeling the circuitry of multiple regulators is to systematically generate different input states by perturbation and measure the response of regulated genes (for instance, Yuh *et al.* [1998]). This approach, though more reliable, can be expensive and time-consuming.

The remainder of the paper is organized as follows. We first introduce key hypotheses and concepts in our regulatory model and provide a mathematical formulation of the problem. We subsequently describe an algorithm to learn the models from binding and gene expression data, demonstrate the algorithm on the basis of CHIP-chip binding data and two large-scale gene expression datasets, and discuss the results and how to validate them. Finally, we summarize the pros and cons of the proposed method and provide possible directions to extend the approach.

2. MODELS OF TRANSCRIPTION REGULATION

2.1. Modeling hypotheses and concepts

We adopt several common hypotheses in the analysis of CHIP-chip and microarray data (Friedman *et al.*, 2000; Hartemink *et al.* 2001; Segal *et al.*, 2003). First, given that a transcription factor binds to a specific promoter, the activity of the factor is modulated by the factor's mRNA abundance. Second, genes co-regulated by a set of transcription factors (i.e., genes appearing in the same module) share the same regulatory program. For computational convenience, we also assume that the relative changes of mRNA levels with respect to a reference condition can be quantized into three states: up-regulation, down-regulation, and no change. Moreover, since microarrays measure gene expression levels over a population of cells, a lack of observed change may be due to a mixture population, where the same gene in some cells is up-regulated while down-regulated in others. We do not distinguish between insignificant changes in expression and those that cannot be predicted from the model.

The key idea of our model is to characterize a regulatory program in terms of two (assumed) properties of individual transcription factors. First, a transcription factor possesses a consistent function as an activator or a repressor so that the function is not inverted in the context of combinatorial control. Second, a transcription factor may have an effect only if its expression changes in certain direction. We categorize the direction of effectiveness into four types. A regulator is necessary if decreasing its expression level leads to responses opposite to its function (e.g., reducing the presence of an activator leads to repression). A regulator is sufficient if increasing its expression level leads to responses consistent with its function (e.g., higher concentrations of repressors cause further down-regulation of relevant genes). A regulator can be both necessary and sufficient, or neither necessary nor sufficient. Unlike the function of a single regulator, we allow the direction of effectiveness of each transcription factor to vary across different regulatory models that it participates in.

The regulatory program involving only a single regulator is uniquely determined by these two properties. Table 1 summarizes the predicted responses of different types of regulators in our categorization.

TABLE 1. RESPONSES OF REGULATED GENES IN EACH COMBINATORIAL CATEGORY

	Necessary	*Sufficient*	*Both*	*Neither*
Activator	$f \downarrow \Rightarrow g \downarrow$	$f \uparrow \Rightarrow g \uparrow$	$f \downarrow \Rightarrow g \downarrow$, $f \uparrow \Rightarrow g \uparrow$	g any value
Repressor	$f \downarrow \Rightarrow g \uparrow$	$f \uparrow \Rightarrow g \downarrow$	$f \downarrow \Rightarrow g \uparrow$, $f \uparrow \Rightarrow g \downarrow$	g any value

To specify a combinatorial function of multiple regulators, we have to define a response for each possible input state. The input state refers to the measured changes in the regulator expression levels relative to control. By assuming the function and the direction of effectiveness of each regulator are preserved in all input states, we can construct the combinatorial function from the predicted response of individual regulators. For each joint input state, the combinatorial function reports the consensus prediction if the individual predictions from applicable factors agree, and an uncertain output otherwise. Note that the consensus is nontrivial since not all factors make a prediction in all states. For example, a necessary activator (which is not also sufficient) makes no prediction in cases when its mRNA expression level has increased relative to a control. The rules of generating the output of the combinatorial function from predictions of individual regulators are described in Section 2.2.

The functional class generated by this characterization represents only a small subset of all possible combinatorial functions: the number of possible combinations of these two properties for n inputs is 8^n, whereas the number of all possible tri-state Boolean functions with n inputs is 3^{3^n}. The drastic reduction of possible functions helps in estimating the functions from limited data. While the number of possible functions is still exponential in n, we can enumerate the possibilities for small n.

Despite its simplification, characterization of a regulatory program with properties of single regulators still retains some combinatorial interactions between regulators. Some of these combinatorial effects have clear mechanistic interpretations. For example, if all regulators in a model are necessary, then they are likely to form a complex or cooperatively bind together on promoters. In contrast, if all regulators are sufficient, then they may independently act on promoters. In general, we can view a necessary regulator as essential for maintaining the basal transcription level and a sufficient regulator as providing an additive boost (activator) or inhibition (repressor) of gene expression.

2.2. Definition of a regulatory model

We define a model of transcription regulation to have three components: a set of transcription factors, a set of genes controlled by these transcription factors, and a regulatory program specifying the relation between the expression of regulators and genes they regulate. We first define a deterministic regulatory program as a function which maps the mRNA state of transcription factors into the mRNA state of a "typical" response of regulated genes,

$$f : S^n \rightarrow S \tag{1}$$

where $S = \{-1, 0, +1\}$ is the quantized state expression changes and n the input size. According to the module assumption, all regulated genes in a model are controlled by the same regulatory program.

The function of each regulator is uniquely determined by the type and the direction of effectiveness, as shown in Table 1. We adopt the following rules to synthesize the predicted response from responses of individual regulators. If the individual responses are all $+1$s or 0s, then the output is $+1$. If the predicted responses are all -1s or 0s, then the output is -1. If the predicted responses contain both $+1$s and -1s, or are all 0s, then the output is 0. These rules simply report the consensus of predicted responses and output 0 if consensus cannot be reached. Note that we do not distinguish between uncertain individual responses (insignificant changes) and responses that are uncertain due to conflicting predictions. We can thus construct the combinatorial function f from Table 1 and the consensus rules. An example of a deterministic combinatorial function of two necessary activators is shown in Table 2.

Relying on the deterministic mapping from states to responses is too rigid in the biological context. We instead construct a probabilistic regulatory program

$$P : S^n \times S \rightarrow [0, 1] \tag{2}$$

TABLE 2. THE COMBINATORIAL FUNCTION OF TWO NECESSARY ACTIVATORS

f_1	f_2	g
-1	-1	-1
-1	0	-1
-1	$+1$	-1
0	-1	-1
$+1$	-1	-1
o.w.	o.w.	0

TABLE 3. THE TABLE OF $P(c_{ge}|f(c_{Re}))$

$f(c_{Re})$	$P(c_{ge}=-1\|f(c_{Re}))$	$P(c_{ge}=0\|f(c_{Re}))$	$P(c_{ge}=+1\|f(c_{Re}))$
-1	$1-\alpha$	α	0
0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
$+1$	0	α	$1-\alpha$

on the basis of the deterministic one. The conditional probability of responses is related to the deterministic function in the following way. Denote c_{ge} as the expression state of regulated gene g in experiment e and c_{Re} as the expression state of regulator set R in experiment e. The conditional probability $P(c_{ge}|c_{Re}, f) \equiv P(c_{ge}|f(c_{Re}))$ depends on the regulated gene expression c_{ge} and the output of the deterministic function $f(c_{Re})$. The response c_{ge} that agrees with $f(c_{Re})$ is assigned a high probability, whereas the c_{ge} that directly contradicts $f(c_{Re})$ has zero probability. However, when $f(c_{Re}) = 0$, each c_{ge} state is assigned an equal probability, consistent with not distinguishing between insignificant and uncertain responses. Table 3 shows the conditional probability table, where $\alpha < 1$ is an adjustable parameter.

3. IDENTIFYING REGULATORY MODELS

In this section, we describe a method of identifying regulatory models from protein-DNA binding and gene expression data. We first define a scoring function (log likelihood function) for binding and expression data. Subsequently, we will adopt a greedy algorithm to find high scoring models and evaluate the significance of the resulting models.

3.1. Likelihood function of a regulatory model

The log likelihood function measures how well the regulatory model fits the relevant binding and expression data. It contains two terms. The term corresponding to binding data is the log likelihood ratio between the regulatory model, where each regulator binds to each regulated gene, and the null model that the binding of each (protein,promoter) pair occurs with probability $\frac{1}{2}$. The term corresponding to expression data is the log likelihood ratio between the regulatory model where the expression states of the regulators govern the responses, and the null model where no such relation exists. The joint scoring function is a weighted sum of these two terms.

More formally, let $M = (R, G, f)$ denote a regulatory model, where R and G are the regulators and regulated genes, respectively, and f is the (deterministic) regulatory program. For each $r \in R$ and $g \in G$, define b_{rg} as a binary random variable indicating whether r binds to g. State b_{rg} is observed only through noisy measurements x_{rg} from binding data. Let E be a collection of expression experiments. For each $r \in R$ and $e \in E$, c_{re} is the expression change of regulator r in experiment e; c_{re} is linked to noisy microarray measurements x_{re}. For each $g \in G$ and $e \in E$, c_{ge}, x_{ge} is defined analogously. Furthermore, define $\{b_{rg}\}$ as the joint state of all indicator variables $b_{rg}, r \in R, g \in G$; $\{c_{re}\}$ and $\{c_{ge}\}$ are defined analogously. Also denote c_{Re} as the state of all $c_{re}, r \in R$ in experiment e.

The marginal likelihood function of binding data under a hypothesis H is

$$P(\{x_{rg}\}|H) = \sum_{\{b_{rg}\}} P(\{b_{rg}\}|H)P(\{x_{rg}\}|\{b_{rg}\}). \tag{3}$$

The conditional probability $P(x_{rg}|b_{rg})$ corresponding to each protein-DNA interaction defines our confidence in the measurements. The conditional probability ratio $\frac{P(x_{rg}|b_{rg}=1)}{P(x_{rg}|b_{rg}=0)}$ is quantified as in Yeang *et al.* (2004) through a Bayesian model selection criterion. We omit the details.

We are interested in two $P(\{b_{rg}\}|H)$ priors. The only setting of $\{b_{rg}\}$ consistent with the regulatory model M is that each factor binds to all of the regulated genes. We refer to this hypothesis as H_1:

$$H_1 : P(\{b_{rg}\}|H_1) = \prod_{r\in R, g\in G} \delta(b_{rg} = 1) \tag{4}$$

where $\delta(.)$ is the indicator function. In contrast, for the null model H_0 under which the regulators do not have any specific relation to the genes, the prior probability of $\{b_{rg}\}$ is uniform

$$H_0 : P(\{b_{rg}\}|H_0) = \frac{1}{2^{|R||G|}}. \tag{5}$$

Based on the priors, and assuming that each measurement x_{rg} is independent of the others, the log likelihood ratio becomes

$$\begin{aligned} L^b(R, G) &= \log P(\{x_{rg}\}|H_1) - \log P(\{x_{rg}\}|H_0) \\ &= |R||G| \log 2 + \sum_{(r,g)} [\log P(x_{rg}|b_{rg} = 1) - \log(P(x_{rg}|b_{rg} = 1) + P(x_{rg}|b_{rg} = 0))]. \end{aligned} \tag{6}$$

The log likelihood ratio of expression data can be constructed similarly. The marginal likelihood function of the expression data under any hypothesis H (H_0 or H_1 discussed below) is given by

$$P(\{x_{re}\}, \{x_{ge}\}|H) = \sum_{\{c_{re}\},\{c_{ge}\}} P(\{c_{re}\}, \{c_{ge}\}|H)P(\{x_{re}\}|\{c_{re}\})P(\{x_{ge}\}|\{c_{ge}\}). \tag{7}$$

As in the case of binding data, we use a uniform null model over the possible expression states $\{c_{re}\}$ and $\{c_{ge}\}$:

$$H_0 : P(\{c_{re}\}\{c_{ge}\}|H_0) = \frac{1}{3^{|E|(|R|+|G|)}}. \tag{8}$$

The alternative model H_1 relates c_{ge} and c_{Re} in each experiment e. It is specified by function f and Table 3. Each state of the regulators c_{Re} is again assigned a uniform prior probability (as in H_0) but the responses are now governed by f:

$$H_1 : P(\{c_{re}\}\{c_{ge}\}|H_1) = \prod_{e\in E} \left[\frac{1}{3^{|R|}} \prod_{g\in G} P(c_{ge}|f(c_{Re})) \right]. \tag{9}$$

The conditional probabilities $P(\{x_{re}\}|\{c_{re}\})$ and $P(\{x_{ge}\}|\{c_{ge}\})$ relating the discrete variables to measurements are again specific to each dataset and assumed to be independent of the regulatory model. We will discuss the choice of measurement error models in Section 4.

Combining Equations (7), (8), (9), we evaluate the log likelihood ratio of expression data. Skipping intermediate steps,

$$\begin{aligned} L^e(R, G, f) &= \log P(\{x_{re}\}, \{x_{ge}\}|H_1) - \log P(\{x_{re}\}, \{x_{ge}\}|H_0) \\ &= -|E||R| \log 3 + \sum_{e \in E} \left[\log \left(\sum_{v \in \{-1,0,+1\}} P_v(e) \cdot \prod_{g \in G} \sum_{c_{ge}} P(c_{ge}|v) P(x_{ge}|c_{ge}) \right) \right] \\ &\quad + |E|(|R| + |G|) \log 3 - \sum_{e \in E} \\ &\quad \times \left[\sum_{r \in R} \log(P(x_{re}|c_{re} = +1) + P(x_{re}|c_{re} = -1) + P(x_{re}|c_{re} = 0)) \right. \\ &\quad \left. + \sum_{g \in G} \log(P(x_{ge}|c_{ge} = +1) + P(x_{ge}|c_{ge} = -1) + P(x_{ge}|c_{ge} = 0)) \right] \end{aligned} \tag{10}$$

where $P_v(e)$ denotes the probability of the regulator states in experiment e which generate deterministic output v:

$$P_v(e) = \sum_{\{c_{Re}\}} \delta(f(c_{Re}) = v) \cdot P(x_{Re}|c_{Re}). \tag{11}$$

We define the joint log likelihood ratio as the weighted sum of the log likelihood functions of binding and expression data:

$$L(R, G, f) = L^b(R, G) + \lambda L^e(R, G, f). \tag{12}$$

Parameter λ is a free parameter specifying the relative importance of expression data with respect to binding data. Since the number of expression experiments far exceeds the number of binding experiments, we have to down-weight the importance of expression data in order to keep the binding data relevant.

3.2. Algorithm for identifying regulatory models

We discuss here a greedy algorithm for optimizing the scoring function in Equation (12). The problem is difficult to solve exactly due to the enormous number of possible combinations of regulators and the genes that they potentially regulate. Our algorithm proceeds by incrementally incorporating regulated genes while, at each stage, identifying the optimal regulatory program. The key steps in the algorithm are the following:

1. Find a collection of regulator sets which co-bind to a set of genes according to the CHIP-chip data. The thresholds of determining significant binding events (the p-value threshold of binding data) and the minimum number of co-bound genes that constitute a module are free parameters. We use $p \leq 0.005$ and require regulators to co-bind to ≥ 10 genes. To simplify the computations involved, we consider only modules with ≤ 3 regulators.
2. For each candidate regulator set, we identify the highest scoring set of genes regulated by these factors and optimize the corresponding regulatory program. We are able to exhaust all possible regulatory programs due to the many simplifications discussed earlier. For each regulatory program, we incrementally add genes into the regulated set, so as to maximize the log likelihood score. Since Equation (12) increases with the number of regulated genes, we use a significance measure to stop adding genes. The p-value is discussed in in Section 3.3 and Appendix 2. Note that each gene may participate in multiple regulatory models. We then compare the scores of regulatory programs (each has a different gene set). Because the log likelihood score grows with the number of genes, we compare the scores of fixed sized

gene sets by choosing the top n (n is the fixed size) genes according to the order of adding genes. The fixed size is the size of the smallest gene set among all regulatory programs. The result of step 2 is a regulatory program and a regulated gene set for each regulator set.

3. Some of the regulatory programs may be spurious or do not have functional roles. We evaluate the p-value of a regulatory program log likelihood score by using a permutation test. Details will be discussed in Section 3.3 and Appendix 3.
4. Due to insufficient data, there are many regulatory programs which fit the data equally or nearly equally well. Thus, reporting one regulatory program may not be very informative. We report the direction of effectiveness for each regulator, which is the consensus among the estimated regulatory programs. We also evaluate the p-value of each reported direction of effectiveness. Details will be discussed in Section 3.3 and Appendix 4.

Step 2 has to be elaborated. Each regulatory program induces a different set of regulated genes. Because the log likelihood score in Equation (12) increases with the number of regulated genes, the regulatory program with the largest set of regulated genes will always be chosen if we maximize the joint log likelihood score. To remove the effect of different regulated gene set sizes, we fix the size of regulated gene sets in the following way. Recall each gene is incorporated in the model in a greedy fashion, so the first n genes of a regulatory program are the top n genes which best conform with the regulatory program. We discard the regulatory programs with small regulated gene sets (< 5 genes) and identify the minimum size among the remaining regulated gene sets. We then compare the log likelihood scores of regulatory programs on the fixed-sized regulated gene sets. This procedure is a tentative solution to alleviate the effect of gene set size on the log likelihood score. In the long run, a more principled way of normalizing Equation (12) in terms of regulated gene set size is needed.

3.3. Evaluating the significance of regulatory models

We have introduced three significance measures (p-values) in the algorithm. The first p-value evaluates the significance of adding a new gene in the regulated gene set. This p-value is calculated by comparing the increment of the log likelihood score generated from empirical data to the increment from random expression data. We consider a randomization scenario where $P(x_{ge}|c_{ge} = 0)$, $P(x_{ge}|c_{ge} = \pm 1)$ of the newly added gene are uniformly sampled from the simplex $P(x_{ge}|c_{ge} = 0) + P(x_{ge}|c_{ge} = -1) + P(x_{ge}|c_{ge} = +1) = 1$. In contrast to sampling methods that use different data permutations, the p-value we have defined can be approximated with an analytic expression. Details about the approximation are described in Appendix 2.

The second p-value evaluates the significance of a specific regulatory model. It is calculated from the following permutation test procedure. The expression data of regulated genes are randomly permuted (over genes and experiments). The optimal regulatory program and its log likelihood score from each permuted data are calculated, and the p-value is the fraction of optimal log likelihood scores from random data that exceed the empirical score. Details about the procedure are reported in Appendix 3.

The third p-value calculates the significance of the combinatorial property of a regulator. It is calculated according to the gap of log likelihood scores between the best model where this property holds and the best model where this property does not hold. For example, to evaluate the significance of "r_1 is a necessary activator," we find the optimal model M_1 among the models where r_1 is a necessary activator and the optimal model M_0 among the models where r_1 is not a necessary activator. We compare the empirical gap score with the gap scores obtained by randomly permuting gene expression data. Notice the gap score of each permuted data is obtained by reoptimizing the regulatory models to fit the permuted data. The p-value is the fraction of the random gap scores exceeding the empirical gap. Details about the procedure also can be seen in Appendix 4.

4. EMPIRICAL ANALYSIS

We applied the algorithm of identifying regulatory models to the protein-DNA interaction data of 106 transcription factors (Lee *et al.*, 2002) and two sets of large-scale gene expression data: Rosetta Compendium data of gene knock-outs (Hughes *et al.*, 2000) and stress response gene expression data published by Gasch *et al.* (2000). Rosetta data contains the log ratios and p-values of steady-state measurements,

whereas Gasch data provides log ratios of time-course measurements. For simplicity, we fix the regulatory functions (activators or repressors) of single regulators according to previous studies curated in the Incyte Yeast Proteome Databases (*www.incyte.com/login.html*).

The conditional probabilities $P(\{x_{rg}\}|\{b_{rg}\})$ of binding data and $P(\{x_{re}\}|\{c_{re}\})$ and $P(\{x_{ge}\}|\{c_{ge}\})$ of Rosetta gene expression data were evaluated using the approximation described by Yeang *et al.* (2004). The conditional probabilities $P(\{x_{re}\}|\{c_{re}\})$ and $P(\{x_{ge}\}|\{c_{ge}\})$ of the Gasch data were evaluated from Gaussian and exponential distributions of the time-course responses of perturbations. Details are described in Appendix 1.

We summarize and analyze the inferred models in the following aspects. We first visualize the regulatory models inferred from two expression datasets and discuss their inferred combinatorial properties. We then validate the inferred models with gene function ontology, literature survey, sensitivity analysis, and the agreement between the models inferred from the two datasets. Finally, we compared the models inferred from Rosetta and Gasch data to check whether certain regulatory models were manifested in multiple datasets.

4.1. Models inferred from Rosetta and Gasch data

Figures 1 and 2 summarize the information about regulatory models inferred from Rosetta and Gasch data. We consider only the regulatory models with up to three regulators and report only the models with significant p-values of likelihood scores ($p \leq 0.02$ for Rosetta data and $p \leq 0.001$ for Gasch data). We set a more stringent cutoff on Gasch data because more significant responses appear in Gasch data according to the quantization method described in Appendix 1. There are 110 valid models inferred from Rosetta data and 144 valid models inferred from Gasch data. We represent a regulatory model as a bipartite graph between regulators (circles) and a regulated gene set (a square). The color of a regulator indicates its regulatory function as an activator (red) or a repressor (green). The color of a regulated gene set indicates the MIPS functional categories enriched in the regulated gene set ($p \leq 0.06$ according to a hypergeometric test with Bonferroni correction). The color of an edge indicates the direction of effectiveness of a regulator in a model: red for necessary, green for sufficient, and black for neither. Two edges can exist between two nodes since a regulator can be both necessary and sufficient. The width of an edge indicates the confidence about about necessity or sufficiency as described in Section 3.3. We use the visualization software Cytoscape (*www.cytoscape.org*) to draw the graphs. The complete list of inferred regulatory models is also tabulated in the Supplementary Webpage Section.

We found the combinatorial properties of many inferred regulatory models to be consistent with the knowledge about the combinatorial interactions of these transcription factors. We summarize these interactions into three categories and draw a number of illustrative examples for each category.

- Each regulator is necessary for a regulated gene set. This pattern appears in regulator pairs such as (Ino2,Ino4), (Swi4,Swi6), (Swi6,Mbp1), and (Fkh1,Fkh2) in Rosetta models. These regulator pairs are known to be components of protein complexes for transcriptional activation. Ino2-Ino4 complex regulates genes involved in phospholipid synthesis (Ambroziak *et al.*, 1994). Protein complexes Swi6-Swi4, Swi6-Mbp1, and Fkh1-Fkh2 activate genes expressed during G1/S, S/G2, or G2/M phases of the yeast cell cycle (Simon *et al.*, 2001).
- Each regulator is sufficient for a regulated gene set. This pattern is common for stress response regulators, for example, (Msn4,Yap1), (Msn2,Yap1), (Msn2,Hsf1) pairs in Rosetta data and (Msn4,Hsf1), (Msn4,Yap1) in Gasch data. This pattern is consistent with the property that each stress response regulator either activates the gene under a different stress condition (for example, Hsf1 for heat shock and Yap1 for hyperoxia) or contributes in an additive or redundant fashion (for example, Msn2 and Msn4) (Gasch *et al.*, 2000).
- Some regulators are both necessary and sufficient, and the others are not strongly effective in either direction. Examples in Rosetta models include several small modules co-regulated by Gcn4 and one of the following regulators involved in amino acid synthesis: Leu3, Cbf1, Abf1, and several ribosome gene sets regulated by Rap1, Fhl1 and several other factors in Gasch models. In these examples, there exist some "master regulators" which control genes in both directions, while other regulators are not correlated with regulated genes at expression levels. This property does not necessarily exclude the functional role of these "inactive" regulators. They may be possible cofactors which regulate transcription via other mechanisms.

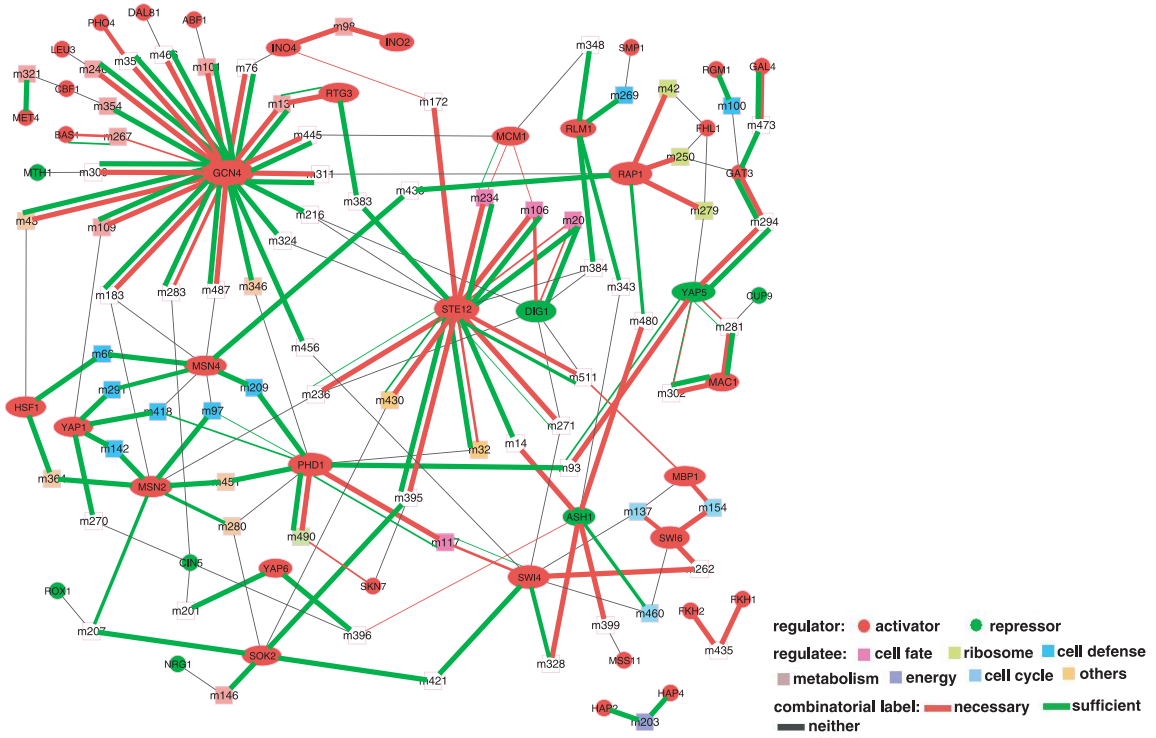


FIG. 1. Models inferred from Rosetta data.

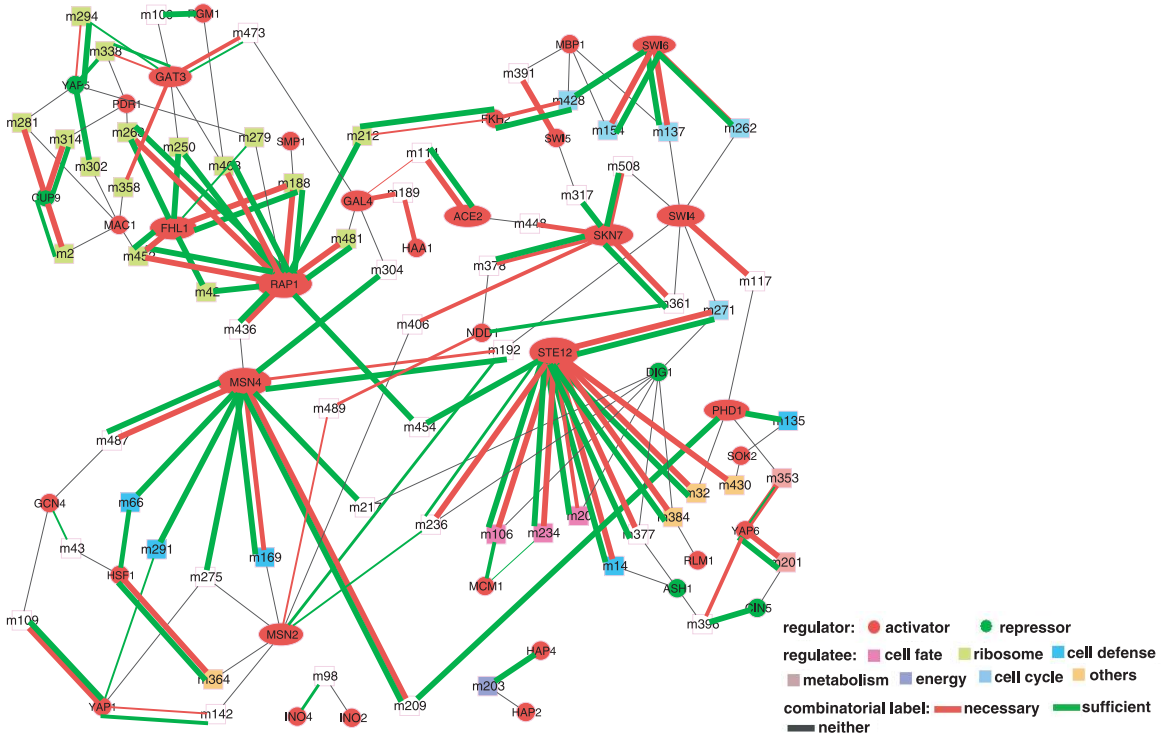


FIG. 2. Models inferred from Gasch data.

Since our regulatory models are based on simplifying assumptions, many true combinatorial interactions of regulators are not retrieved. It is difficult to assess the false negatives of the algorithm due to the lack of complete knowledge about combinatorial gene regulation. Instead, we draw several illustrative examples from known combinatorial interactions of yeast genes.

- The well-known interaction of Gal4-Gal80 complex on galactose metabolic genes does not appear in Figs. 1 and 2. The Rosetta module regulated by Gal4 (m473) is not enriched with galactose metabolic genes, and Gal80 does not appear in Figs. 1 and 2. This is because the expression level of Gal4 is low even under active state (Hartemink *et al.*, 2001). Hence, its regulatory function on galactose metabolic genes cannot be revealed by expression data alone. Although Gal80 expression level is known to modulate in certain datasets (e.g., Hartemink *et al.* [2001]), it does not vary significantly in either Rosetta or Gasch data.
- The combinatorial interaction of Ste12 and Dig1 on pheromone response genes is only partially retrieved. Dig1 inhibits the phosphorylation of Ste12 (Bardwell *et al.*, 1998); hence, the inhibitory function of Dig1 is valid only when Ste12 is present. This combinatorial function cannot be captured by our models since the effectiveness of a regulator depends on the state of other regulators.
- Sok2 is known to be both an activator and repressor for different genes (Shenhar *et al.*, 2001). We assign it as a repressor since it represses more genes. However, this assignment also excludes the regulatory models where Sok2 is an activator.

4.2. Validation of inferred models

In addition to the qualitative properties described in Section 4.1, we validated the inferred models using four quantitative tests. First, we investigated whether regulated genes are enriched with functional categories according to Munich Information Center for Protein Sequences (MIPS) database (*http://mips.gsf.de/*). Second, we checked from previous works whether regulators participating in the same model were known to have functional interactions. Third, we demonstrated that the inferred models were robust against the variation of free parameter values.

For each regulatory model, we evaluated the hypergeometric p-values of the enrichment of MIPS categories with Bonferroni correction. We considered the models with significant log likelihood values (permutation p-value ≤ 0.02 for Rosetta models and p-value ≤ 0.001 for Gasch models, including the models of single regulators). Overall, about half of the inferred models are enriched with at least one MIPS category ($p \leq 0.06$): 46% of the Rosetta models (51 out of 110) and 45% of the Gasch models (65 out of 144) are enriched. Due to the incompleteness of the MIPS database and the conservative estimation of Bonferroni correction, more inferred models are expected to be involved in specific cellular processes.

We also searched PubMed and Incyte Yeast Proteome Databases (*www.incyte.com/login.html*) to check whether regulators participating in the same model were known to jointly control one or multiple genes. More than two-thirds of the regulator sets in the significant models were verified in previous works: 60% of the significant Rosetta models with multiple regulators (46 out of 77) and 67% of the significant Gasch models with multiple regulators (46 out of 69) contain regulators whose interactions were reported in previous works. The functional enrichment of regulated genes and previously reported combinatorial interactions of regulators in inferred models are reported in the Supplementary Webpage Section.

We further demonstrated that the inferred models were robust against the variations of three free parameters: λ, which appeared in the joint log likelihood function (Equation (12)), is the relative weight between expression and binding data; α in Table 3 relates the the prediction of a regulatory program to the hidden states of expression changes; and p^{stop} in the greedy algorithm specifies the stopping criterion of the p-values of adding genes (Section 3.2). The default settings of these parameters are $\lambda = 0.1$, $\alpha = \frac{1}{3}$, $p^{stop} = 0.1$. We performed robustness tests by varying each parameter while fixing the other two as the default values. Inferred models generated from the new parameter settings were compared to the default models in two aspects. First, we calculated the average overlap rate of regulated gene sets (with respect to the default models) over all models. Second, we counted the fraction of new models which had identical inferred directions of effectiveness to the default models. Figure 3 shows the sensitivity of parameters in the Rosetta and Gasch models. Both sensitivity measures are very robust against each parameter in each dataset except α on Rosetta data. For example, when varying λ from 0.01 to 0.9, the average overlap rate of Gasch models ranges between 90% and 100%, and more than 85% of inferred models agree on

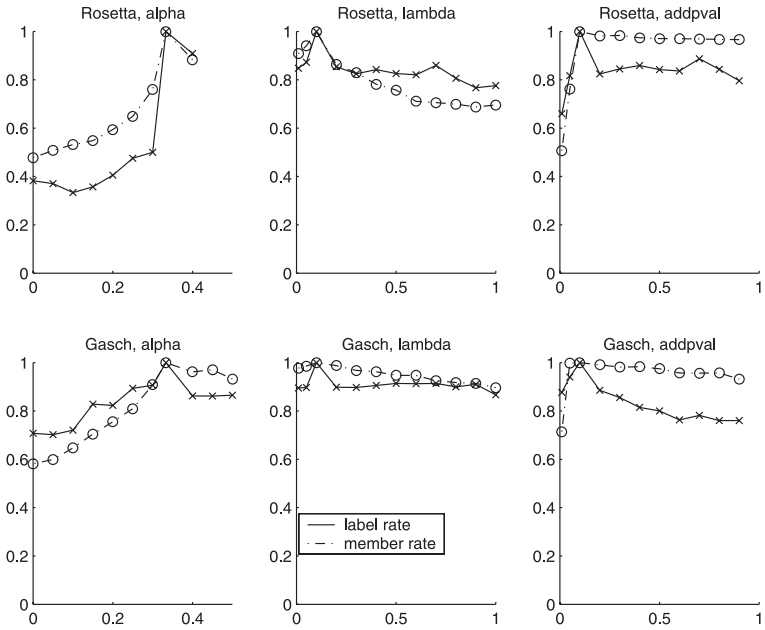


FIG. 3. Robustness tests on parameters. **Top:** Rosetta data. **Bottom:** Gasch data. Solid line: overlap of combinatorial labels. Dashed line: overlap of regulated gene sets.

directions of effectiveness. In contrast, models inferred from Rosetta data are sensitive to α: the average overlap rate drops to 50% when α varies from $\frac{1}{3}$ to 0.1.

4.3. Overlap between Rosetta and Gasch models

The quality of inferred models can also be judged by the robustness of models with respect to different datasets. We have generated two sets of regulatory models from Rosetta and Gasch expression data, respectively. A natural question is to what extent these models are overlapped. The consensus parts of the models suggest they are more likely to reflect the underlying system. The parts where they differ may be due to condition-specific properties of gene regulation: certain regulatory systems are revealed only in one dataset. However, without further validation, these results are less confident.

We investigate the overlapped parts between Rosetta and Gasch models. There are 110 significant Rosetta models (permutation p-value ≤ 0.02) and 144 significant Gasch models (permutation p-value ≤ 0.001). We consider the interesection of the regulator sets of these models and find 42 regulator sets appear in both sets of models. We then compare Rosetta and Gasch models corresponding to these 42 regulator sets in two aspects. First, we check the overlap between the regulated gene sets of the corresponding models. Second, we inspect the consensus of inferred directions of effectiveness of the corresponding models.

Tables 4 and 5 shows the comparison results of Rosetta and Gasch models. They suggest that these two sets of models are significantly overlapped. Among the 42 regulatory model pairs, about 60% of them (25 out of 42) are significantly overlapped in their regulated gene sets (more than 40% of members with respect to either model are overlapped). Moreover, the inferred directions of effectiveness are identical from both datasets in 83% (35 out of 42) of the models. That means all necessary regulators or all sufficient regulators in the two models coincide.

By inspecting the overlapped regulatory models, we find they correspond to the regulatory processes which are better captured in both datasets. Among the 42 significant models whose regulator sets appear in both datasets, 31 are enriched with genes belonging to certain MIPS categories. Moreover, the fraction of enriched gene sets increases when we consider the 25 models which are significantly overlapped in the regulated gene members (21 out of 25) and the 35 models which agree upon the directions of effectiveness (29 out of 35). The regulatory processes involved in these overlapped models include stress responses, mating responses, ribosomal regulation and cell cycle.

TABLE 4. OVERLAP OF INFERRED MODELS BETWEEN ROSETTA AND GASCH DATA, PART 1[a]

Regulator									*Overlap*	*Overlap*
1	*2*	*3*	*SizeR*	*SizeG*	*NeceR*	*NeceG*	*SuffR*	*SuffG*	*rate1*	*rate2*
Fkh2			7	50	1	1	0	1	42.9%	6%
Reb1			27	37	1	1	0	1	48.1%	35.1%
Sok2			54	44	0	1	1	1	5.6%	6.8%
Dig1	Ste12		57	46	11	01	10	01	31.6%	39.1%
Rap1			31	148	1	1	0	1	93.5%	19.6%
Rap1	Fhl1		28	121	11	10	00	00	100%	23.1%
Fkh1			12	9	1	1	0	1	16.7%	22.2%
Ste12			49	107	1	1	1	1	28.6%	13.1%
Hap4			24	40	0	1	1	1	95.8%	57.5%
Gcn4			125	12	1	0	1	1	6.4%	66.7%
Gal4			16	56	0	1	1	1	43.8%	12.5%
Msn4	Hsf1		8	22	00	01	11	11	37.5%	13.6%
Sum1			7	43	1	1	0	0	14.3%	2.32%
Fhl1			14	130	0	1	1	1	0%	0%
Msn4			13	16	0	1	1	1	23.1%	18.8%
Dig1	Ste12	Mcm1	19	25	010	010	000	010	57.9%	44.0%
Phd1	Swi4		9	15	01	00	10	10	11.1%	6.7%
Swi5			19	29	1	1	0	1	42.11%	27.59%
Rtg3	Gcn4		25	18	00	00	11	01	56%	77.8%
Msn2	Yap1		13	25	00	01	11	11	69.2%	36.0%
Yap5			7	62	1	0	1	1	0%	0%
Msn2	Msn4		55	31	00	11	11	11	36.4%	64.5%

[a]R: Rosetta, G: Gasch, rate1: wrt Rosetta modules, rate2: wrt Gasch modules.

TABLE 5. OVERLAP OF INFERRED MODELS BETWEEN ROSETTA AND GASCH DATA, PART 2[a]

Regulator									*Overlap*	*Overlap*
1	*2*	*3*	*SizeR*	*SizeG*	*NeceR*	*NeceG*	*SuffR*	*SuffG*	*rate1*	*rate2*
Aro80			9	25	0	1	1	1	77.8%	28%
Ash1			48	22	1	1	0	1	18.8%	40.9%
Dig1			66	23	1	1	1	1	18.2%	52.2%
Rap1	Fkh2		17	16	10	11	00	00	64.7%	68.8%
Ste12	Mcm1		24	30	11	10	10	00	54.2%	43.3%
Dig1	Ste12	Msn2	8	15	010	010	010	000	75%	40%
Gcr2			7	12	0	1	0	1	0%	0%
Rap1	Gat3	Fhl1	20	61	100	100	000	000	100%	32.8%
Pdr1	Rap1	Fhl1	18	28	000	010	000	000	100%	64.3%
Msn2	Msn4	Yap1	17	24	000	000	111	011	70.6%	50%
Rap1	Fhl1	Yap5	20	52	000	100	000	000	100%	38.5%
Mac1	Cup9		5	10	000	000	010	001	0%	0%
Msn4	Yap1		22	22	00	01	11	11	59.1%	59.1%
Gat3	Yap5		7	38	00	10	00	11	0%	0%
Mac1	Yap5		5	18	00	11	00	01	0%	0%
Fzf1			6	24	0	1	1	1	0%	0%
Rtg3			47	29	1	1	1	1	31.9%	51.7%
Msn2	Hsf1		10	26	00	01	11	11	30%	11.5%
Fkh1	Fkh2		12	24	11	11	00	01	75%	37.5%
Msn4	Rap1		7	10	00	01	11	01	0%	0%

[a]R: Rosetta, G: Gasch, rate1: wrt Rosetta modules, rate2: wrt Gasch modules.

5. DISCUSSION

We have described a simple computational approach to capture combinatorial effects of multiple transcription factors in transcription control. We identify regulatory models—including subsets of regulators and genes together with a regulatory program—from binding and expression data. We define regulatory programs with multiple regulators according to two properties of single transcription factors: 1) the function of a regulator and 2) its direction of effectiveness. The inferred models agree substantially with known functions and interactions. Moreover, the inferred models are robust against specific parameter values.

There are, however, many unresolved issues. Most combinatorial functions cannot be reduced to the properties of individual regulators. For example, the direction of effectiveness of a regulator may depend on the state of other regulators. The assumptions in our model are simplistic. For example, some regulators are not modulated through mRNA (protein) levels but primarily by altering protein modification states (Lee *et al.*, 1999). Binding and expression data alone are unlikely to capture such regulatory effects. A transcription factor can be both activator and repressor, depending on the cofactors it interacts with and the sets of regulated genes. Finally, some of the inferred models do not correspond to known biological functions and may be false positives. Better error models are needed to weed out a greater fraction of false positives.

APPENDIX 1. QUANTIZATION OF TIME-COURSE EXPRESSION DATA

In this appendix, we will show a method of evaluating the conditional probabilities $P(x_{re}|c_{re})$ and $P(x_{ge}|c_{ge})$ from time-course gene expression data. In the stress response dataset, x_{re} and x_{ge} are time-course measurements of expression responses under a stress condition. The goal is to convert x_{re} into conditional probabilities $P(x_{re}|c_{re}=+1)$, $P(x_{re}|c_{re}=-1)$, $P(x_{re}|c_{re}=0)$.

Denote $y \in \{-1, 0, +1\}$ as the actual, quantized expression change of a gene under one experimental condition and $x(t_1), \ldots, x(t_n)$ as its n time-course measurements. We relate the discrete state y to measurements $x(t_1), \ldots, x(t_n)$ with a two-level process. The discrete state y generates a continuous time-course expression profile $m(t_1), \ldots, m(t_n)$; and $x(t_1), \ldots, x(t_n)$ are noisy measurements of $m(t_1), \ldots, m(t_n)$. We model measurement errors $x(t_1) - m(t_1), \ldots, x(t_n) - m(t_n)$ as iid Gaussian random variables with zero mean and variance σ^2.

The actual expression profile $m(t_1), \ldots, m(t_n)$ is a zero vector given $y = 0$. Thus, $P(x(t_1), \ldots, x(t_n)| y = 0)$ is the product of normal densities:

$$P(x(t_1), \ldots, x(t_n)|y=0) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} \prod_{i=1}^{n} e^{-\frac{x(t_i)^2}{2\sigma^2}}. \tag{13}$$

We model the prior probabilities $P(m(t_1), \ldots, m(t_n)|y = \pm 1)$ with an iid exponential distribution:

$$\begin{aligned} P(m(t_1), \ldots, m(t_n)|y=+1) &= \prod_{t_i=1}^{n} P(m(t_i)|y=+1), \\ P(m(t_i)|y=+1) &= \begin{cases} \gamma e^{-\gamma m(t_i)} & \text{if } m(t_i) \geq 0, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{14}$$

The expression $P(m(t_1), \ldots, m(t_n)|y = +1)$ assigns a nonzero probability to each nonnegative expression profile, and penalizes the expression profiles deviating from 0; $P(m(t_1), \ldots, m(t_n)|y = -1)$ is defined analogously. By marginalizing over $m(t_i)$, the conditional probability $P(x(t_1), \ldots, x(t_n)|y = +1)$ becomes

$$\begin{aligned} P(x(t_1), \ldots, x(t_n)|y=+1) &= \prod_{i=1}^{n} \int_0^{\infty} P(m(t_i)|y=+1) P(x(t_i)|m(t_i)) dm(t_i) \\ &= \prod_{i=1}^{n} \gamma e^{(-\gamma x(t_i) + \frac{1}{2}\gamma^2\sigma^2)} \left(1 - \Phi\left(\frac{-(x(t_i) - \gamma\sigma^2)}{\sigma}\right)\right) \end{aligned} \tag{15}$$

where $\Phi(.)$ is the standard normal cumulative distribution function. Similarly,

$$\begin{aligned} P(x(t_1), \ldots, x(t_n)|y = -1) &= \prod_{i=1}^{n} \int_{-\infty}^{0} P(m(t_i)|y = -1) P(x(t_i)|m(t_i)) dm(t_i) \\ &= \prod_{i=1}^{n} \gamma e^{(\gamma x(t_i) + \frac{1}{2}\gamma^2\sigma^2)} \left(\Phi \left(\frac{-(x(t_i) + \gamma\sigma^2)}{\sigma} \right) \right) \end{aligned} \tag{16}$$

where σ and γ are free parameters. In the empirical analysis, we set $\sigma = \gamma = 0.5$, for they are close to the variance of the entire Gasch data.

APPENDIX 2. CALCULATING P-VALUES OF ADDING A NEW GENE IN THE MODEL

The incremental algorithm of finding the regulated gene set stops when the p-value of adding a new gene to the model becomes insignificant. In this appendix, we describe a method of analytically computing the p-value of gene addition.

Recall the expression log likelihood score of the regulatory model (Equation (10)):

$$\begin{aligned} L^e(R, G, f) = &-|E||R| \log 3 + \sum_{e \in E} \left[\log \left(\sum_{v \in \{-1,0,+1\}} P_v(e) \cdot \prod_{g \in G} \sum_{c_{ge}} P(c_{ge}|v) P(x_{ge}|c_{ge}) \right) \right] \\ &+ |E|(|R| + |G|) \log 3 - \sum_{e \in E} \\ &\times \left[\sum_{r \in R} \log(P(x_{re}|c_{re} = +1) + P(x_{re}|c_{re} = -1) + P(x_{re}|c_{re} = 0)) \right. \\ &\left. + \sum_{g \in G} \log(P(x_{ge}|c_{ge} = +1) + P(x_{ge}|c_{ge} = -1) + P(x_{ge}|c_{ge} = 0)) \right]. \end{aligned} \tag{17}$$

To simplify calculation, we assume $P(x_{ge}|c_{ge} = +1) + P(x_{ge}|c_{ge} = -1) + P(x_{ge}|c_{ge} = 0) = 1$. Hence, we can ignore all the constant terms independent of f:

$$\begin{aligned} L^e(R, G, f) &= C + \sum_{e \in E} \left[\log \left(\sum_{v \in \{-1,0,+1\}} P_v(e) \cdot \prod_{g \in G} \sum_{c_{ge}} P(c_{ge}|v) P(x_{ge}|c_{ge}) \right) \right] \\ &= C + \sum_{e \in E} T_e \end{aligned} \tag{18}$$

where $T_e = \log[\sum_{v \in \{-1,0,+1\}} P_v(e) \cdot \prod_{g \in G} \sum_{c_{ge}} P(c_{ge}|v) P(x_{ge}|c_{ge})]$.

The p-value of adding a new gene is based on the randomization scenario of uniformly sampling conditional probabilities $P(x_{ge}|c_{ge})$ of the newly added gene from the simplex $P(x_{ge}|c_{ge} = +1) + P(x_{ge}|c_{ge} = -1) + P(x_{ge}|c_{ge} = 0) = 1$. The conditional probabilities of regulators and genes already in the regulated set remain intact. If we are able to calculate the distribution of each T_e, then we can approximate the distribution of $\sum_e T_e$ using the central limit theorem and thereby evaluate the p-value of the empirical data.

However, T_e depends on the conditional probabilities of other genes in the regulated set as well. The distribution is more difficult to evaluate when the size of the regulated set grows. Instead, we replace T_e

with another statistic:

$$T_e^{'} = \log\left[\sum_{v\in\{-1,0,+1\}} P_v(e)\cdot\sum_{c_{ge}} P(c_{ge}|v)P(x_{ge}|c_{ge})\right]. \tag{19}$$

Approximately, $T_e^{'}$ corresponds to the contribution of the newly added gene on experiment e to the log likelihood score. We introduce the following notations: $p_{ge} = P(x_{ge}|c_{ge} = +1)$, $q_{ge} = P(x_{ge}|c_{ge} = -1)$, $u_{ge} = P(x_{ge}|c_{ge} = 0)$. Unfold the equation with these notations:

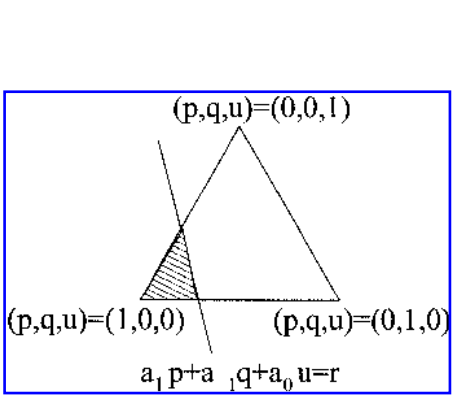
$$\begin{aligned} T_e^{'} &= \log\Big(P_1(e)((1-\alpha)p_{ge}+\alpha u_{ge}) + P_{-1}(e)((1-\alpha)q_{ge}+\alpha u_{ge}) \\ &\quad + P_0(e)\left(\frac{1}{3}p_{ge}+\frac{1}{3}q_{ge}+\frac{1}{3}u_{ge}\right)\Big) \\ &\equiv \log(a_{1e}p_{ge}+a_{-1e}q_{ge}+a_{0e}u_{ge}). \end{aligned} \tag{20}$$

We want to calculate the cumulative distribution of $T_e^{'}$ when (p_{ge}, q_{ge}, u_{ge}) is uniformly sampled from the simplex $S \equiv \{(p_{ge}, q_{ge}, u_{ge})|p_{ge}+q_{ge}+u_{ge} = 1, 0 \le p_{ge}, q_{ge}, u_{ge} \le 1\}$. This is equivalent to calculating the area of a polytope in the 3-D simplex. Without loss of generality, assume $a_{1e} > a_{0e} > a_{-1e}$ and r as the empirical value of $T_e^{'}$. As shown in the left graph of Fig. 4, $\{(p_{ge}, q_{ge}, u_{ge})|(p_{ge}, q_{ge}, u_{ge}) \in S, T_e^{'}(p_{ge}, q_{ge}, u_{ge}) \le r\}$ is the shaded area within the simplex. Thus, the density function of $exp(T_e^{'})$ is a saw-tooth function as shown in the right graph of Fig. 4. The density of $T_e^{'}$ is

$$\begin{aligned} \pi(y_e) &\equiv Pr(y_e \le \log(a_{1e}p_{ge}+a_{-1e}q_{ge}+a_{0e}u_{ge}) \le y_e+dy_e) \\ &= \begin{cases} \dfrac{2(e^{y_e}-a_{-1})e^{y_e}dy_e}{(a_1-a_{-1})(a_0-a_1)} & \log(a_{-1}) \le y_e \le \log(a_0), \\ \dfrac{2(a_1-e^{y_e})e^{y_e}dy_e}{(a_1-a_{-1})(a_1-a_0)} & \log(a_0) \le y_e \le \log(a_1). \end{cases} \end{aligned} \tag{21}$$

The p-value of the approximated test statistic is

$$p = Pr\left(T = \sum_e y_e \ge T^0\right). \tag{22}$$



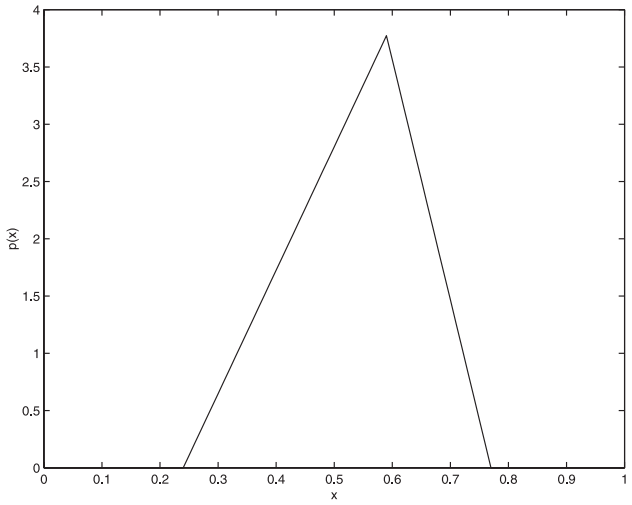


FIG. 4. Density function of the test statistic.

The density of T is the convolution of densities $\pi(y_1) \star \pi(y_2) \star \cdots \star \pi(y_{|E|})$. Its calculation requires the integrations of piecewise $|E|$-order polynomials and is tedious. We use the central limit theorem to approximate the density of T as Gaussian. The mean and variance of the Gaussian density are the average of the mean and variance of individual densities, which can be efficiently evaluated. Denote the mean and variance as μ and σ, the p-value is

$$p = 1 - \phi\left(\frac{T^0 - T}{\sigma}\right). \tag{23}$$

where ϕ is the standard Gaussian cumulative distribution.

APPENDIX 3. CALCULATING p-VALUES OF REGULATORY MODELS

The significance of a regulatory model is evaluated by comparing the empirical likelihood score with the likelihood scores of randomly permuted data. Since we preselect candidate regulator sets and regulated gene sets by thresholding on the p-values of binding data, the binding part of the empirical likelihood score is expected to be significant. Hence, we focus on testing the expression part of the score.

We perform the following procedures to obtain permuted data. We fix the expression data of regulators and randomly permute (both genes and conditions) the expression data restricted to the regulated gene set. We then reoptimize the regulatory programs which best fit the permuted data. The p-value of the empirical likelihood score is the fraction of random trials which yield likelihood scores higher than the empirical value. Notice we do not regenerate regulated gene sets by using the incremental algorithm, but fix the regulated gene set and reshuffle their expression data. This procedure may overestimate the significance of the empirical value since randomization is limited to the selected subset. However, it also avoids time-consuming randomization over the whole dataset.

APPENDIX 4. CALCULATING p-VALUES OF THE COMBINATORIAL PROPERTIES OF REGULATORS

Often, expression data do not uniquely determine a combinatorial function: there are multiple functions which fit the data equally or nearly equally well. It can be misleading to report only the optimal function(s) since they may contain spurious information. For instance, suppose r is an activator that control gene set G. In some experiments of dataset E, both r and G are down-regulated; r and G are unchanged in the remaining experiments. In our construction, the models "r is necessary" and "r is both necessary and sufficient" fit the data equally well. This is because no input states in E can test whether r is sufficient. If the model that "r is both necessary and sufficient" yields a higher score due to noise in the data, then we may draw a wrong conclusion from the reported model.

This problem is alleviated by also reporting the confidence pertaining to the combinatorial property of each regulator. If the combinatorial property (whether a regulator is necessary or sufficient) is relevant in the data, then it should contribute to fit the expression data. In other words, the likelihood score will be substantially degraded if we remove this combinatorial property from the model.

Testing the contribution of a combinatorial property is the rationale for evaluating the significance of this property. The gap between the likelihood score of the optimal model where this property holds and the optimal model where this property does not hold measures the contribution of this property to fit the data. In the previous example, the best model where "r is sufficient" holds is that r is both necessary and sufficient, and the best model where "r is not sufficient" holds is that r is necessary. Obviously, these two models yield very similar likelihood scores. Hence, the property that r is sufficient is not significant.

We apply a permutation test to calculate the p-value of a combinatorial property. We permute the expression data of both the regulator set and regulated gene set together. In every permutation, we identify the optimal model where a combinatorial property holds and the optimal model where it does not hold. We then calculate the difference of likelihood scores between the two models. The p-value is the fraction of random permutations which yield gap scores greater than the empirical gap score.

TABLE 6. THE TESTED MIPS FUNCTIONAL CATEGORIES

Index	*Function*	*Index*	*Function*
01	Metabolism	08	Cellular transport
02	Energy	10	Cellular communication
03	DNA processing	11	Stress response
04	Transcription	13	Regulation w/ environment
05	Protein synthesis	14	Cell fate, cell cycle
06	Protein fate		

APPENDIX 5. EVALUATING FUNCTIONAL ENRICHMENT IN REGULATED GENE SETS

We apply the standard hypergeometric test with multiple-hypothesis correction to evaluate the functional enrichment in the regulatory models. Suppose there are a total of n genes with n_1 genes belonging to a specific functional category. A subset contains m genes with m_1 genes belonging to this category. The p-value of the hypergeometric test is the probability that by randomly drawing m genes (without replacement) from the sample, $\geq m_1$ of them belong to this category. This probability is

$$p = \sum_{k=m_1}^{min(m,n_1)} \frac{\binom{n_1}{k}\binom{n-n_1}{m-k}}{\binom{n}{m}}. \tag{24}$$

When there are multiple categories, there is a higher probability that a randomly drawn subset is enriched with members in any of these categories. Suppose there are l categories and p' is the minimum of the empirical hypergeometric p-values among these categories. The overall p-value is the probability that at least one of the randomly sampled p-values $\leq p'$:

$$p = Pr((p_1 \leq p') \cup (p_2 \leq p') \cup \cdots \cup (p_l \leq p')) \leq \sum_{i=1}^{l} Pr(p_i \leq p') = lp'. \tag{25}$$

The inequality arises from the union bound of random events, and $Pr(p_i \leq p') = p'$ since a p-value is uniformly distributed under the null hypothesis. Therefore, the p-value of enrichment in at least one category is the minimum p-value of enrichment among each category multiplies the number of categories.

We apply the hypergeometric test on 11 MIPS functional categories shown in Table 6.

SUPPLEMENTARY WEBPAGE

Details about the inferred regulatory models can be found in the supplementary webpage *www.csail.mit.edu/~tommi/suppl/jcb05/*.

ACKNOWLEDGEMENTS

This work was supported in part by NIH grants GM68762 and GM69676. We thank Julia Zeitlinger and Ernst Fraenkel from the MIT Whitehead Institute and thank John Barnett, Georg Gerber, Karen Sachs, Jason Rennie, and David Gifford from the MIT Computer Science and Artificial Intelligence Laboratory for helpful commments and discussions.

REFERENCES

Ambroziak, J., and Henry, S.A. 1994. INO2 and INO4 gene products, positive regulators of phospholipid biosynthesis in *Saccharomyces cerevisiae*, form a complex that binds to the INO1 promoter. *J. Biol. Chem.* 269(21), 15344–15349.

Bardwell, L., Cook, J.G., Zhu-Shimoni, J.X., Voora, D., and Thorner, J. 1998. Differential regulation of transcription: Repression by unactivated mitogen-activated protein kinase Kss1 requires Dig1 and Dig2 proteins. *PNAS* 95(26), 15400–15405.

Bar-Joseph, Z., Gerber, G.K., Lee, T.I., Rinaldi, N.J., Yoo, J.Y., Robert, F., Gordon, D.B., Fraenkel, E., Jaakkola, T.S., Young, R.A., and Gifford, D.K. 2003. Computational discovery of gene modules and regulatory networks. *Nature Biotech.* 21, 1337–1342.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. 2000. Using Bayesian networks to analyze expression data. *J. Comp. Biol.* 7, 601–620.

Gardner, T.S., di Bernardo, D., Lorenz, D., and Collins, J.J. 2003. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* 301, 102–105.

Gasch, A.P., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D., and Brown, P.O. 2000. Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell* 11(12), 4241–4257.

Hartemink, A.J., Gifford, D.K., Jaakkola, T.S., and Young, R.A. 2001. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *PSB*, 422–433.

Hughes, T.R., Marton, M.J., Jones, A.R., Roberts, C.J., Stoughton, R., *et al.* 2000. Functional discovery via a compendium of expression profiles. *Cell* 102, 109–126.

Lee, J., Godon, C., Lagniel, G., Spector, D., Garin, J., Labarre, J., and Toledano, M.B. 1999. YAP1 and SKN7 control two specialized oxidative stress response regulons in yeast. *J. Biol. Chem.* 274(23), 16040–16046.

Lee, T., Rinaldi, N.J., Robert, F., Odom, D.T., Bar-Joseph, Z., *et al.* 2002. A transcriptional regulatory network map for *Saccharomyces cerevisiae*. *Science* 298, 799–804.

McNabb, D.S., Xing, Y., and Guarente, L. 1995. Cloning of yeast HAP5: A novel subunit of a heterotrimeric complex required for CCAAT binding. *Genes Development* 9(1), 47–58.

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

Segal, E., Barash, Y., Simon, I., Friedman, N., and Koller, D. 2002. From promoter sequence to expression: A probabilistic framework. *RECOMB*, 263–272.

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. *Nature Genet.* 34(2), 166–176.

Shea, M.A., and Ackers, G.K. 1985. The OR control system of bacteriophage lambda: A physical-chemical model for gene regulation. *J. Mol. Biol.* 181, 211–230.

Shenhar, G., and Kassir, Y. 2001. A positive regulator of mitosis, Sok2, functions as a negative regulator of meiosis in *Saccharomyces cerevisiae*. *Cellular Biol.* 21(5), 1603–1612.

Simon, I., Barnett, J., Hannet, N., Harbison, C.T., Rinaldi, N.J., *et al.* 2001. Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell* 106, 697–708.

Tanay, A., and Shamir, R. 2001. Computational expansion of genetic networks. *Bioinformatics* 17(Suppl. 1), S270–S278.

Tong, A.H., Lesage, G., Bader, G.D., Ding, H., *et al.* 2004. Global mapping of the yeast genetic interaction network. *Science* 303, 808–813.

Yeang, C.H., Ideker, T., and Jaakkola, T.S. 2004. Physical network models. *J. Comp. Biol.* 11(2–3), 243–262.

Yuh, C.H., Bolouri, H., and Davidson, E.H. 1998. Genomic cis-regulatory logic: Experimental and computational analysis of a sea urchin gene. *Science* 279, 1896–1902.

Address correspondence to:
Chen-Hsiang Yeang
Center for Biomolecular Science
University of California
Santa Cruz, CA 95064

E-mail: chyeang@soe.ucsc.edu