Modeling the combinatorial functions of multiple transcription factors

Chen-Hsiang Yeang and Tommi Jaakkola

Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA, chyeang@csail.mit.edu

Abstract. A considerable fraction of yeast gene promoters are bound by multiple transcription factors. To study the combinatorial interactions of multiple transcription factors is thus important in understanding gene regulation. In this paper, we propose a computational method to identify the co-regulated gene groups and 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 optimally fit the data. Empirical analysis indicates the inferred regulatory models agree with the known combinatorial interactions between regulators and are robust against the settings of various free parameters.

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 [1]. To model the combinatorial interactions of transcription factors, it is necessary to relate the states of activities 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. Simplification of possible regulatory programs is therefore important for modeling the combinatorial interactions of multiple transcription factors.

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 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., [2–4]) 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., [5]). Some works limited the scope to synergistic or complementary effects of regulator pairs, for example, [6] and [7]. Others attacked the combinatorial functions of multiple regulators with different computational models, such as Boolean networks [8], regression trees [9], and many others. However, since these models targeted only the functional relations of data, the resulting models can be difficult to interpret in terms of the underlying mechanisms. 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, [10]. This approach, though more reliable, is also expensive and time-consuming.

The rest of the paper is organized as follows. We will first introduce the hypotheses and concepts of our gene regulatory model and give it a mathematical definition. Following this introduction we will describe an algorithm to learn the models from binding and gene expression data. We then apply the algorithm to the CHIP-chip binding data and two large-scale gene expression datasets, and demonstrate the modeling results and their validations. Finally we will discuss the pros and cons of the method and directions of future extension.

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 ([2, 3, 11, 9]). 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 appeared in the same module) can be predicted by the same regulatory program and mRNA levels of transcription factors. For computational convenience, we also add the following assumptions. We model the relative changes of mRNA levels with respect to a reference condition and quantize those changes into three states: up-regulation, down-regulation, no change.

The key idea of our model is to characterize a regulatory program in terms of two properties of individual transcription factors. First, a transcription factor possesses a consistent function as an activator or a repressor. This function is not inverted in the context of combinatorial control. Second, a transcription factor may take 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 the responses opposite to its function. A regulator

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

is sufficient if increasing its expression level leads to the responses consistent with its function. 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 a transcription factor varies when it participates in different regulatory models. The predicted response of a regulatory program of a single regulator is uniquely determined by these two properties. Table 1 lists the predicted responses from different states of a single transcription factor.

A combinatorial function of multiple regulators gives predicted responses under each possible input state. 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 corresponding to each regulator. Briefly, each joint input state is the concatenation of the input states of single regulators. For each joint input state, the combinatorial function reports the consensus of predictions according to the input state and the two properties of each regulator. If contradiction occurs then the function reports an uncertain output. 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}. With drastic reduction of the possible functions we obtain a more tractable class that is possible to estimate 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 a basal transcription level under the reference condition, and a sufficient regulator as providing an additive enhancement or reduction 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 data of regulators and regulated genes. 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 a single regulator is uniquely determined by the function and direction of effectiveness of the regulator, as shown in Table 1. Thus at each state of multiple regulators, we can predict the output response according to the input state of each regulator. We adopt the following rules to synthesize the predicted responses from single regulators. If the predicted responses are all +1s 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 +1s and -1s, or are all 0s, then the output is 0. These rules simply report the consensus of predicted responses and output 0 if contradiction occurs. Notice we do not distinguish between the uncertain state and the state of an insignificant change under these rules. We can thus construct the combinatorial function f from Table 1 and the synthesis rules. An example of a deterministic combinatorial function of two necessary activators is shown in Table 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

The deterministic function is too rigid and does not consider the uncertainty of the regulatory program. To take uncertainty into account, we construct a probabilistic regulatory program as a conditional probability function:

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

The conditional probability 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 c_{ge} that

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	$1-\alpha$

agrees with $f(c_{Re})$ is assigned a high probability. However, when $f(c_{Re}) = 0$ each c_{ge} state is assigned an equal probability. Table 3 shows the conditional probability table, where α is a free 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) of binding and expression data according to the model. Next, we adopt a greedy algorithm to identify the models which optimize the scoring function, and evaluate the significance of the inferred model.

3.1 Likelihood function of a regulatory model

We define a log likelihood function of a regulatory model in terms of how well it fits binding and expression data. It contains two terms. The term corresponding to binding data is the log likelihood ratio between the regulatory model that each regulator binds to each regulated gene, versus 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 that the expression states of the regulators and regulated genes conform with the regulatory program, versus the null model that there is no relation between the expression states of regulators and regulated genes. The joint scoring function is the weighted sum of these two terms.

We define the following notations for the log likelihood function of binding and expression data. Denote $M = (R, G, f)$ as a regulatory model, where R and G are regulators and regulated gene sets and f the (deterministic) regulatory program. For each $r \in R$ and $g \in G$, define b_{rg} as a binary variable indicating whether r binds to g. b_{rg} is not directly observed but through a noisy measurement outcome x_{rg} from binding data. Denote E as a collection of expression experiments. For each $r \in R$ and $e \in E$, define c_{re} as the expression change of regulator r in experiment e. c_{re} is linked with a noisy measurement outcome x_{re} from microarray data. For each $g \in G$ and $e \in E$, c_{ge} and x_{ge} are defined analogously. Furthermore, denote $\{b_{rg}\}$ as a 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 a state of all $c_{re} : r \in R$ in a specific 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 $P(x_{rg}|b_{rg})$ of each pair-wise interaction reflects the confidence of binding (for example, CHIP-chip) experiments. We use an asymptotic statistic and model selection criterion to calculate the ratio $\frac{P(x_{rg}|b_{rg}=1)}{P(x_{rg}|b_{rg}=0)}$ from the measurement p-value. Details are described in [12].

We are interested in two $P(\{b_{rg}\}|H)$ priors. The only $\{b_{rg}\}$ state consistent with the regulatory model M is each factor binds to each regulated gene. Denote this hypothesis of binding states 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 a null model H_0 under which the regulators do not have any specific relation to the genes, the prior probability of $\{b_{rg}\}$ is given by

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

By applying both priors and the independence of each x_{rg}, the log likelihood ratio becomes:

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

The log likelihood ratio of expression data can be similarly constructed. The marginal likelihood function of expression data under a hypothesis H is

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

Similar to binding data, the null hypothesis of expression data assigns a uniform probability to each possible expression state $\{c_{re}\}$ 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 specifies the relation between c_{ge} and c_{Re} in each experiment e. It is specified by function f and Table 3. Each input state c_{Re} is assigned a uniform probability as in H_0.

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

The conditional probabilities $P(\{x_{re}\}|\{c_{re}\})$ and $P(\{x_{ge}\}|\{c_{ge}\})$ are again dataset-specific and independent of the regulatory model. We will discuss the choice of error models in Section 4.

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

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

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:

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

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 degrade the importance of expression data in order to make binding data relevant.

3.2 Algorithm of identifying regulatory models

We want to identify the regulatory models which optimize the joint scoring function in equation 12. This problem is difficult due to the enormous number of combinations of regulators, regulated genes and regulatory programs. We use a greedy algorithm which incrementally incorporates regulated genes and identifies the optimal regulatory program. The key steps in the algorithm are as follows.

1. Find a collection of regulator sets which co-bind to a certain number of genes according to the CHIP-chip data. The thresholds of determining significant binding events (the p-value threshold of binding data) and the number of co-bound genes are free parameters. We set $p \leq 0.005$ and regulators co-bind to ≥ 10 genes. Furthermore, we only consider the sets of ≤ 3 regulators.
2. For each candidate regulator set, identify the optimal regulated genes and regulatory programs. We are able to exhaust all possible regulatory programs due to the simplifications discussed earlier. For each regulatory program, we incrementally add genes into the regulated set, such that the log likelihood score is maximized. Since equation 12 increases with the number of regulated genes in the model, we have to specify a criterion for stopping adding genes in the set. We will describe a p-value of calculation adding a new gene in Section 3.3. We allow each gene to be assigned to 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 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.
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 optimal regulatory programs. We also evaluate the p-value of each reported direction of effectiveness. Details will be discussed in Section 3.3.

Step 2 has to be elaborated. Each regulatory program induces a different set of regulated genes. Because the log likelihood score in equation 12 grows 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 used three significant measures (p-values) in the algorithm procedures. 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 that $P(x_{ge}|c_{ge} = 0), P(x_{ge}|c_{ge} = \pm 1)$ of the newly added gene are uniformly sampled from the simplex $P(x_{ge}|c_{ge} = 0) + P(x_{ge}|c_{ge} = -1) + P(x_{ge}|c_{ge} = +1) = 1$. Rather than random samplings, the p-value under this scenario can be analytically approximated. Details about the approximation are described in the Supplementary Webpage.

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 the Supplementary Webpage.

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 re-optimizing 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 the Supplementary Webpage.

4 Empirical analysis

We applied the algorithm of identifying regulatory models to the protein-DNA interaction data of 106 transcription factors [11] and two sets of large-scale gene expression data: Rosetta Compendium data of gene knock-outs [13] and stress response gene expression data published by Gasch et al. [14]. 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.

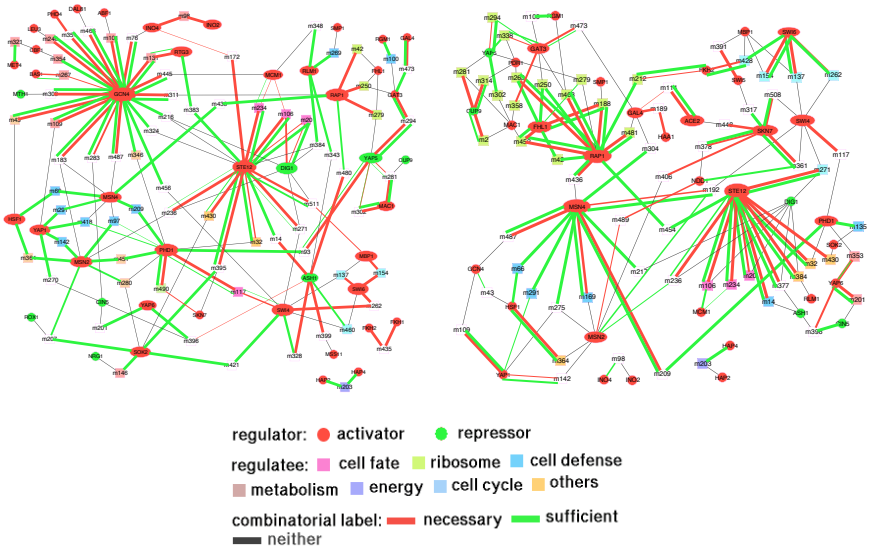
The conditional probabilities $P(\{x_{rg}\}|\{b_{rg}\})$ of binding data and $P(\{x_{re}\}|\{c_{re}\})$ and $P(\{x_{ge}\}|\{c_{ge}\})$ of Rosetta gene expression data were evaluated using the approximation described in [12]. The conditional probabilities $P(\{x_{re}\}|\{c_{re}\})$ and $P(\{x_{ge}\}|\{c_{ge}\})$ of the Gasch data were evaluated from Gaussian and exponential distributions of the time-course responses of perturbations. Details are described in the Appendix.

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, and sensitivity analysis.

4.1 Models inferred from Rosetta and Gasch data

Figure 1 summarizes the information about regulatory models inferred from Rosetta and Gasch data. We only consider the regulatory models with up to three regulators. We represent a regulatory model as a bi-partite 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

Fig. 1. Models inferred from Rosetta (left) and Gasch (right) data



in the regulated gene set ($p \leq 0.06$ according to hyper-geometric 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 4. We use the visualization software Cytoscape (www.cytoscape.org) to draw the graphs.

We found the combinatorial properties of many inferred regulatory models are 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), (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 ([15]). Protein complexes Swi6-Swi4, Swi6-Mbp1 and Fkh1-Fkh2 activate genes expressed during G1/S, S/G2 or G2/M phases of yeast cell cycle ([16]).
- 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 slightly 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) ([14]).
- 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.

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 the 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 Figure 1. The Rosetta module regulated by Gal4 (m473) is not enriched with galactose metabolic genes, and Gal80 does not appear in Figure 1. This is because the expression level of Gal4 is low even under active state [3]. 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., [3]), it does not vary significantly in both Rosetta and Gasch data.
- The combinatorial interaction of Ste12 and Dig1 on pheromone response genes is only partially retrieved. Dig1 inhibits the phosphorylation of Ste12 [17], 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 activator and repressor for different genes [18]. We assign it as an 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 performed three quantitative validations on the inferred models. First, we investigated the enrichment of functional categories in the regulated gene set according to Munich Information Center for Protein Sequences (MIPS) database (http://mips.gsf.de/) in the regulated gene sets. 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 hyper-geometric 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.

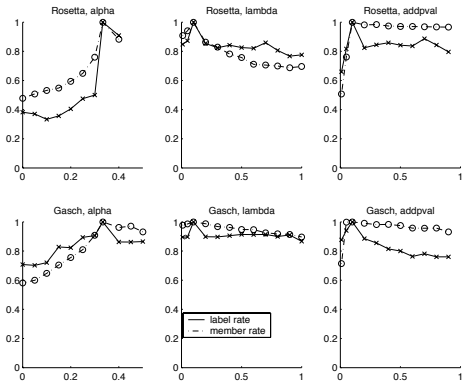
We also searched PubMed and Incyte Yeast Proteome Databases (http://www.incyte.com/login.html) to check whether regulators participating in the same model were known to jointly control one or multiple genes. More than two thirds of the regulator sets in the significant models were verified in previous 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 complete list of all regulatory models and their validations are reported in the Supplementary Webpage.

We further demonstrated the inferred models were robust against the variations of three free parameters: λ 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, p^{stop} in the greedy algorithm specifies the stopping criterion of the p-values of adding genes (Section 3.2). The default setting of these parameters is $\lambda = 0.1, \alpha = \frac{1}{3}, p^{stop} = 0.1$. 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 2 shows the sensitivity of parameters in 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 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.

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 transcrip-

Fig. 2. Robustness tests on parameters



Top: Rosetta data. Bottom: Gasch data.
Solid line: overlap of combinatorial labels.
Dash line: overlap of regulated gene sets.

tion 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 [19]. Binding and expression data alone are unlikely to capture such regulatory effects. A transcription factor can be both activator and repressor, depending on the co-factors 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: quantization of time-course expression data

In the 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 $x(t_1), \cdots, x(t_n)$ are its n time-course measurements. We relate the discrete state y to measurements $x(t_1), \cdots, x(t_n)$

with a two-level process. The discrete state y generates a continuous time-course expression profile $m(t_1), \cdots, m(t_n)$; and $x(t_1), \cdots, x(t_n)$ are noisy measurements of $m(t_1), \cdots, m(t_n)$. We model measurement errors $x(t_1) - m(t_1), \cdots, x(t_n) - m(t_n)$ as iid Gaussian random variables with zero mean and variance σ^2.

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

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

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

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

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

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

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

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

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.

Supplementary Webpage

Details about the calculations of p-values and inferred regulatory models can be found in the Supplementary Webpage http://www.csail.mit.edu/~tommi/suppl/recomb05/.

Acknowledgements

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

References

1. McNabb, D. et al. Cloning of yeast HAP5: a novel subunit of a heterotrimeric complex required for CCAAT binding. Genes Development. **9(1)** (1995) 47-58
2. Friedman, N. et al. Using Bayesian networks to analyze expression data. Journal of Computational Biology. **7** (2000) 601-620
3. Hartemink, A. et al. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. Pacific Symposium of Biocomputing. (2001) 422-433
4. Segal, E. et al. From promoter sequence to expression: a probabilistic framework. Proceedings of the 6th International Conference on Research in Computational Molecular Biology. (2002) 263-272
5. Bar-Joseph, Z. et al. Computational discovery of gene modules and regulatory networks. Nature Biotechnology. **21** (2003) 1337-1342
6. Pilpel, Y. et al. Identifying regulatory networks by combinatorial analysis of promoter elements. Nature Genetics. **29** (2001) 153-159
7. Tong, A. et al. Global mapping of the yeast genetic interaction network. Science. **303** (2004) 808-813
8. Tanay, A. et al. Computational expansion of genetic networks. Bioinformatics. **17 Suppl 1** (2001) S270-S278
9. Segal, E. et al. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nature Genetics. **34(2)** (2003) 166-176
10. Yuh, C.h. et al. Genomic cis-regulatory logic: experimental and computational analysis of a sea urchin gene. Science. **279** (1998) 1896-1902
11. Lee, T. et al. A transcriptional regulatory network map for *Saccharomyces cerevisiae*. Science. **298** (2002) 799-804
12. Yeang, C.H. et al. Physical network models. Journal of Computational Biology. **11(2-3)** (2004) 243-262
13. Hughes, T. et al. Functional discovery via a compendium of expression profiles. Cell. **102** (2000) 109-126
14. Gasch, A. et al. Genomic expression programs in the response of yeast cells to environmental changes. Molecular Biology of Cell. **11(12)** (2000) 4241-4257
15. Ambroziak, J. et al. INO2 and INO4 gene products, positive regulators of phospholipid biosynthesis in *Saccharomyces cerevisiae*, form a complex that binds to the INO1 promoter. Journal of Biological Chemistry. **269(21)** (1994) 15344-15349
16. Simon, I. et al. Serial regulation of transcriptional regulators in the yeast cell cycle. Cell. **106** (2001) 697-708
17. Bardwell, L. et al. Differential regulation of transcription: repression by unactivated mitogen-activated protein kinase Kss1 requires Dig1 and Dig2 proteins. PNAS. **95(26)** (1998) 15400-15405
18. Shenhar, G. et al. A positive regulator of mitosis, Sok2, functions as a negative regulator of meiosis in *Saccharomyces Cerevisiae*. Cellular Biology. **21(5)** (2001) 1603-1612
19. Lee, J. et al. YAP1 and SKN7 control two specialized oxidative stress response regulons in yeast. Journal of Biological Chemistry. **274(23)** (1999) 16040-16046