BIOINFORMATICS ORIGINAL PAPER

Vol. 23 no. 6 2007, pages 755–763
doi:10.1093/bioinformatics/btl676

Systems biology

Inferring pairwise regulatory relationships from multiple time series datasets

Yanxin Shi[1,2], Tom Mitchell[1,3] and Ziv Bar-Joseph[1,3,4,*]

[1]Machine Learning Department, [2]Language Technologies Institute, [3]Computer Science Department and [4]Department of Biological Sciences, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA

Received on September 13, 2006; revised on December 18, 2006; accepted on January 4, 2007

Advance Access publication January 19, 2007

Associate Editor: Thomas Lengauer

ABSTRACT

Motivation: Time series expression experiments have emerged as a popular method for studying a wide range of biological systems under a variety of conditions. One advantage of such data is the ability to infer regulatory relationships using time lag analysis. However, such analysis in a single experiment may result in many false positives due to the small number of time points and the large number of genes. Extending these methods to simultaneously analyze several time series datasets is challenging since under different experimental conditions biological systems may behave faster or slower making it hard to rely on the actual duration of the experiment.

Results: We present a new computational model and an associated algorithm to address the problem of inferring time-lagged regulatory relationships from multiple time series expression experiments with varying (unknown) time-scales. Our proposed algorithm uses a set of known interacting pairs to compute a temporal transformation between every two datasets. Using this temporal transformation we search for new interacting pairs. As we show, our method achieves a much lower false-positive rate compared to previous methods that use time series expression data for pairwise regulatory relationship discovery. Some of the new predictions made by our method can be verified using other high throughput data sources and functional annotation databases.

Availability: Matlab implementation is available from the supporting website: http://www.cs.cmu.edu/~yanxins/regulation_inference/index.html

Contact: zivbj@cs.cmu.edu

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

1 INTRODUCTION

There are two primary sources for regulation inference from gene expression data. The first is perturbation experiments which usually knock out a gene and study the downstream effects (Hughes *et al.*, 2000; Peer *et al.*, 2001; Yeang and Jaakkola, 2003). The second is time series experiments in which researchers use time-lagged correlations to search for regulatory relationships (Balasubramaniyan *et al.*, 2005; Qian *et al.*, 2001; Schmitt *et al.*, 2004). While perturbation experiments can identify direct regulators of the affected gene, the use of such experiments is limited due to technical and biological reasons [for example, 20% of the yeast genes are essential (Tong *et al.*, 2001)].

Time series expression data has been used to study a wide range of biological systems in many different species (Bar-Joseph, 2004). This type of data now accounts for over 40% of publicly available expression datasets (Ernst *et al.*, 2005). Unlike perturbation experiments which usually start with a single perturbed gene, time series data implies a number of regulatory interactions. Current methods using time series expression data for inferring time-lagged regulatory relationships focus on a *single* dataset in which the lag is assumed to be stationary. However, the application of these methods to a dataset containing measurements of thousands of genes over a relatively small number of time points leads to a large number of false positives. In such a dataset, many of the inferred regulatory relationships may result from noise or from unrelated sources (co-occurrence as opposed to activation, see Section 3).

A possible solution for this problem is to combine different datasets (measuring the same set of genes under different experimental conditions) and search for regulatory relationships that are present in a subset of these datasets. However, combining multiple datasets for this task is a non-trivial problem. First, different pairs of genes usually have different lags, even in the same dataset. For example, the lags may depend on affinity properties of transcription factors (TFs). Second, for a given pair, the actual time lag may differ between different experiments since the *timescale* of the series data may change. For example, using different arrest methods leads to very different cell cycle durations in yeast (Spellman *et al.*, 1998). These different cell cycle durations translate to differences on the molecular level which affect the time it takes a TF to activate the genes it regulates (Aach and Church, 2001). Third, even for a pair of genes displaying time-lagged regulation this relationship might exist in only a subset of the datasets. For example, different pathways may be activated under different conditions.

In some cases (for example, when studying just the cell cycle) the differences in dynamics can be dealt with by looking for an alignment between the experiments assuming a common expression pattern for genes in all experiments. However, when combining more diverse experiments (for example, cell

*To whom correspondence should be addressed.

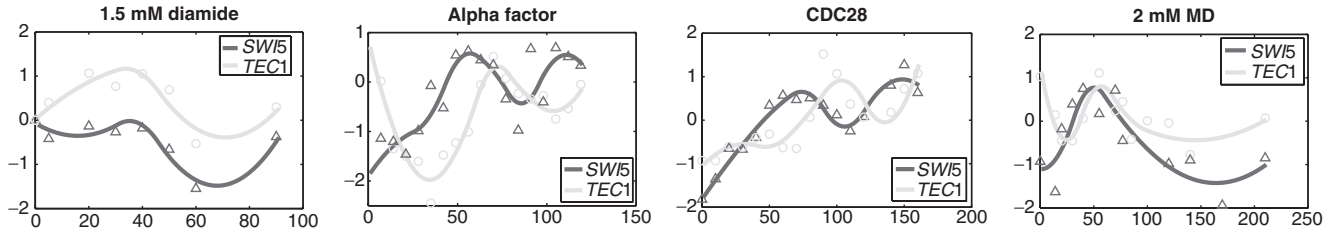


Fig. 1. Expression profiles of a pair of genes in four time series experiments carried out under different conditions. Plots show the expression profiles of the transcription factor *SWI5* (dark curve) and the gene *TEC*1 (light curve). *SWI5* is a known activator of *TEC*1 (Kato *et al*., 2004). While there is a clear lagged regulatory relationship between this pair, the expression profiles of the genes between figures have no common expression pattern and the actual lag is also different. The original data points for *SWI5* and *TEC*1 are indicated by dark triangles and light circles, respectively.

cycle and stress experiments) such an assumption cannot hold as shown in Figure 1.

Formally, in this article we consider the task of combining diverse time series datasets for pairwise lagged regulatory relationship inference. We formalize this problem as follows: the input is a set of M datasets, each containing N time series profiles of the same length for genes $\{G_1, \ldots, G_N\}$. In addition, we are given a subset, P, of pairs of genes: $P = \{\langle G_{a_j}, G_{b_j}\rangle : j = 1, \ldots, P\}$ where $a_j, b_j \in \{1, \ldots, N\}$. We assume that we can associate an unknown, linear,[1] *timescale factor*, R_i $(i = 1, \ldots, M)$, with each of the datasets. We also assume an unknown *canonical lag* d_j $(j = 1, \ldots, P)$ for each of the pairs $p_j \in P$. For each such pair p_j, we model the expected lag in dataset i as $R_i d_j$.

Our goal is to infer the unknown parameters R_i and d_j and assign a value to an indicator variable, Z_{ij}. This variable is set to 1 if there is a time-lagged regulation between the pair of genes p_j in dataset i and to 0 otherwise.

After presenting a formal model for this problem, we discuss an expectation maximization (EM) algorithm for estimating the model parameters. Once these parameters are known, the prediction problem for each new pair can be carried out independently.

We applied our algorithm to a set of 16 time series datasets from yeast. These datasets measured several responses, including different types of stress, cell cycle and DNA damage. As we show in Section 3, the model learned by our algorithm was able to classify pairs as interacting or not, achieving much better accuracy than the current, single experiment-based, methods. We have also analyzed new predictions made by our algorithm and found that some of them are supported by other high throughput datasets and by functional annotation databases.

1.1 Related work

There have been two primary computational approaches for regulatory relationship inference from time series expression data. The first used time lag analysis in a single dataset and the second used correlation coefficients (implicitly assuming a fixed delay in all experiments) to combine time series experiments.

Representative articles from the first (single experiment based) approach include Qian *et al*. (2001) which searched a cell cycle expression dataset for time-lagged and inverted correlations using a local alignment algorithm. Schmitt *et al*. (2004) applied a similar analysis to two highly sampled yeast datasets (independently for each one). Balasubramaniyan *et al*. (2005) used the Spearman rank correlation to compute the time-lagged correlation between genes in a single time series expression dataset. However, these methods cannot be directly applied to the more general problem of combining data from multiple datasets under different experimental conditions. As for the second approach, Lee *et al*. (2004) used correlation coefficients to combine a large number of human expression datasets to search for correlated pairs. Mutual information (Liu *et al*., 2005) is also widely used for determining regulatory relationships. Liang (1998) used mutual information to infer interactions and regulatory relationships between genes. Zhao *et al*. (2006) aimed to predict regulatory relationships using pairwise mutual information followed by an application of the maximum description length (MDL) principle for model selection. These methods assume a fixed time delay which might not be true across different experimental conditions. As mentioned above, the main difference between our method and these previous methods is in the ability to infer regulatory, time-lagged relationships from multiple time series datasets. Our method combines the advantages of the first approach (a more reasonable biological assumption about the lag duration) with the advantages of the second approach (relying on much more data) in a unified framework.

There have been a lot of recent interests in using various types of graphical models to search for regulatory relationships in expression and other types of data. For example, Ong *et al*. (2002) used dynamic Bayesian networks (DBNs) to determine regulatory relationships in a single *E.coli* time series dataset. In order to use DBNs with more than one time series dataset, one must determine a common time unit so that edges in the network have the same meaning in both datasets. We are not aware of any work that combined multiple time series datasets using DBNs. We believe that the ratios learned by our algorithm will be useful for such temporal mapping (see also Section 4).

A number of recent articles presented methods for combining time series datasets that study the same biological system. Spellman *et al*. (1998) used the Fourier transform to determine the phase of different cell cycle datasets and then combined them to identify cycling genes. Aach and Church (2001) used dynamic

[1]For gene expression data, the linearity assumption for scaling can be justified (Bar-Joseph *et al*., 2003*a*). However, a more general problem can also be stated using a more complex scaling function.

programming to align two expression datasets. Bar-Joseph *et al.* (2003*a*) used continuous alignment for the same purpose. All the above methods assume that the expression of a gene in one experiment can be transformed to its expression in another experiment (by an appropriate temporal mapping). In contrast, our method does not rely on such assumption. Thus, our algorithm is the first to combine experiments under very different conditions for the task of pairwise time-lagged regulation inference.

There are a few other existing methods for analyzing time series data which, while they have not been used so far in this domain, can be applied to identify pairwise regulatory relationships. In this article, we compare our results to two such methods. The first, cross recurrence quantification method (Marwan, 2003) constructs a binary recurrence matrix for any two time series. Position i, j in the matrix is set to 1 if the distance between the ith time point in the first series and the jth time point in the second is below a certain threshold, otherwise it is set to 0. The longest diagonal of ones in this matrix represents the best lag assignment for this pair. The second method we compare to is the multichannel singular spectrum analysis (MSSA) (Ghil *et al.*, 2002). This method calculates a lag-covariance matrix for each pair of time series. Entry i, j in this matrix is the lagged correlation coefficient between the first series starting at position i and the second starting at position j (possibly truncated for one of the series to ensure that both have the same length). While these two methods search all possible time lags between two time series, they do not utilize the relationships between different conditions (for example, to infer a canonical lag d for a pair). In addition, these methods treat each pair independently and cannot use information from other pairs in the same condition.

1.2 Pairwise versus combinatorial interactions

The use of time-lagged analysis for gene expression data relies on a number of assumptions. While these assumptions only hold for a subset of the pairs, it is a large enough set so as to justify their use. In addition, the results of our algorithms can be used as a pre-processing step for a more detailed analysis as we discuss below.

In our model, the expression values of the TF are assumed to represent their activation levels. This assumption ignores post-transcriptional modification which may impact the activation levels of a TF. However, as was shown by Segal *et al.* (2003), this assumption does hold for many TFs and thus can be used for recovering at least a subset of the regulatory relationships.

Another simplifying assumption is the pairwise relationship, which ignores the combinatorial process that is used to regulate a subset of the genes. However, there are many cases in which these assumptions hold:

- Many biological pathways are linear, and pairwise interactions have been shown to play an important role in many biological systems (Qian *et al.*, 2001). In addition, linear pathways can be used to represent many regulatory interactions (Yeang *et al.*, 2004).
- Many combinatorial relationships involve an OR logic which can be recovered by our method (Beer and Tavazoie, 2004).
- Our method provides important data pre-processing so that more sophisticated combinatorial methods (such as DBNs) can be used. These combinatorial methods often rely on the assumption that timescales in training datasets are the same and thus require the timescale mapping learned by our method before they can be applied to expression data.

2 A PROBABILISTIC ALGORITHM FOR COMBINING TIME SERIES EXPRESSION DATASETS

2.1 Multiple datasets tabular combination (MDTC) model

We introduce the multiple datasets tabular combination (MDTC) model. This model can be applied to infer pairwise regulatory relationship between TFs and genes from multiple time series microarray expression datasets collected under a variety of experimental conditions. Figure 2a presents the tabular representation of our model. Each column represents a particular TF–gene pair, and each row (except the first) represents a particular experimental dataset. The first row represents the canonical lag d_j for each pair. This parameter allows us to associate different lags with different pairs. The MDTC model associates four variables with each cell in the table: the observed expression profiles for the TF and gene in this cell (denoted by T_{ij} and G_{ij}, respectively), an indicator variable Z_{ij} for the existence of a regulatory relationship between this TF–gene pair in this dataset and, if $Z_{ij} = 1$, the actual lag (denoted by D_{ij}) for this relationship. Both Z_{ij} and D_{ij} are unobserved. In addition, the MDTC model associates one parameter with each row: the timescale factor R_i. R_i represents the linear transformation required to translate the time unit of one experiment to another.

Given the column and row parameters, the expected lag, Expected_Lag_{ij}, for each cell can be computed as the product of the canonical lag for the TF–gene pair and the timescale factor for this experiment, $R_i d_j$.

Figure 2b uses a graphical model to illustrate the dependencies among variables in the MDTC model. Let $T_{ij}(t)$ and $G_{ij}(t)$ denote the expression values of the TF and gene of pair j in datasets i at time t, respectively, and let L_i denote the length of experiment i, θ denote the model parameters $\{R_i, d_j, (\sigma_i^D)^2, (\sigma_i^G)^2\}$ $(i = 1, \ldots, M; j = 1, \ldots, P)$.

The conditional probabilities in our graphical model are defined as follows. The probability of $G_{ij}(t)$ conditioned on $\{T_{ij}, D_{ij}, Z_{ij} = 1, \theta\}$ is:

$$P(G_{ij}(t) \mid T_{ij}, D_{ij}, Z_{ij} = 1, \theta) \sim \begin{cases} \mathcal{N}(0, (\sigma_i^G)^2) & t \in [0, D_{ij}]; \\ \mathcal{N}(T_{ij}(t - D_{ij}), (\sigma_i^G)^2) & t \in (D_{ij}, L_i]; \end{cases} \quad (1)$$

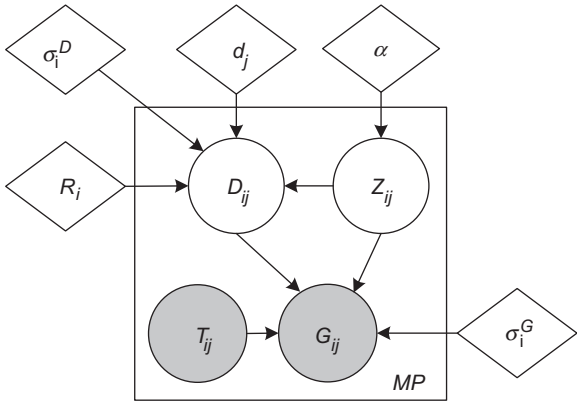
and the probability of $G_{ij}(t)$ conditioned on $\{T_{ij}, D_{ij}, Z_{ij} = 0, \theta\}$ is:

$$P(G_{ij}(t) \mid T_{ij}, D_{ij}, Z_{ij} = 0, \theta) \sim \mathcal{N}(0, 1); \quad (2)$$

Equations (1) and (2) reflect the key assumption in our model.

Timescale factor	Experimental conditions	TF–gene pair 1	...	TF–gene pair P
1	Canonical condition	d_1	...	d_P
R_1	Condition 1	$V_{1,1}$	...	$V_{1,P}$
R_2	Condition 2	$V_{2,1}$	...	$V_{2,P}$
...	...	...	...	...
R_M	Condition *M*	$V_{M,1}$	...	$V_{M,P}$

(**a**)



(**b**)

Fig. 2. The multiple datasets tabular combination model. (**a**) Tabular representation. $V_{ij} = \{T_{ij}, G_{ij}, D_{ij}, Z_{ij}\}$, where T_{ij} and G_{ij} are the expression profiles for the TF and gene in pair j under condition i, respectively. D_{ij} is the actual lag. Z_{ij} is a binary variable indicating whether there is a regulatory relationship in this cell. R_i is the timescale factor for condition i. d_j is the canonical lag for pair j. (**b**) Graphical model representation. See text for details.

When $Z_{ij} = 1$, the gene's profile is a lagged noisy repeat of the TF's profile [Equation (1)]. The distance of this lag is D_{ij}. A Gaussian noise with variance $(\sigma_i^G)^2$, is added to this repeat. This noise represents the biological and experimental noise that may lead to slight difference from the expected expression level. We learn different values of σ_i (noise levels) for different experiments, because our work uses several expression datasets from different labs and using different types of arrays (cDNAs, affy, etc.). Prior to its activation by the TF (between time point 0 and D_{ij}) the gene's profile is modelled as a Gaussian with zero mean and the same variance. When $Z_{ij} = 0$, the gene might be either regulated by another TF or not activated. To reflect this uncertainty, each point in the profile is modelled as a Gaussian with zero mean and unit standard deviation in Equation (2) (we initially normalize all profiles to zero mean and unit standard deviation).

Based on Equations (1) and (2), we can derive the overall dependency of the expression profile G_{ij} on its parents by integrating the probabilities:

$$P(G_{ij} \mid T_{ij}, D_{ij}, Z_{ij}, \theta) = \exp\left(\int_{t=0}^{L_i} \log P(G_{ij}(t) \mid T_{ij}, D_{ij}, Z_{ij})\, \mathrm{d}t\right); \quad (3)$$

This equation models the probability of a gene's profile as a product (or sum in log space) of the probabilities of its values in the individual time points.

Since we do not try to explain the profile of the TF in the pair, equal probability is assigned to any TF profile:

$$P(T_{ij}(t) \mid \theta) \sim \mathcal{N}(0, 1); \quad (4)$$

Again, the overall dependency of the expression profile T_{ij} on its parents can be obtained by:

$$P(T_{ij} \mid \theta) = \exp\left(\int_{t=0}^{L_i} \log(P(T_{ij}(t)))\, \mathrm{d}t\right); \quad (5)$$

Note that, in Equations (3) and (5), we use integral to represent the conditional probabilities of G_{ij} and T_{ij}. In practice, we approximate this integral by uniformly sampling at a high rate from a continuous representation of the profiles. However, for the continuous representation we use there is a general way to integrate the observation noise as described by Bar-Joseph *et al.* (2003*b*). We have thus used this general notation when presenting our model.

We model the conditional probabilities of lag D_{ij} as follows:

$$P(D_{ij} \mid Z_{ij} = 1, \theta) \sim \mathcal{N}(\text{Expected\_Lag}_{ij}, (\sigma_i^D)^2); \quad (6)$$

$$P(D_{ij} \mid Z_{ij} = 0, \theta) \sim \text{Uniform}(0,\ L_i); \quad (7)$$

In Equations (6) and (7), when $Z_{ij} = 1$ the actual lag D_{ij} is assumed to follow a Gaussian distribution[2] whose mean is Expected_Lag_{ij} which is equal to $R_i d_j$, and variance is $(\sigma_i^D)^2$. $(\sigma_i^D)^2$ represents biological and experimental noise which may lead to lags that are slightly different from the expected lag. When $Z_{ij} = 0$ no regulatory relationship exists for this pair. Thus, the lag D_{ij} is not meaningful and any value in its range is equally probable.

Finally, in Equation (8) we assign the same prior probability, α, to every Z_{ij}. This prior can be determined by domain knowledge about the expected number of interactions.

$$P(Z_{ij} = 1 \mid \theta) = \alpha \quad (8)$$

2.2 Learning the parameters of the MDTC model

We use an EM algorithm to estimate our model parameters, $\theta = \{R_i, d_j, (\sigma_i^D)^2, (\sigma_i^G)^2\}\,(i = 1, \ldots, M; j = 1, \ldots, P)$, by seeking to maximize the expected likelihood.

[2]Since the actual lag should be between zero and the length of the profile, the distribution is in fact a truncated Gaussian distribution. In practice, we did sample D_{ij} according to this truncated distribution. However, since the normalization term for this truncated Gaussian is very close to 1, we ignored its effect when updating the parameters in M-step.

In the E-step, we calculate the expectation of the complete log-likelihood listed below. The expectation is under the distribution of the hidden variables given the observed variables and the parameters. Namely:

$$E(LL) = \sum_{i=1}^{M}\sum_{j=1}^{P}\big\{E(\log P(G_{ij}|T_{ij}, D_{ij}, Z_{ij}, \theta)) + E(\log P(T_{ij}|\theta)) + E(\log P(D_{ij}|Z_{ij}, \theta)) + E(\log P(Z_{ij}|\theta))\big\} \quad (9)$$

where M is the number of experimental conditions (rows), and P is the number of pairs (columns). This expectation is intractable in that it contains the integral over the joint distribution of Z_{ij} and D_{ij}. We used Gibbs sampling to approximate this expectation.

D_{ij} is sampled from the conditional distribution, $P(D_{ij}|T_{ij}, G_{ij}, Z_{ij}, \theta)$, which is proportional to $P(G_{ij}|T_{ij}, D_{ij}, Z_{ij}, \theta) \times P(D_{ij}|Z_{ij}, \theta)$, where

$$P(G_{ij}|T_{ij}, D_{ij}, Z_{ij} = 1, \theta) = \exp\left(-\frac{1}{2(\sigma_i^G)^2} \times \mathrm{SE}_{ij}^1(D_{ij})\right) \times \exp\left(L_i \times \log\left(\frac{1}{\sqrt{2\pi}\sigma_i^G}\right)\right) \quad (10)$$

$$P(G_{ij}|T_{ij}, D_{ij}, Z_{ij} = 0, \theta) = \exp\left(-\frac{1}{2} \times \mathrm{SE}_{ij}^0\right) \times \exp\left(L_i \times \log\left(\frac{1}{\sqrt{2\pi}}\right)\right) \quad (11)$$

where $\mathrm{SE}_{ij}^1(D_{ij})$ is a function of D_{ij} representing the squared error of the gene's profile compared to its mean when $Z_{ij} = 1$ [defined by Equation (1)]. Similarly, SE_{ij}^0 is the squared error compared to zero which is the mean of the gene's profile when $Z_{ij} = 0$ [defined by Equation (2)]. In practice, we approximate the squared error by uniformly sampling a set of points on the domain of the profile and calculating the weighted sum of squared difference between two curves evaluated on these points.

Z_{ij} is sampled from the conditional probability $P(Z_{ij}|D_{ij}, T_{ij}, G_{ij}, \theta)$, which is proportional to $P(G_{ij}|T_{ij}, D_{ij}, Z_{ij}, \theta) \times P(D_{ij}|Z_{ij}, \theta) \times P(Z_{ij}|\theta)$, where $P(G_{ij}|T_{ij}, D_{ij}, Z_{ij}, \theta)$ can be calculated by Equations (10) and (11).

In the M-step, we search for the parameters, $\theta = \{R_i, d_j, (\sigma_i^D)^2, (\sigma_i^G)^2\}$ $(i = 1, \ldots, M; j = 1, \ldots, P)$, in order to maximize the expected log-likelihood approximated in the E-step. The final update rules, as well as their derivations are presented in the supporting material due to lack of space. Briefly, the update rule for $(\sigma_i^G)^2$ can be computed in close form using standard Gaussian MLE techniques. Unlike $(\sigma_i^G)^2$, there are no closed form rules for updating R_i, d_j and $(\sigma_i^D)^2$. We used coordinate ascent to find approximate solutions for these equations.

In our EM algorithm, we set the initial values of all timescale factors R_i to be 1. The initial value of d_j for TF–gene pair j is either randomly sampled or set to be the average value of lags corresponding to the maximum correlation scores in each experiment for this pair. σ_i^D is initially set to be standard deviation between lags corresponding to maximum correlation scores (the initial values of d_j). Finally, the initial value for σ_i^G is estimated from repeated experiments which are available in many cases for time point 0.

For inference, we define a confidence score, conf_{ij}, to be the posterior probability of Z_{ij} given the observed variables, $P(Z_{ij}|T_{ij}, G_{ij}, \theta)$. This posterior is approximated in the final E-step by samples of Z_{ij} and D_{ij}.

Interested readers are referred to our Supplementary Material for a detailed derivation of MDTC model and learning algorithm.

2.3 Predicting new pairs

Given a set of pairs and multiple datasets we can use the algorithm presented in Section 2.2 to learn the model parameters. Using these, we can employ an algorithm to make predictions regarding new TF–gene pairs. Given a new TF–gene pair, we construct a new table with only one column for the new pair. Our iterative algorithm runs on this table holding the learned parameters $\{R_i, (\sigma_i^D)^2, (\sigma_i^G)^2\}$, fixed to estimate the canonical lag d and the confidence scores $(\mathrm{con}f_i)$ for this new pair. Following convergence we threshold the confidence scores to arrive at a prediction for each condition. While the learning method may take a long time to converge (~12 h for 500 training pairs in 16 experiments on a 3.2 GHz CPU), the prediction algorithm is very quick (a few seconds for each pair) and is easy to parallelize making it possible to apply our method to datasets with tens of thousands of genes (for example, human time series expression data).

For a schematic illustration of our proposed algorithm, please refer to our Supplementary Material.

3 RESULTS

3.1 Datasets and positive examples

We collected time series microarray expression data for the yeast *Saccharomyces cerevisiae* under different experimental conditions from two online databases (SMD and NCBI's GEO). After removing datasets containing <6 time points we ended up with 16 datasets. These datasets cover various conditions including stresses, cell cycle using different arrest methods and responses to DNA damage. The experiment lengths range from 80 to 480 min. These datasets are summarized in supporting website. Most time series expression datasets contain only a small number of time points which are usually not uniformly sampled. As a pre-processing step we first fitted splines, which were shown to provide a good fit for this data (Bar-Joseph *et al.*, 2003*a*), to the discretely sampled values of each gene. Thus, our algorithm uses a continuous representation of the time series expression profiles of genes. Following spline assignment the original data is ignored. Next, we normalized each continuous curve setting its mean to 0 and standard deviation to 1. This step helps in overcoming differences in amplitude between TFs and the genes they regulate. Finally, for computational convenience we rescaled the profiles under every condition so that they have a unit length. Although we alter the time unit in each experiment, since we know the original length of each experiment we

can easily reconstruct the actual timescale factors, $R_i (i = 1, \ldots, M)$.

Our TF–gene pair space is made of 184 TFs and 6229 genes from *S.cerevisiae*. We have also extracted a set of 1039 known interactions of TF–gene pairs. These interactions have been manually curated from the literature (Lee *et al.*, 2002) and are denoted as 'real pairs' below. Our algorithm searches for both activation and repression regulatory relationships. In order to identify repression by a TF we carry out the same algorithm described above using the inverted TF profile.

3.2 Timescale factors

To compute the parameters of our model, we randomly selected 500 pairs from the set of real pairs mentioned in Section 3.1, and ran our algorithm to infer model parameters from these 500 real pairs.

One of the key problems in combining time series expression datasets is the difference in timescale. The timescale factor, R_i, learned by our proposed algorithm allows us to translate a timescale in one experiment to a timescale in another experiment. While this is useful for our purpose (inferring lagged regulatory relationship), it can also be useful for other types of expression analysis. For example, one can learn DBNs from multiple expression datasets using this transformation. We have thus tried to verify that the factors learned are indeed meaningful.

As mentioned in the Section 1, there have been previous attempts to compute such temporal transformations. While these methods relied on a very different model (trying to align the expression profiles of individual genes), it is interesting to compare the results learned by our model and those previous results. We have thus compared our timescale factors with previous research on aligning cell cycle expression data. Bar-Joseph *et al.* (2003*b*) reported cell cycle lengths for four experimental conditions (CDC15, CDC28, Alpha and FKH1/2 knockout). Three of these datasets (CDC15, CDC28 and Alpha) were also analyzed by Spellman *et al.* (1998) and these two articles agreed on the cell cycle durations for these experiments. All four experiments are present in our datasets. Figure 3a shows the agreement between the timescale factors learned by MDTC model and these previous studies. As can be seen, the agreement is very good, indicating that our model is able to learn reasonable timescale factors.

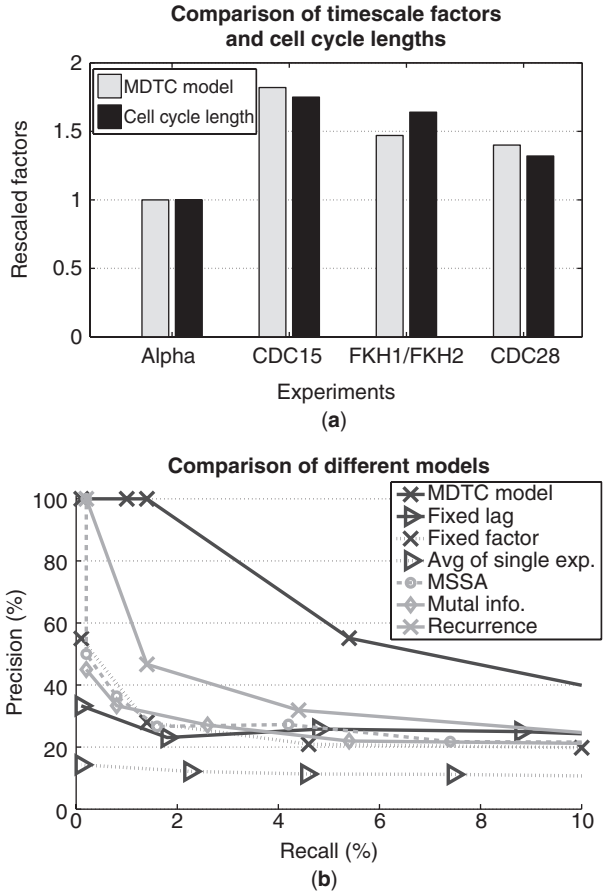


Fig. 3. (**a**) Comparison of the timescale factors learned by MDTC model and cell cycle lengths reported by Bar-Joseph *et al.* (2003*b*) for four different gene expression datasets. All the timescale factors and lengths are divided respectively by the timescale factor and length of the 'Alpha' condition for rescaling. (**b**) Comparison of precision–recall curves for seven different methods for regulatory relationship inference from time series expression data. Recall is defined as the fraction of real pairs that are correctly predicted (sum of correct predictions divided by the total number of true interactions in the datasets). Precision is defined as the fraction of correct predictions out of all predicted interacting pairs (sum of correct predictions divided by the total number of interactions identified by the algorithm).

3.3 Predicting new TF–gene pairs

Using the prediction algorithm presented in Section 2.3, we can predict whether a TF–gene pair has a regulatory relationship. While the exact number of true TF–gene pairs in yeast is unknown, it is unlikely that a gene will be bound by more than 10 TFs (Harbison *et al.*, 2004). Since there are roughly 200 TFs in yeast, a loose upper bound on the total number of interacting TF–gene pairs is 5% of the total pairs. Thus, more than 95% of random pairs represent negative examples. To test the ability of our algorithm to identify TF–gene pairs, we first used 500 real-pairs to learn the model parameters. We next chose another (non-overlapping) set of 500 real pairs and generated 9500 'random pairs'. We ran our prediction algorithm on every pair in both sets (real and random). We used the threshold that worked best for the initial set ($S = 0.5$) to convert our posteriors to binary values. In Figure 3b, we present a precision–recall curve for different values of C (the number of experiments in which a pair passes the threshold). We also present six other precision–recall curves which are derived for methods suggested in the past for inferring regulatory relationships from time series expression data or for methods used to analyze other types of time series data (see Section 1.1 for detailed description of these methods). 'Fixed lag' is the method in which we carried out a similar analysis to the MDTC model except that we had all lags fixed to zero. This method is designed to simulate correlation-based methods (Lee *et al.*, 2004). In 'Fixed factor', we run MDTC model except that we fix all ratios to one. This method accounts for the methods that

do not consider timescaling when combining multiple experiments but still allow lags greater than zero. 'Avg of single exp.' uses the algorithm proposed by Qian *et al.* (2001) which computes lagged regulatory relationships for individual experiments. We averaged the precision–recall curves of the individual experiments to obtain the final curve shown in Figure 3b. 'MSSA' is the multichannel singular spectrum analysis (Ghil *et al.*, 2002). For this method we only looked at the upper diagonal of the resulting matrix which corresponds to cases in which the TF precedes the gene. 'Mutual info.' computes the mutual information with a delay of one time point between the TF and gene. 'Recurrence' uses cross recurrence quantification method (Marwan, 2003) to identify the longest diagonal line for each pair. The cutoff value ξ for entries in the matrix for this method was set to be 0.1 using cross-validation. Similar to the way we converted posteriors to binary values for our method we thresholded the scores for these three methods to convert them to binary values (the cross-validated thresholds are 0.6, 0.2 and 4 for MSSA, mutual information and cross recurrence, respectively). The precision–recall curves shown in Figure 3b are drawn as a function of C, the number of experiments in which the value for each method passed the threshold. As can be seen, our method that uses both time-lagged analysis and multiple datasets outperforms all other methods.

We note that the false-positive rate is based on the assumption that all random pairs are negative. This is clearly not the case. Indeed some of the highly scoring random pairs can be verified using other types of data (see Section 3.4). The low coverage (6% for a precision of 50%) can be explained by a number of observations. First, the training data we used to determine which pairs are real is based on small-scale regulatory (binding) experiments. While these experiments can confirm binding of a TF to a gene, this binding may not represent actual activation or repression which is our goal. Second, some TFs are activated post-translationally making it hard to detect their regulatory roles from expression data. Finally, not all conditions under which a particular pair is interacting are included in our 16 datasets, and even those that are included might not be present in high enough number (we require a high value of C for inclusion in the top 6%).

The fact that our method can accurately identify even a small set of interacting pairs is an important issue. While protein–DNA binding data is available for yeast under some conditions, it is not available for most TFs under most conditions. The pairs predicted by our algorithm to be interacting along with the conditions in which they were determined to be interacting can serve to suggest new binding experiments for these conditions. Even more importantly, our method can be readily applied to higher organisms, including humans, where little direct binding information exists.

3.4 Validating predicted TF–gene pairs

As mentioned above, some of the random pairs may actually be real. To test the agreement between our predictions and other high throughput biological data we have used two external

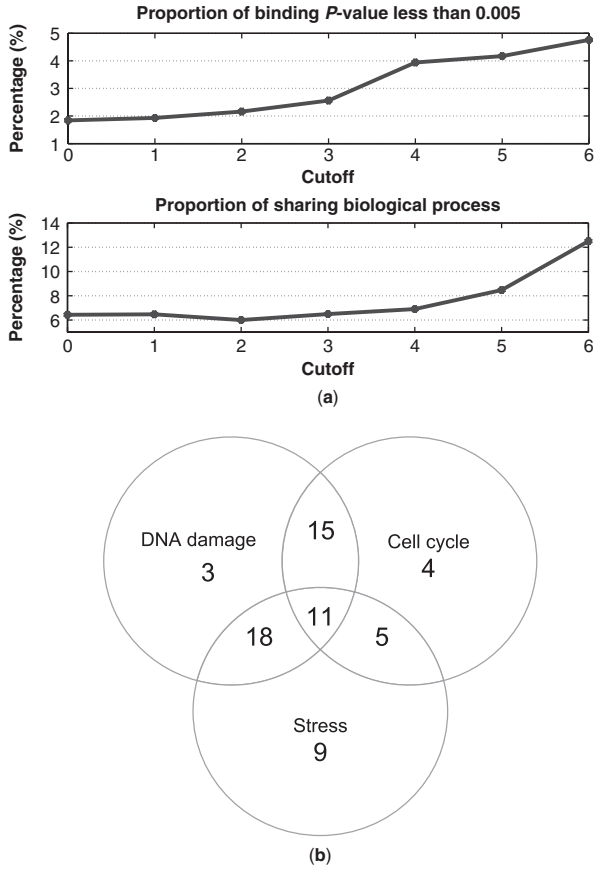


Fig. 4. (**a**) Top: Proportion of random pairs with a binding P-value <0.005 as a function of cutoff, C. Bottom: Proportion of random pairs sharing the same third level GO annotation for biological process as a function of cutoff, C. (Proportions are calculated only for pairs whose annotations are not missing.) (**b**) Venn diagram, showing the number of pairs which are predicted to be interacting in each of the three different groups of experimental conditions studied.

sources. The first is a set of protein–DNA binding experiments, primarily in general growth (YPD) media (Harbison *et al.*, 2004), which provide binding P-value for each TF–gene pair indicating their affinity. The second is the gene ontology (GO) database, which annotates for TFs and genes the biological processes in which they are involved. Figure 4a presents the results of this analysis. As can be seen, the higher the experiment cutoff (or the more experiments in which this pair was predicted to display lagged regulatory relationship), the more enriched the set of predicted pairs for both binding P-values less than 0.005 and GO co-annotations. The fact that only a few of the pairs are validated using the binding P-values is the result of the conditions under which these experiments were carried out. Most TFs were only profiled in YPD media. In contrast, the majority of our time series experiments are stress related, and so it is not surprising that many of the predicted pairs are not found in YPD media.

Figure 1 shows the plots of four conditions with top confidence scores for the $SWI5-TEC1$ pair which appears in

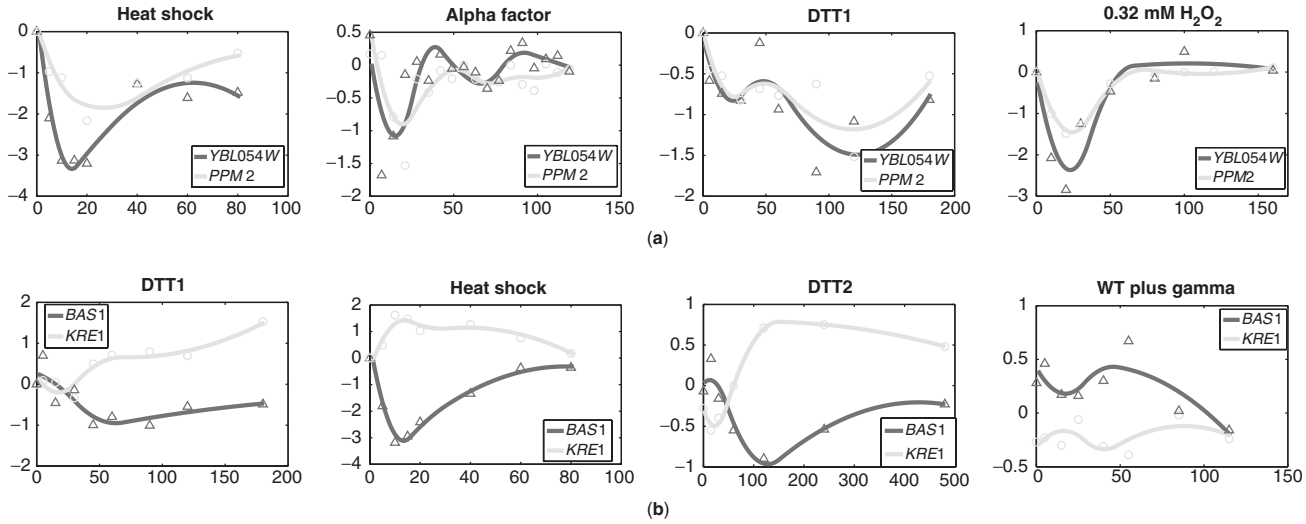


Fig. 5. Expression profiles of TFs and genes under different conditions. The *x*-axis is time and the *y*-axis is the expression level. (**a**) Expression profiles of *YBL*054*W* and *PPM*2 under four different conditions. (**b**) Expression profiles of *BAS*1 and *KRE*1 under four different conditions. Original data points for TFs and genes are indicated by dark triangle and light circle, respectively.

our 'random pair' list, yet is predicted to be interacting by our method. As reported by Kato *et al*. (2004), *SWI*5 was determined to be an activator of *TEC*1 during the M/G1 cell cycle phase. Therefore, this pair is an example of 'random pair' which is actually interacting. The apparently positive correlation shown in Figure 1 is also a strong indication of the regulatory relationship of *SWI*5−*TEC*1 pair.

Figure 5 shows the plots of two other TF–gene pairs detected by our algorithm. While we were unable to find any reference for their relationship, the strong correlated curves for *YBL*054*W*−*PPM*2 and anti-correlated curves for *BAS*1−*KRE*1 in many different experiments may indicate that their relationships exist. Additional example pairs detected by our algorithm can be found on our supporting website.

3.5 Common pathways in different experimental conditions

The experiments we combined in our model can be divided into three groups of experimental conditions. These include stress (eight experiments), cell cycle (five) and DNA damage (three). To test the relationships between these condition groups, we looked at top scoring pairs in our data to see under which conditions they were determined to interact. Figure 4b summarizes these results. As can be seen, most pairs appear in more than one group. This is partially due to the fact that we required at least three experiments with high confidence scores, and our analysis specifically looked for such pairs.

However, it is also apparent that some condition groups share more pairs than others. Specifically, while stress and DNA damage had 18 unique common pairs, and cell cycle and DNA damage had 15, there were only 5 such pairs shared between cell cycle and stress. This may indicate that pathways that are activated in DNA damage response include general stress response pathways and pathways related to cell cycle (DNA repair). However, stress and cell cycle share a relatively small number of pathways. This hypothesis is at least in part supported by previous work. For example, the activation of the DNA repair pathway as part of the cell cycle was noted by Zhou and Elledge (2000). Similarly, various stress conditions have been shown to arrest growth and decrease the activity of cycling genes (McGrath-Morrow and Stahl, 2001). Our results support these two conclusions and indicate that these common responses are activated by a small set of TF–gene pairs.

4 CONCLUSIONS AND FUTURE WORK

In this article, we developed a probabilistic model to combine multiple time series datasets for gene regulatory relationship inference. The MDTC model learns a temporal transformation between different datasets using an input set of potential interacting pairs. Using this transformation it can assign confidence scores to new pairs under each condition representing the posterior probability that the regulatory relationship exists in that condition. Our results indicate that by combining multiple datasets we can overcome problems associated with time-lagged analysis of single experiments, most notably the high false-positive rate. By being flexible with the time lag of the regulatory relationship we can improve upon correlation-based methods that assume a lag of zero.

As mentioned in Section 1, a limitation of our model is that it only considers pairwise relationships, which could be one explanation for the low coverage in Figure 3b. However, we believe that the timescale factors learned by the proposed algorithm can be used to extend our framework to consider combinatorial relationships by converting time units in different experiments to a common temporal scale, and then using DBNs to explore combinatorial regulations.

ACKNOWLEDGEMENTS

This work was supported in part by NSF CAREER award 0448453 to ZBJ.

Conflict of Interest: none declared.

REFERENCES

Aach,J. and Church,G.M. (2001) Aligning gene expression time series with time warping algorithms. *Bioinformatics*, **17**, 495–508.

Balasubramaniyan,R. *et al.* (2005) Clustering of gene expression data using a local shape-based similarity measure. *Bioinformatics*, **21**, 1069–1077.

Bar-Joseph,Z. (2004) Analyzing time series gene expression data. *Bioinformatics*, **20**, 2493–2503.

Bar-Joseph,Z. *et al.* (2003*a*) Continuous representations of time series gene expression data. *J. Comput. Biol.*, **10**, 341–356.

Bar-Joseph,Z. *et al.* (2003*b*) Comparing the continuous representation of time series expression profiles to identify differentially expressed genes. *PNAS*, **100**, 10146–10151.

Beer,M. and Tavazoie,S. (2004) Predicting gene expression from sequence. *Cell*, **117**, 185–198.

Ernst,J. *et al.* (2005) Clustering short time series gene expression data. *Bioinformatics*, **21** (Suppl 1), I159–I168.

Ghil,M. *et al.* (2002) Advanced spectral methods for climatic time series. *Rev. Geophys.*, **40**(1), 1003.

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

Hughes,T. *et al.* (2000) Functional discovery via a compendium of expression profiles. *Cell*, **102**, 109–126.

Kato,M. *et al.* (2004) Identifying combinatorial regulation of transcription factors and binding motifs. *Genome Biol.*, **5**, R56.

Lee,H. *et al.* (2004) Coexpression analysis of human genes across many microarray data sets. *Genome Res.*, **14**, 1085–1094.

Lee,T. *et al.* (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, **798**, 799–804.

Liang,S. (1998) Reveal, a general reverse engineering algorithm for inference of genetic network architectures. In *Proceedings of Pacific Symposium on Biocomputing*, Vol. **3.** pp. 18–29.

Liu,X. *et al.* (2005) An entropy-based gene selection method for cancer classification using microarray data. *BMC Bioinformatics*, **6**, 76.

Marwan,N. (2003) Encounters with neighbours – current developments of concepts based on recurrence plots and their applications. Ph.D. Thesis, University of Potsdam, Potsdam, Germany.

McGrath-Morrow,S. and Stahl,J. (2001) Growth arrest in a549 cells during hyperoxic stress is associated with decreased cyclin b1 and increased p21(waf1/cip1/sdi1) levels. *Biochem. Biophys. Acta*, **1538**, 90–97.

Ong,I. *et al.* (2002) Modelling regulatory pathways in e. coli from time series expression profiles. *Bioinformatics*, **18** (Suppl 1), S241–S248.

Peer,D. *et al.* (2001) Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, **17** (Suppl 1), S215–S224.

Qian,J. *et al.* (2001) Beyond synexpression relationships: local clustering of time-shifted and inverted gene expression profiles identifies new, biologically relevant interactions. *J. Mol. Biol.*, **314**, 1053–1066.

Schmitt,W. *et al.* (2004) Elucidation of gene interaction networks through time-lagged correlation analysis of transcriptional data. *Genome Res.*, **14**, 1654–1663.

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

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.

Tong,A. *et al.* (2001) Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science*, **294**, 2364–2368.

Yeang,C. *et al.* (2004) Physical network models. *J. Comput. Biol.*, **11**, 243–262.

Yeang,C. and Jaakkola,T. (2003) Physical network models and multi-source data integration. In *Proceedings of the Seventh Annual International Conference on Research in Computational Molecular Biology (RECOMB)*. pp. 312–321.

Zhao,W. *et al.* (2006) Inferring gene regulatory networks from time series data using the minimum description length principle. *Bioinformatics*, Advance Access published on July 15, 2006.

Zhou,B. and Elledge,S. (2000) The DNA damage response: putting checkpoints in perspective. *Nature*, **408**, 433–439.