BIOINFORMATICS ORIGINAL PAPER

Vol. 26 no. 15 2010, pages 1879–1886
doi:10.1093/bioinformatics/btq289

Systems biology

Advance Access publication June 4, 2010

Learning transcriptional networks from the integration of ChIP–chip and expression data in a non-parametric model

Ahrim Youn[1,*], David J. Reiss[2] and Werner Stuetzle[3]

[1]National Cancer Institute, Bethesda, MD, [2]Institute for Systems Biology and [3]Department of Statistics, University of Washington, Seattle, WA, USA

Associate Editor: Martin Bishop

ABSTRACT

Results: We have developed *LeTICE* (*Le*arning *T*ranscriptional networks from the *I*ntegration of *C*hIP–chip and *E*xpression data), an algorithm for learning a transcriptional network from ChIP–chip and expression data. The network is specified by a binary matrix of transcription factor (TF)–gene interactions partitioning genes into modules and a background of genes that are not involved in the transcriptional regulation. We define a likelihood of a network, and then search for the network optimizing the likelihood.

We applied *LeTICE* to the location and expression data from yeast cells grown in rich media to learn the transcriptional network specific to the yeast cell cycle. It found 12 condition-specific TFs and 15 modules each of which is highly represented with functions related to particular phases of cell-cycle regulation.

Availability: Our algorithm is available at http://linus.nci.nih.gov/Data/YounA/LeTICE.zip

Contact: youna2@mail.nih.gov

Supplementary Information: Supplementary data are available at *Bioinformatics* online.

Received on February 17, 2010; revised on May 27, 2010; accepted on May 30, 2010

1 INTRODUCTION

The process of transcription, whereby an RNA product is produced from the DNA, is an important control point for regulating the activity of genes and their protein products in particular cell types or in response to a particular signal. The ‘circuit diagram’ of this transcriptional control process in which many transcription factors (TFs) act simultaneously and interactively to control the transcription of many genes is called a transcriptional regulatory network (TRN).

Learning a TRN from data is intrinsically difficult since most of the available data are highly condition dependent. Therefore, using data obtained from a specific condition or experiment can reveal only limited parts of the network which are active in that specific condition. As a way of inferring and understanding aspects of a transcriptional network, previous researchers (Bar-Joseph *et al.*, 2003; Lemmens *et al.*, 2006; Segal *et al.*, 2003b) introduced the concept of ‘modules’. Though the exact definition of a module varies between papers; in general, a module is a set of genes sharing a common set of regulating TFs.

The earliest attempts to learn a TRN began with an introduction of expression data since these data provide a global view of expression levels of thousands of genes at a time. Friedman *et al.* (2000) and Segal *et al.* (2003b) tried to learn a TRN using the assumption that the expression levels of genes depend on the expression levels of the TFs regulating those genes. A fundamental limitation of this approach is that expression data only measure the mRNA abundances, while it is the TF proteins that are directly involved in the regulation of genes. Therefore, the mRNA levels of the TFs may not be highly correlated with those of the genes they regulate. Second, high correlation of expression levels only provides indirect evidence for the transcriptional hierarchy of the corresponding gene regulation. For example, two genes, A and B, may be highly correlated because (i) A regulates B or (ii) B regulates A, or (iii) both are regulated by a third gene C. These three cases may not be easily distinguished using microarray data alone.

Due to these intrinsic limitations of expression data for learning TRNs, it is important to integrate other available information, such as ChIP–chip location data or DNA motif data. Location data often are presented as a matrix of P-values, with rows corresponding to genes and columns corresponding to TFs. For each TF j and gene i, the corresponding element of the matrix is the P-value for the hypothesis that there exists no interaction between TF j and the promoter region of gene i. Therefore, location data provide direct evidence for the relation between TFs and the genes they regulate by identifying physical interactions between TFs and DNA regions. However, location data also have limitations since this physical interaction may indicate binding but not function (Bar-Joseph *et al.*, 2003). Also, location data are highly condition dependent and difficult to obtain.

Another widely used data type are motif data that provide information about which potential TF binding sites exist in the promoter region of a gene. Motif data provide less direct evidence for the relation between TFs and genes than location data because motifs are merely potential binding sites which may not be bound by TFs. For this reason, we do not use motif data in our algorithm.

Because location, motif and expression data provide complementary information, many researchers have proposed methods for modeling a TRN by integrating these data types. In general, they have taken either a regression approach or a clustering approach. The regression approach taken by Bonneau *et al.* (2006), Chen *et al.* (2007), Gao *et al.* (2004) and Liao *et al.* (2003) is based on the rather strong assumption that the expression levels of TFs are correlated with the expression levels of the genes they regulate. The clustering approach followed by Bar-Joseph *et al.* (2003),

*To whom correspondence should be addressed.

Brynildsen *et al.* (2006), Lemmens *et al.* (2006), Liu *et al.* (2007) and Segal *et al.* (2003a) is based on the weaker assumption that expression levels of genes regulated by the same TFs are correlated.

Of the algorithms that use the clustering approach, the most widely cited are the GRAM algorithm (Bar-Joseph *et al.*, 2003) and the ReModiscovery algorithm (Lemmens *et al.*, 2006). GRAM integrates location and expression data, whereas ReModiscovery integrates location, expression and motif data to learn transcriptional modules. However, they share a common structure. They threshold P-value matrices representing location data and (for ReModiscovery) score matrices representing motif data with single cutoffs t_c and t_m, respectively. Then they seed the procedure by detecting modules of tightly coexpressed genes that share common subsets of TFs and motifs that exceed these thresholds. Finally, they relax the thresholds and add additional genes whose expression patterns are similar to the core expression profile of the modules.

We have developed an algorithm, which we call *LeTICE* (*Le*arning *T*ranscriptional networks from the *I*ntegration of *C*hIP–chip and *E*xpression data). It follows the clustering approach like GRAM and ReMoDiscovery. However, it differs from these algorithms in significant ways.

First, *LeTICE* defines a probabilistic model that integrates the location and expression data and fits this model by maximizing its likelihood. As a result, it simultaneously generates all modules using the entire set of TFs, and thus can identify combinatorial interactions between TFs. Most other algorithms use subsets of TFs and a P-value cutoff to build modules sequentially and then use expression data to measure the quality of each module and to adjust it. Thus, there is an asymmetry in integrating the two sources of data for those algorithms.

Second, *LeTICE* uses a non-parametric probabilistic model for the expression data, and therefore does not impose any assumptions about the distribution such as the often violated normality assumption.

Finally, *LeTICE* identifies condition specific TFs, i.e. TFs that are active in regulating genes in a given condition. Finding such condition-specific TFs is important, but it is hard to do so simply by analyzing expression data or location data separately. Many TFs actively regulating genes show constant expression profiles, and therefore identifying condition-specific TFs by variation in expression levels can result in many false negatives. Finding condition-specific TFs using only location data is not effective either. If a TF does not regulate any genes then the P-values for the TF are uniformly distributed between 0 and 1. Therefore, choosing condition-specific TFs using a P-value threshold can result in many false positives. By properly integrating expression and location data, *LeTICE* is able to model TF bindings and identify the genes regulated by these TFs on a condition-specific basis.

2 METHODS

2.1 Overview of the algorithm

We represent the TRN as a binary matrix B with rows representing genes and columns representing TFs, where $B_{ij}=1$ if TF j regulates gene i and $B_{ij}=0$ otherwise.

Figure 1 shows an overview of the algorithm. The matrix B in the center is the binding matrix that we want to determine. For that, we define the likelihood $P(B|L,E)$ of B given the location data L and the expression data E.

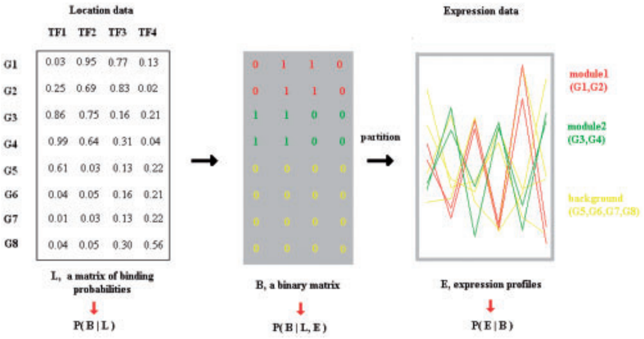

Fig. 1. Overview of the *LeTICE* algorithm. The matrix B in the center is the binding matrix that we want to determine. The left matrix L is a matrix of binding probabilities between TFs (columns) and genes (rows). From this matrix, we obtain $P(B|L)$, the likelihood of B given L. The matrix B partitions genes into modules and the background. The rows (genes) in the matrix B with the same color belong to the same module or background. Here, red and green rows form modules and yellow rows form a background. The box to the right shows expression profiles of genes, each of which are colored as in the matrix B. From this, we obtain $P(E|B)$, the likelihood of the expression profiles E given the binding matrix B.

According to Bayes rule, $P(B|L,E) \sim P(B|L)\cdot P(E|B,L) = P(B|L)\cdot P(E|B)$ since we assume that the likelihood of the expression data is independent of the location data once the binary matrix B is known.

We assume that genes either belong to modules or the background. We define a module to be a set of genes that are regulated by the same set of TFs and have similar expression profiles. We define the background as a set of genes that are not regulated by any of the investigated TFs and which do not need to show similar expression profiles. The i-th module M_i has to have a unique binding pattern b_i, which is determined to maximize $P(B_k = b_i, k \in M_i|L)$, where B_k is the k-th row of the binding matrix B (binding pattern of the gene k). Also, its gene members should have similar expression profiles that is measured by the density $P(E^{M_i}|B)$, where E^{M_i} is the expression data for genes belonging to module M_i.

Therefore, the binding matrix B can only exist in the form that generates a set of modules M_i satisfying the above condition and the background M_0 with its binding pattern $b_0 = \vec{0}$.

We define $P(B|L)$ as the product of binomial probabilities for bindings between every TF and gene: $P(B|L) = \prod_{i,j} L_{ij}^{B_{ij}} (1-L_{ij})^{1-B_{ij}}$, where $L_{ij} = P(B_{ij}=1)$. We assumed independence of individual binding events. This is only an approximation to the truth, which is necessary since there are insufficient data to estimate the dependency structure among the TF bindings. The binding probabilities L_{ij} can be obtained by converting P-values, the original form of location data (Chen *et al.*, 2007). The process of converting P-values to binding probabilities is explained in the Supplementary Material.

We define the likelihood $P(E|B)$ of expression profiles for a set of modules and the background to measure the compactness of the expression profiles of the modules. For that, we use the likelihood of expression profiles under a mixture model. We assume expression profiles of genes in the same module or background are generated from the same density. The difficulty in choosing the density estimate for the mixture components arises because modules and a background have very different distributions. To fit two very different kinds of distributions equally well, we chose a kernel density estimate since kernel density estimates do not require any assumptions about the distribution of the data, and therefore are very flexible. The detailed description of the likelihood $P(E|B)$ is provided in Section 2.2.

In summary, the binding matrix B which generates a set of modules and background has the likelihood $P(B|L,E) \sim P(B|L)\cdot P(E|B)$, where $P(B|L)$ measures how likely genes in modules or background have the same binding pattern and $P(E|B)$ measures how similar the expression profiles of genes in each module are. Using numerical optimization, *LeTICE* finds the binding matrix B which maximizes $P(B|L)\cdot P(E|B)$. The optimization process is described in Section 2.4.

2.2 Defining the likelihood for a set of modules and a background

A given binary binding matrix B results in a partition of genes into modules and a background. We define modules as subsets of size at least five regulated by at least one TF, whose members have similar expression profiles. To measure the compactness of modules, we use the likelihood of expression profiles under a mixture model. Let $\hat{f}_k(E_i|B)$ be the estimated density for the expression profiles of the i-th gene in the k-th module or the background if $k=0$. Then, the likelihood $P(E|B)$ of expression data given the binding matrix is

$$P(E|B)=\prod_{k=0}^{G(B)}\prod_{i\in M_k}\hat{f}_k(E_i|B), \qquad (1)$$

where $G(B)$ is the total number of modules generated by the binary matrix B, M_k is the set of indices of genes belonging to the k-th component, and E_i is the expression profile of the i-th gene.

When we estimate the density of the i-th gene, $\hat{f}_k(E_i|B)$, we leave out the i-th gene. That is, the density of the i-th gene is estimated from other genes belonging to the same subset. If we include the expression level of the gene itself when estimating its density, genes belonging to small subsets will get higher density due to overfitting.

A key issue is the choice of density estimate for the mixture components $\hat{f}_k$. We can use either a parametric or a non-parametric estimate. If we use a parametric estimate, we need to make a strong assumption about the distribution of the data. A Gaussian density may be a reasonable model for the expression profiles of genes in a module. However, the assumption of a Gaussian distribution does not make much sense for the greater number of genes belonging to the background. We could use a different parametric model for the background, but using different models may induce a bias in the assignment of genes to modules or background. If the model for the background is too inflexible to fit the data, then the optimization process will find a binary matrix B generating a set of modules such that the size of the background is small. On the other hand, if the background model fits the data too well, the process will find a binary matrix such that the size of the background is large. To avoid this kind of bias, we want to use the same type of model for both background and modules. This model needs to be flexible enough to fit two very different kinds of distributions. Therefore, we use kernel density estimates to model the distributions of expression levels for modules and a background, leading to a likelihood that is similar to the criterion for kernel k-means clustering (Dhillon *et al.*, 2004).

2.2.1 *Kernel density estimates*

We standardize expression profiles of all genes to have mean 0 and length 1. Therefore, expression profiles are points on the unit sphere in N dimensions, where N is the length of an expression profile. The distance between points x and y on the sphere is represented by an angle x^Ty, which is same with a Pearson's correlation coefficient between two points.

Let $X_1,\cdots X_n$ be points on the sphere assumed to be generated from some unknown density. A kernel density estimate at a point x on the sphere is defined as

$$\hat{f}(x;\kappa)=\frac{1}{n}c_0(\kappa)\sum_{i=1}^{n}K(\kappa x^TX_i),$$

where κ is a smoothing parameter and $c_0(\kappa)$ is a normalizing constant (Hall *et al.*, 1987). The most commonly used kernel is $K(t)=e^t$. Since x^TX_i is the Pearson's correlation coefficient between x and X_i, the density at x is an increasing function of the correlation between x and the other members of the subset.

The smoothing parameter κ controls the smoothness of the estimate; the smaller κ, the smoother the estimate. An important issue in kernel density estimation is the choice of smoothing parameter κ. To avoid bias when assigning genes to modules or background, we use the same smoothing parameter for all the mixture components. Estimating a smoothing parameter for density estimates in high-dimensional data is a challenging problem for which no good solution is known. We first tried to find κ that maximizes the total likelihood. However, this resulted in a very large value of κ and essentially no smoothing. Therefore, we use a heuristic for determining κ, we choose κ such that on average 90% of the sum of kernel values for each gene is contributed by $m=10$ other genes. In our simulation study (Supplementary Material), we tried several values for m on simulated data and found that $m=10$ gave the best result.

2.3 Pre-selecting transcriptionally active genes and TFs

Screening out genes and TFs that do not appear to be part of the condition-specific network under study reduces the the number of parameters to be estimated by the algorithm and their variability. Brynildsen *et al.* (2006) also suggested that due to the high degree of inconsistency between expression and location data, it is necessary to select genes whose expression and location data are mutually consistent. Of course, the benefits of pre-selection come at a cost—we may occasionally miss relevant genes and TFs because they do not meet the selection criteria.

2.3.1 *Gene selection*

We expect that genes that are part of condition-specific networks show variation in their expression patterns. Therefore, We exclude genes with constant expression levels from further consideration. There are various statistical methods available to identify genes that are differentially expressed over time or experiments (Simon and Lam, 2006; Storey *et al.*, 2005). For the specific case of cell-cycle data, there are statistics measuring the periodicity of expression levels (Orlando *et al.*, 2008; Spellman *et al.*, 1998). If the purpose of the analysis is to find the TRN controlling the cell-cycle network, these measures may be more appropriate than omnibus measures since cell-cycle genes show periodic expression patterns.

2.3.2 *TF selection*

TFs whose bindings affect the transcription of the genes are considered to be transcriptionally active. Some TFs bind to the upstream regions of genes without affecting transcription of those genes. For these TFs, their binding patterns are independent of the correlation patterns of expression levels of genes and we filter out these TFs.

For a TF to regulate expression of some genes, the TF must first bind those genes, and thus binding probabilities between the TF and the regulated genes should be large. Therefore, a group $C_j=\{i|P(B_{ij}=1|L)>0.5\}$ will contain most of the genes regulated by TF j. The genes regulated by TF j will belong to some modules, thus some pairs of those genes will have high correlations. Therefore, if a TF j regulates expression of some genes, there will be highly correlated pairs of genes within the group C_j. In other words, the set of correlations between genes in C_j will contain a higher proportion of large values compared to the set of correlations of randomly chosen genes.

We calculate the 95th percentile ρ of the Pearson's correlation coefficients between all pairs of genes. We then calculate the proportion of correlations higher than ρ in the set of correlation coefficients between genes in C_j and define a non-parametric P-value p_j by comparing it with the proportions of correlations higher than ρ from a random group with same size as C_j.

If p_j is small, then the proportion of large correlations between genes in C_j is significantly higher than the proportion in random groups, and we conclude that TF j regulates some genes. We validate the performance of this TF pre-selection step in our simulation study (see Supplementary Material).

After TF pre-selection, we remove all genes that have low binding probabilities (<0.1) for all the pre-selected TFs, since they will probably be assigned to the background by the optimization process. By removing these genes in advance, we reduce the background size.

2.4 Optimization of the objective function

Our goal is to find a binding matrix B, which maximizes the likelihood $P(B|E,L)$. This is a combinatorial optimization problem, and finding a global optimum is impossible. Our optimization algorithm consists of three stages: initialization, randomized greedy ascent and post-processing. Post-processing is deterministic, but the local optimum found by randomized greedy ascent depends on the initial number N of modules and their initial binding patterns, and the steps that happen to be chosen in the ascent. We run the entire algorithm repeatedly (1000 times) for different values of N and report the best solution. A detailed description of the optimization process is provided in the Supplementary Material.

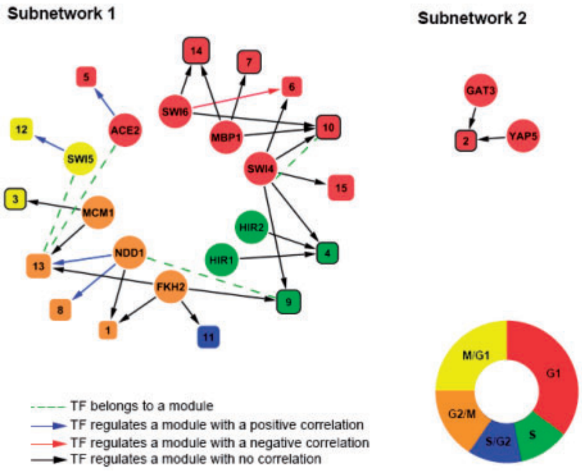
Subnetwork 1
Subnetwork 2
14
7
5
6
SWI6
ACE2
12
MBP1
10
SWI5
SWI4
15
3
MCM1
HIR2
NDD1
4
HIR1
13
FKH2
9
8
1
11
GAT3
2
YAP5
M/G1
G1
G2/M
S/G2
S
TF belongs to a module
TF regulates a module with a positive correlation
TF regulates a module with a negative correlation
TF regulates a module with no correlation

Fig. 2. Graph of TRN built by *LeTICE*. Rectangles represent gene modules and circles represent TFs. Each gene module is given a unique integer and the gene modules having GO $P < 0.05$ have thick black border lines. There is an arrowed edge between a TF and a gene module if the TF binds the genes in the module, whereas there is a dotted edge if the TF itself belongs to the module. The arrowed edges are colored according to the correlation between expression levels of a TF and genes: if significantly positive, blue colored, if negative, red colored. The other edges are black colored. Each module and TF is colored according to the cell-cycle phase (G1, G2/M, M/G1, S and S/G2) they belong to.

3 RESULTS

3.1 Application to the Yeast cell-cycle data

3.1.1 Data

- Expression data :
  We use the cell-cycle expression data of Cho *et al.* (1998), Orlando *et al.* (2008) and Spellman *et al.* (1998): two replicate series from Orlando *et al.* (2008) obtained using elutriation, one series from Spellman *et al.* (1998) obtained using arrest of alpha factor and one series from Cho *et al.* (1998) obtained using cdc28 temperature-sensitive mutant. They were all obtained in rich media. We use 510 genes that are selected to have periodic expression patterns by the periodicity measures of both Orlando *et al.* (2008) and Spellman *et al.* (1998). We calculated $P(E^i|B)$ separately for each subseries i and multiplied them together to obtain the total likelihood $P(E|B)$.
- Location data :
  We use the data from Lee *et al.* (2002) obtained for yeast cells grown in a rich medium. They provide P-values for testing binding between 113 TFs and 6270 genes.

3.1.2 Cell-cycle TRN The TRN found by *LeTICE* is shown in Figure 2. Rectangles represent gene modules and circles represent TFs. *LeTICE* found 12 condition-specific TFs out of 113 TFs : ACE2, FKH2, GAT3, HIR1, HIR2, MBP1, MCM1, NDD1, SWI4, SWI5, SWI6 and YAP5. These include TFs that are well known to play key roles in cell-cycle regulation: ACE2, FKH2, MBP1, MCM1, NDD1, SWI4, SWI5 and SWI6. We assign a unique integer to each module and the size of rectangles represent the size of modules. Modules with global GO P-value (which will be explained later in this chapter) less than 0.05 have bold border lines. There are 15 modules containing 5, 6, 7, 7, 9, 9, 12, 13, 13, 14, 14, 15, 19, 19 and 23 genes, respectively, for a total of 185 genes.

There are two different types of edges : arrowed and dotted. An arrowed edge between a TF and a module means that the TF regulates the genes belonging to the module, whereas a dotted edge means that the TF itself belongs to the module. For each arrowed edge linking a TF to a module, we calculated the non-parametric P-value for the correlation between the TF and the mean expression level of the module by comparing it with correlations between all 113 TFs and the mean expression levels of all the modules (Bar-Joseph *et al.*, 2003). The P-value for the correlation is the fraction of all the correlations whose absolute value is larger than the absolute value of the correlation associated with the edge in question. Edges between a TF and a gene module whose correlation is positive with $P < 0.05$ are colored blue and edges whose correlation is negative with $P < 0.05$ are colored red. The other edges are colored black. As mentioned in Section 1, *LeTICE* does not use the assumption that the expression levels of the TFs and their regulated genes are correlated since this assumption fails for many TFs as we can see in Figure 2. There are many black edges in Figure 2 and if we used the assumption, these edges could not be detected.

Spellman *et al.* (1998) classified genes involved in the cell cycle into the five phase groups G1, G2/M, M/G1, S and S/G2 according to their times of peak expression. We used the classification of Spellman *et al.* (1998) to assign each module and TF to the most represented phases of the set of genes belonging to the same module, or those regulated by the same TF, respectively. We also obtain P-values of the phases by measuring how frequently the phase appears in the set compared to the entire set of genes using hypergeometric distribution. All modules and TFs detected by *LeTICE* are assigned to phases with $P < 0.05$ and they are colored according to their assigned phases in Figure 2.

By assigning each TF to a phase according to the phases of their regulated genes, we make two interesting observations. First, the phase assigned by the phases of its regulated genes and the phase by its time of peak expression from Spellman *et al.* (1998) are different for every TF. While all the condition-specific TFs were assigned to a phase according to the phases of their regulated genes, many of them were not assigned to any phases according to their times of peak expression by Spellman *et al.* (1998) due to their constant expression levels. Of the 12 condition-specific TFs, only 5 TFs were assigned to phases by their times of peak expression: ACE2 to G2/M, GAT3 to M/G1, NDD1 to G1, SWI4 to M/G1 and SWI5 to G2/M, whereas ACE2 was assigned to G1, GAT3 to G1, NDD1 to G2/M, SWI4 to G1 and SWI5 to M/G1 by the phases of the regulated genes. This implies that there are various time lags between the times of peak expression of TFs and the times of peak expression of their regulated genes. Again, this observation calls in question the

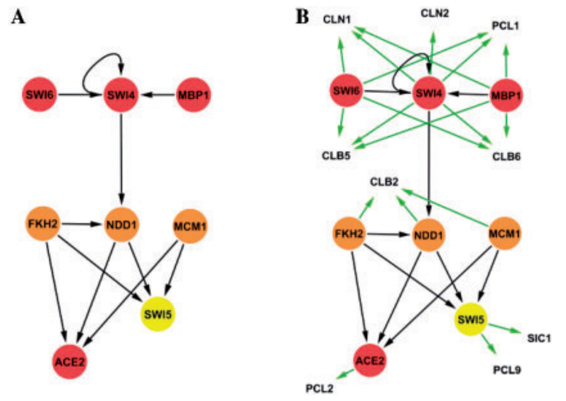

Fig. 3. TF network extracted from TRN. (**A**) TF network consisting only of TFs, which was generated from Figure 2 by deleting all genes that are not the condition-specific TFs and then deleting isolated TFs. (**B**) TF network generated by adding cyclins, which *LeTICE* found to be regulated by the TFs in Figure 3A.

assumption that the expression levels of the TFs and their regulated genes are correlated.

Second, all TFs regulate genes belonging to either the same phase or to adjacent phases.

The complete list of genes, which belongs to each module and of TFs regulating each module, and detailed description of the modules are provided in the Supplementary Material.

3.1.3 Extraction and interpretation of the TF subnetwork As shown in Figure 2, some TFs regulate other TFs which control transcription in the next phase of the cell cycle. From Figure 2, we generated the TF network in Figure 3A consisting only of TFs by first deleting all genes, which are not condition-specific TFs and then deleting isolated TFs. It contains TFs MBP1, SWI4/5/6, NDD1, MCM1, FKH2 and ACE2, all known to play key roles in regulating the cell cycle. It is almost identical to the previously suggested cell-cycle transcriptional network (Simon *et al.*, 2001), which is interesting because it was generated *de novo* using all 113 TFs, whereas the previously suggested TF network was constructed using the location data of only nine known cell-cycle regulatory TFs. This implies that on the transcriptional level, this small TF network controls the whole cell-cycle regulatory network.

The G1-specific TF complexes SWI4/SWI6/MBP1 regulate the G2/M-specific TF NDD1. NDD1 is a limiting component of the complex MCM1/FKH2/NDD1 that regulates M/G1-specific TFs SWI5 and ACE2. This serial regulation of TFs is linear rather than cyclic. Cyclic regulation of TFs is achieved by cyclin-dependent kinases (CDKs), which bind with cyclins to be activated.

To better understand the cyclic flow in the network, we added cyclins and CDK inhibitors which *LeTICE* found to be regulated by the TFs in Figure 3A to generate Figure 3B.

In yeast, the cell cycle is regulated by two CDKs, CDC28 and PHO85. CDC28 associates with cyclins CLN1/2 and CLB2/5/6 and PHO85 associates with cyclins PCL1/2/9.

It is known that CDC28 associated with CLN3 and PHO85 associated with PCL9 phosphorylate the repressor of SBF (the pair of SWI4 and SWI6) and MBF (the pair of SWI6 and MBP1) to promote its dissociation from SBF and MBF (Huang *et al.*, 2009). SBF and MBF then start transcription of G1/S cyclins CLN1/CLN2, which in turn associate with CDC28 to further phosphorylate the repressor of SBF and MBF leading to its complete dissociation from SBF and MBF, forming a positive feedback loop. PCL1, regulated by SBF and MBF and PCL2, regulated by ACE2 are known to have redundant function with CLN1 and CLN2. SBF and MBF also activate CLB5/CLB6, which associate with CDC28 to play important roles for entry into and progression through the S phase.

NDD1 activated by SBF and MBF combine with MCM1 to activate CLB2, which inactivates SBF and MBF, helping to end the G1/S program and start the G2/M program (Amon *et al.*, 1993; Spellman *et al.*, 1998).

The TF SWI5 regulated by MCM1, NDD1 and FKH2 transcribes SIC1 and PCL9. Expression of SIC1 helps to inactivate CLB2 activity, helping to simultaneously end the M phase and the repression of the G1/S phase. PCL9–PHO85 and CLN3–CDC28 activate SBF and MBF, starting a new round of cell cycle (Futcher, 2002).

We compared the network in Figure 3B with the one built by Simon *et al.* (2001) using the location data of nine cell-cycle TFs. The biggest difference between the two networks is that our network does not contain the TF FKH1. FKH1 was not selected as a condition-specific TF by *LeTICE* because the genes regulated by FKH1 do not show high correlation with each other. Also, FKH1 do not have high binding probabilities with any condition-specific TFs.

In addition to that, the cyclin CLB1/CLN3 and the CDK inhibitor FAR1 were not assigned to any TFs. It is not surprising that CLB1 and FAR1 were not assigned to any TFs since their binding probabilities to any condition-specific TFs are very low (at most 0.01 for CLB1 and 0.27 for FAR1). The cyclin CLN3 has a high binding probability to one TF, MCM1 (0.99), but the correlation of the expression pattern of CLN3 with other genes regulated by MCM1 is not high. In fact, this irregular pattern of CLN3 was also pointed out by Wittenberg *et al.* (1990) and Tyers *et al.* (1992) who show that although CLNl and CLN2 are periodically transcribed with a peak at the late G1, CLN3 is expressed throughout the cell cycle and shows less fluctuation in mRNA and protein levels.

3.2 Validation with Gene Ontology annotation

We want to validate the TRN which *LeTICE* found. However, the ground truth about the network is not known. As a substitute, we measure the internal homogeneity of modules with respect to Gene Ontology (GO) annotation. For this purpose, we use the GO induced similarity from the Bioconductor function SimUI within biological process ontology (Wolting *et al.*, 2006).

SimUI uses the graph induced by GO terms associated with each gene to calculate the similarity between any pair of genes. It is the number of nodes that the two graphs have in common divided by the number of nodes in the union of the two graphs. Hence the similarities are bounded between 0 and 1 and more similar gene pairs have values closer to one (Wolting *et al.* (2006)).

For each module, we calculate the mean of GO similarities between all pairs of genes in the module. Then, we calculate the non-parametric *P*-value of the mean GO similarity (which we will call GO *P*-value from now on) by comparing the mean GO similarity with a hundred thousand other mean GO similarities obtained from random modules of the same size using the same genes. Since GO *P*-values are obtained by comparing mean GO similarities with those

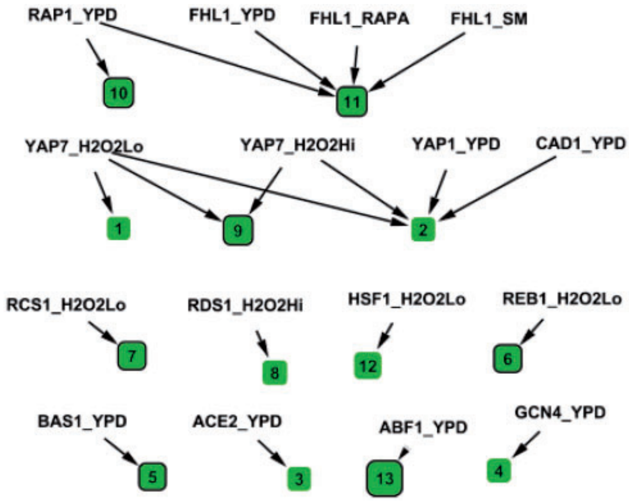

Fig. 4. Graph of TRN in the condition of H_2O_2 treatment built by *LeTICE*. Rectangles represent gene modules. Modules with global GO p-value less than 0.05 have bold border lines. TFs are represented by their names and the conditions connected by a dash. 'YPD' represents rich media, 'SM', amino acid starvation, 'RAPA', nutrient deprivation, 'H2O2Hi', high level H_2O_2 treatment, 'H2O2Lo', moderate level H_2O_2 treatment.

from random modules of the same size using the same genes, they are not biased with regard to module size and are not biased due to the gene pre-selection. We use the fraction of modules with GO $P < 0.05$ as a summary measure of performance of an algorithm. *LeTICE* generated 15 modules and 7 of them (46.7%) have GO $P < 0.05$. For a random set of modules, only 5% of them would have GO $P < 0.05$. Therefore, this analysis supports the claim that *LeTICE* finds biologically meaningful results.

3.3 Application to heterogeneous datasets

LeTICE can be expected to work best when both location data and expression data are obtained under the same conditions, since both gene expression patterns and TF binding patterns are highly condition dependent. However, we will illustrate that even when *LeTICE* is applied to heterogeneous data, it can still find modules regulated by TFs whose binding patterns match the gene transcription patterns. For this purpose, we apply *LeTICE* to the expression data from Gasch *et al.* (2000) and the location data from Harbison *et al.* (2004).

Gasch *et al.* (2000) measured changes in the expression levels of 6200 yeast genes over time under several different conditions. We select the time series obtained when yeast was treated with hydrogen peroxide (H_2O_2) since it is one of the longest time series. We use the BRB array tool developed by Simon and Lam (2006) to pre-select 913 genes that are differentially expressed over time.

Harbison *et al.* (2004) obtained location data for 203 TFs in rich media conditions. Of these 203 TFs, 84 were also profiled in at least 1 of 12 specific environmental conditions in which they were known to be essential for growth. There were a total of 352 combinations of TFs and conditions. Two of the 12 conditions match the condition under which expression data were obtained: high level of H_2O_2 treatment and moderate level of H_2O_2 treatment.

The TRN found by *LeTICE* is shown in Figure 4. Rectangles represent gene modules. Seven of the 13 modules (53.8%) drawn with bold border lines have GO $P < 0.05$. TF-condition pairs are represented by the TF names and the conditions connected by a dash. 'YPD' represents rich media, 'SM', amino acid starvation, 'RAPA', nutrient deprivation, 'H2O2Hi', high level H_2O_2 treatment and 'H2O2Lo', moderate level H_2O_2 treatment.

The TFs in the network found by *LeTICE* are completely different from the cell-cycle TFs in Figure 2. For all active TFs with the exception of FHL1, only binding patterns observed under rich media or H_2O_2 treatment are correlated with expression levels. Module 11 is regulated by FHL1 under conditions 'YPD', 'RAPA' and 'SM'. The bindings of FHL1 appear not to be condition-specific for genes in Module 11. The five TFs in the network profiled under H_2O_2 treatment are known to be essential for growth or implicated in gene regulation in the condition of H_2O_2 treatment (Harbison *et al.*, 2004). This again supports that *LeTICE* can identify the right condition-specific TFs.

This result suggests that *LeTICE* can identify condition-specific TFs and can learn a meaningful result even when the condition of the expression and location data are different.

3.4 Benefits from integrating location and expression data

We investigated the benefit of integrating location and expression data compared to using only one of the two data types.

First, we generated modules from expression data only. We used partitioning around medoids (Kaufman and Rousseeuw, 1990) to cluster the 510 genes used in our analysis into 41 subsets so that the average size of the subsets is similar to the average size of modules found by *LeTICE*. We define subsets containing at least five genes as modules. The percentage of modules having GO $P < 0.05$ is 26.7%.

Second, we generated modules by using only location data with a fixed P-value cutoff p_0. We set $B_{ij} = 1$ if the P-value between gene i and TF j is less than p_0, and $B_{ij} = 0$ otherwise. A set of at least five genes having the same binding pattern is defined as a module. For each TF j for which the number of genes with $B_{ij} = 1$ is less than five, we zeroed out those B_{ij} because we restrict the module size to be at least five, which is equivalent to requiring that every TF binds at least five genes if it binds any.

For the frequently used cutoff $p_0 = 0.005$, there are 54 TFs, each of which regulates at least five genes. These TFs include the 12 condition-specific TFs *LeTICE* found. While most of the 12 TFs have key roles in cell-cycle regulation, the remaining 42 TFs do not, which shows that *LeTICE* finds better candidates for condition-specific TFs than using the fixed P-value cutoff for location data.

Since there are many TFs regulating genes, many small subsets of genes having the same binding pattern are generated. Only 11 of them have size at least five. Of these modules, the percentage with GO $P < 0.05$ is 18.2%.

These results show that the integration of location and expression data does help to find better modules and better candidates for condition-specific TFs.

4 DISCUSSION

We developed *LeTICE*, an algorithm for building a TRN by integrating expression and location data. The TRN is defined by a binding matrix B, which partitions genes into modules and a background. We defined a likelihood $P(B|E,L)$ and then find the matrix B maximizing $P(B|E,L)$ through an optimization process.

The contribution of *LeTICE* is 2-fold: first, it produces biologically meaningful results and second, it is methodologically innovative.

4.1 Biological results

- *LeTICE* found 12 condition-specific TFs including key cell-cycle regulators ACE2, FKH2, MBP1, MCM1, NDD1, SWI4, SWI5 and SWI6.
- *LeTICE* generated 15 modules each of which is highly represented with functions related to the cell-cycle regulation such as DNA synthesis, repair, compacting newly synthesized DNA, cytokinesis, cell wall construction and DNA pre-replication complex formation. The internal homogeneity of the modules as measured by GO similarity is superior to that of modules found by using only expression data or location data.
- *LeTICE* generated a TF network containing TFs MBP1, SWI4/5/6, NDD1, MCM1, FKH2 and ACE2 known to play key roles in regulating the cell cycle. Interestingly, it is almost identical to the previously derived cell-cycle transcriptional network (Simon *et al.*, 2001) although it was generated automatically using all 113 TFs, while the previously derived TF network was generated using only nine key cell-cycle regulators.

4.2 Methodological innovations

- *LeTICE* uses a probabilistic model for the binary binding matrix given the location and expression data. It integrates two heterogeneous data sources in a principled way without requiring arbitrary weights or thresholds.
- *LeTICE* does not require any assumptions about the distribution of the expression data since it uses a non-parametric model. It does not rely on the questionable assumption that the expression levels of TFs and regulated genes are correlated. It only makes the weaker assumption that genes regulated by the same TFs tend to show similar expression profiles.
- *LeTICE* generates all gene modules simultaneously using the entire set of TFs, while other algorithms (Bar-Joseph *et al.*, 2003; Lemmens *et al.*, 2006) find gene modules sequentially using subsets of TFs. It can cope with combinatorial interactions of TFs and is able to find condition-specific TFs.

4.3 Future work

Although location data provide the most accurate information about physical bindings between TFs and genes, due to their high condition dependency, they can only reveal parts of the TRN active in the given specific condition.

In contrast to location data, motif data are abundant and condition independent. Motif data contain ample information from lots of research and especially for yeast, many TFs have known target binding motifs. Motif data and location data are essentially about the same thing: TF binding. In principle, our probabilistic approach to integrating location and expression data should also be applicable to motif data. The problem with motif data is that they contain many more false positives since motifs are merely potential binding sites, which may not be bound by TFs. Therefore, the probability distribution of B given motif occurrences needs to take into account of high false positive rates of motifs, whereas the non-parametric likelihood for measuring compactness of expression levels of genes in modules should still be applicable.

ACKNOWLEDGEMENTS

We thank Thomas Richardson for helpful discussions and also thank Guang Chen, Shane Jensen and Christian Stoeckert for kindly sharing the unpublished data. A.Y. conceived the study, implemented the algorithm and prepared the manuscript. W.S. provided the guidance during the study and helped revise the manuscript. D.R. contributed to GO similarity calculations and helped revision. All authors read and approved the final manuscript.

Funding: NSF grant DMS-9803226.

Conflict of Interest: none declared.

REFERENCES

Amon,A. *et al.* (1993) Mechanisms that help the yeast cell cycle clock tick: G2 cyclins transcriptionally activate G2 cyclins and repress G1 cyclins. *Cell*, **74**, 993–1007.

Bar-Joseph,Z. *et al.* (2003) Computational discovery of gene modules and regulatory networks. *Nat. Biotechnol.*, **21**, 1337–1342.

Bonneau,R. *et al.* (2006) The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biol.*, **7**, R36.

Brynildsen,M.P. *et al.* (2006) A Gibbs sampler for the identification of gene expression and network connectivity consistency. *Bioinformatics*, **22**, 3040–3046.

Chen,G. *et al.* (2007) Clustering of genes into regulons using integrated modeling-COGRIM. *Genome Biol.*, **8**, R4.

Cho,R.J. *et al.* (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell.*, **2**, 65–73.

Dhillon,I.S. *et al.* (2004) Kernel k-means: spectral clustering and normalized cuts. In *KDD '04: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, pp. 551–556,

Friedman,N. *et al.* (2000) Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, **7**, 601–620.

Futcher,B. (2002) Transcriptional regulatory networks and the yeast cell cycle. *Curr. Opin. Cell Biol.*, **14**, 676–683.

Gao,F. *et al.* (2004) Defining transcriptional networks through integrative modeling of mRNA expression and transcription factor binding data. *BMC Bioinformatics*, **5**, 31.

Gasch,A.P. *et al.* (2000) Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell*, **11**, 4241–4257.

Hall,P. *et al.* (1987) Kernel density estimation with spherical data. *Biometrika*, **74**, 751–762.

Harbison,C.T. *et al.* (2004) Transcriptional regulatory code of a eukaryotic genome. *Nature*, **431**, 99–104.

Huang,D. *et al.* (2009) Dual regulation by pairs of cyclin-dependent protein kinases and histone deacetylases controls G1 transcription in budding yeast. *PLoS Biol.*, **7**, e1000188.

Kaufman,L. and Rousseeuw,P. J. (1990) *Finding Groups in Data. An Introduction to Cluster Analysis. Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics*. Wiley, New York.

Lee,T.I. *et al.* (2002) Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science*, **298**, 799–804.

Lemmens,K. *et al.* (2006) Inferring transcriptional modules from ChIP-chip, motif and microarray data. *Genome Biol.*, **7**, R37.

Liao,J.C. *et al.* (2003) Network component analysis: reconstruction of regulatory signals in biological systems. *Proc. Natl Acad. Sci. USA*, **100**, 15522–15527.

Liu,X. *et al.* (2007) Bayesian hierarchical model for transcriptional module discovery by jointly modeling gene expression and ChIP-chip data. *BMC Bioinformatics*, **8**, 283.

Orlando,D.A. *et al.* (2008) Global control of cell-cycle transcription by coupled CDK and network oscillators. *Nature*, **453**, 944–947.

Segal,E. *et al*. (2003a) Genome-wide discovery of transcriptional modules from DNA sequence and gene expression. *Bioinformatics*, **19** (Suppl. 1), i273–i282.

Segal,E. *et al*. (2003b) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet*., **34**, 166–176.

Simon,I. *et al*. (2001) Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell*, **106**, 697–708.

Simon,R. and Lam,A.P. (2006) *BRB Array Tools Users Guide*. Biometric Research Branch, National Cancer Institute.

Spellman,P.T. *et al*. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, **9**, 3273–3297.

Storey,J.D. *et al*. (2005) Significance analysis of time course microarray experiments. *Proc. Natl Acad. Sci. USA*, **102**, 12837–12842.

Tyers,M. *et al*. (1992) The Cln3-Cdc28 kinase complex of S. cerevisiae is regulated by proteolysis and phosphorylation. *EMBO J*., **11**, 1773–1784.

Wittenberg,C. *et al*. (1990) G1-specific cyclins of S. cerevisiae: cell cycle periodicity, regulation by mating pheromone, and association with the p34CDC28 protein kinase. *Cell*, **62**, 225–237.

Wolting,C. *et al*. (2006) Cluster analysis of protein array results via similarity of Gene Ontology annotation. *BMC Bioinformatics*, **7**, 338.