Netw Model Anal Health Inform Bioinforma (2012) 1:3–17
DOI 10.1007/s13721-012-0008-4

ORIGINAL ARTICLE

Data simulation and regulatory network reconstruction from time-series microarray data using stepwise multiple linear regression

Yiqian Zhou · Rehman Qureshi · Ahmet Sacan

Received: 8 January 2012 / Revised: 9 April 2012 / Accepted: 12 April 2012 / Published online: 3 May 2012

Abstract Time-series microarray data can capture dynamic genomic behavior not available in steady-state expression data, which has made time-series analysis especially useful in the study of dynamic cellular processes such as the circadian rhythm, disease progression, drug response, and the cell cycle. Using the information available in the time-series data, we address three related computational problems: the prediction of gene expression levels from previous time steps, the simulation of an entire time-series microarray dataset, and the reconstruction of gene regulatory networks. We model the gene expression levels using a linear model, due to its simplicity and the ability to interpret the coefficients as interactions in the underlying regulatory network. A stepwise multiple linear regression method is applied to fit the parameters of the linear model to a given training dataset. The learned model is utilized in predicting and replicating the time course of the expression levels and in identifying the regulatory interactions. Each predicted interaction is also associated with a statistical significance to provide a confidence measure that can guide prioritization in further costly manual or experimental verification. We demonstrate the performance of our approach on several yeast cell-cycle datasets and show that it provides comparable or greater accuracy than existing methods and provides additional quantitative information about the interactions not available from the other methods.

Y. Zhou · R. Qureshi · A. Sacan (✉)
School of Biomedical Engineering, Drexel University, Philadelphia, PA 19104, USA
e-mail: ahmet.sacan@drexel.edu

Y. Zhou
e-mail: yz86@drexel.edu

R. Qureshi
e-mail: raq22@drexel.edu

1 Introduction

The advent of microarray technologies has enabled a high-throughput evaluation of gene expression, providing a large-scale snapshot of the cellular activity at the molecular level. The availability of these tools has allowed researchers to explore the behavior of entire genomes under different experimental conditions, in a search for mechanistic basis to various cellular behaviors. The analysis of these microarray experimental results has led to new breakthroughs in the understanding, diagnosis, prognosis, and treatment of disease, as well as insights into the functioning of the basic biology of various organisms (Golub and Slonim 1999; van de Vijver et al. 2002; Fan et al. 2010; Wong and Chang 2005).

Gene expression can often be quantified by determining the relative amounts of mRNA transcripts. In this type of microarray experiment, mRNA is harvested from a sample which is then reverse-transcribed into cDNA. This cDNA is labeled with a fluorescent molecule and then allowed to bind to DNA probes attached to the surface of the microarray chip. The process of complementary binding between the cDNA and the DNA probes on the chip is known as hybridization. The fluorescence values that are measured from the chip enable the quantification of the relative amounts of cDNA present in each sample, which determines the relative gene expression (Zhu et al. 2000).

The techniques for analyzing steady-state microarray data are well-characterized (Quackenbush 2002; Mutch et al. 2002; Tusher et al. 2001; Jeffery et al. 2006). However, these techniques are ill-suited to the analysis of

time-series microarray data. Time-series microarray experiments involve harvesting mRNA from an experiment at regular time intervals. This experimental design leads to multiple data points for each gene that can be used to evaluate gene expression over time in a high-throughput manner. Time-series expression data have the potential to provide more comprehensive information about the underlying behavior and inter-relationships of genes than the traditional time-invariant experiments. Furthermore, it can allow for the interpretation of dynamic behaviors in complex biological systems (Aach and Church 2001). Time-series microarray data have many applications including the analysis of circadian rhythms, disease progression, drug response, and the study of the cell cycle (Aach and Church 2001; Spellman et al. 1998; Cho et al. 1998).

Knowledge of the relationships between genes can facilitate the reconstruction of the underlying gene regulatory networks. Each gene's expression can be modified or controlled by various biochemical processes. Transcription factors can directly regulate the synthesis of mRNA, but the expression of genes can indirectly affect the expression of other genes. A gene can inhibit the expression of another gene or it can stimulate the expression of another gene. These activation and inhibition relationships can be represented as a directed graph with nodes representing genes and edges representing the effect of one gene on another. There are several methods of reverse-engineering or modeling gene regulatory networks from two-condition differential expression experiments and time-series experiments. These methods include Boolean networks (Kauffman 1969; Hecker et al. 2009; Gardner and Faith 2005; Abul et al. 2006), correlation networks (Margolin et al. 2006a; Basso et al. 2005; Faith et al. 2007; Stuart et al. 2003), differential equation models (van Someren et al. 2000; Gardner et al. 2003; di Bernardo et al. 2005; Bansal et al. 2006; Chen et al. 1999; Sakamoto and Iba 2001), Bayesian network models (Gardner and Faith 2005; Margolin et al. 2006a), and dynamic Bayesian network models (Margolin et al. 2006b).

Boolean networks represent the earliest attempts at gene regulatory network modeling (Kauffman 1969). Boolean network models describe the expressions of individual genes as binary variables. The state of a gene is determined as a Boolean function of the state of the other gene expressions. Once the data have been discretized, a Boolean network that explains the data must be created. The Reverse Engineering Algorithm (REVEAL) is one algorithm that accomplishes this task (Hecker et al. 2009). REVEAL works by computing the mutual information between sets of two or more genes and trying to find the smallest set of input genes that completely explain the state of an output gene (Gardner and Faith 2005). Boolean network modeling is easy to implement and interpret as a large-scale system. However, since microarray expression data are rather noisy, discretization of the data presents a challenge and may fail to accurately describe the system. Many studies on Boolean networks examine only simulated datasets (Hecker et al. 2009; Margolin et al. 2006b), thus their practical performance is also debatable.

Association networks are among the simplest models, as they ignore directionality and strength of the regulations and model the interaction among a set of genes by an undirected graph with edges weighted by correlation or another measure of similarity or statistical dependence (Hecker et al. 2009; Gardner and Faith 2005; Stuart et al. 2003). Two genes are predicted to interact with one another if their expression patterns are similar, which is often determined by whether or not they meet a predetermined threshold value of the association measure of choice (Margolin et al. 2006a; Basso et al. 2005; Faith et al. 2007). Other methods such as Euclidean distance or mutual information can be used as alternatives to correlation (Hecker et al. 2009). The main advantages of correlation networks are their simplicity and low computational cost. However, correlation networks are ill-equipped to extract more complex information out of microarray datasets. They can only be used to construct undirected graphs, although this is a general limitation for models that do not use time-series data. Furthermore, they are static, cannot accurately distinguish between co-regulation and causality, and have difficulty identifying nonlinear many-to-one interactions (Opgen-Rhein and Strimmer 2007).

Differential equation models provide a more powerful and descriptive formalism for capturing interactions in biological networks (van Someren et al. 2000; Gardner et al. 2003; di Bernardo et al. 2005; Bansal et al. 2006; Chen et al. 1999) (Sakamoto and Iba 2001; Weaver et al. 1999). Differential equation models utilize a system of differential equations that describe gene expression changes as functions of other gene expressions and possibly external environmental factors (Hecker et al. 2009). Differential equation models represent the networks in a more quantitative manner. However, they can be difficult to analyze and generally have a high computational cost. Another problem is that there can be multiple solutions, meaning that multiple ODE systems can be identified from a single dataset. A priori knowledge is often required to provide enough constraints to identify a single solution (Hecker et al. 2009). This problem is compounded by the lack of experimental data to identify the parameters of the interaction kinetics. Consequently, unlike metabolic networks that have well-known reaction kinetics, functional forms that can accurately model complex regulatory interactions have not been available yet.

Bayesian network (BN) models are directed acyclic graphs and represent the expression of each gene as a random variable determined by a probability distribution function that is expressed as a product of conditional probabilities (Gardner and Faith 2005; Margolin et al.

2006a, b). A BN model must find the directed acyclic graph that best represents the data, as determined by means of a scoring function. Popular scoring functions include the Bayesian Information Criteria (BIC) and the Bayesian Dirichlet equivalence (BDe) (Margolin et al. 2006b). Both of these measures impose a penalty for complexity to prevent over-fitting the data. One challenge facing the implementation of BN models is the sheer number of possible directed acyclic graphs that can be constructed from a set of genes. Determining all possible graphs for the set of genes and finding the graph with the maximum score is an NP-hard problem. This issue is addressed by the use of heuristic search such as the greedy-hill climbing approach, the Markov Chain Monte Carlo Method, or simulated annealing. Often several high-scoring networks are found using this approach. This problem is typically addressed by using either model averaging or bootstrapping to determine the most likely network and determine confidence intervals for the interactions (Margolin et al. 2006b).

The main advantages of Bayesian networks are their ability to avoid over-fitting, handle incomplete or noisy data, and combine heterogeneous data types. The main disadvantage is their inability to model feedback loops since the graphs modeling the network cannot include cycles. This issue is addressed in dynamic Bayesian Networks (DBNs) which are an extension of the original Bayesian Network model. DBNs model time-series data rather than steady-state data when performing network reconstruction and have a higher computational cost compared to the traditional Bayesian Network models (Margolin et al. 2006b). Rather than modeling the networks as a directed acyclic graph, DBNs consist of two layers of nodes. Every gene possesses a node in each layer and one layer corresponds to the expression of the gene at time t, while the next layer represents the expression of the gene at time $t + \Delta t$. The edges connecting the genes in the first layer to the genes in the second layer enable the modeling of feedback loops (Hecker et al. 2009). Bayesian and dynamic Bayesian networks have become widely used in gene regulatory network reconstruction (Friedman et al. 2000; Hartemink et al. 2001; Segal et al. 2003; Nachman et al. 2004; Rangel et al. 2004).

While a great deal of focus has been placed on the network reconstruction problem, the prediction and simulation of gene expression values have not received as much attention. We note that methods developed to infer the presence or absence of regulatory interactions are not directly applicable to the prediction problem. On the other hand, the methods that focus on the prediction problem may not lend themselves to the interpretation of their model for inference of interactions. In this study, we use a linear model to represent gene interaction networks and simultaneously solve the network reconstruction and gene expression prediction problems. The neural network approach of Maraziotis et al. (Abul et al. 2006) (referred here as FuzzyNet) is closest in its goals to the problems being investigated in this study. In FuzzyNet, a recurrent neural fuzzy network is trained for time-series data. While neural networks are generally not amenable to interpretation, the rules generated by FuzzyNet allow identification of regulatory interactions. However, unlike the approach described herein, Fuzzynet does not predict the strength of the predicted interactions and also does not provide a confidence measure for its predictions.

In this study, we present a linear model for time-series data and use stepwise multiple linear regression (SMLR) to learn the model parameters from the training dataset(s). To the best of our knowledge, this is the first time a linear model of interaction has been reported to solve the prediction, simulation, and reconstruction problems. The rest of this report is organized as follows. In Sect. 2, we formally define the computational problems and describe our linear model and the process of fitting it to data by using stepwise multiple linear regression. In Sect. 3, we describe the datasets used in the experiments, and present empirical justification for the choice of parameters, including the number of interactions, the statistical significance threshold for interactions, and the number of time points considered in the input. We then present results for the next time step prediction of expression values, the simulation of the entire time-course data, and finally, the inference of the regulatory network. Results are compared with similar studies where applicable. We conclude with a summary of our contributions, contrasting with existing solutions.

2 Methods

Time-series microarray data can be described as an $N \times T$ data matrix, representing the mRNA levels of N genes over T consecutive time points. In this study, we focus on three related computational problems, as illustrated in Fig. 1. In the single time point *prediction* problem (Fig. 1b), one attempts to learn a function that can generate the expression levels in time t from the expression levels at the preceding time point(s). Each pair of time points in Fig. 1b provides a training instance for learning such a function. In the time-series data *simulation* problem (Fig. 1c), the entire time-series data are generated from only the initial conditions given at the first time point. In this study, we model the simulation problem simply as iterations of the single time point prediction problem, leaving more complex approaches accounting deficiencies of this straightforward extension, such as error accumulation, as future work. In the *network reconstruction* problem (Fig. 1d), one attempts to discover the underlying gene regulatory network from the microarray data. While

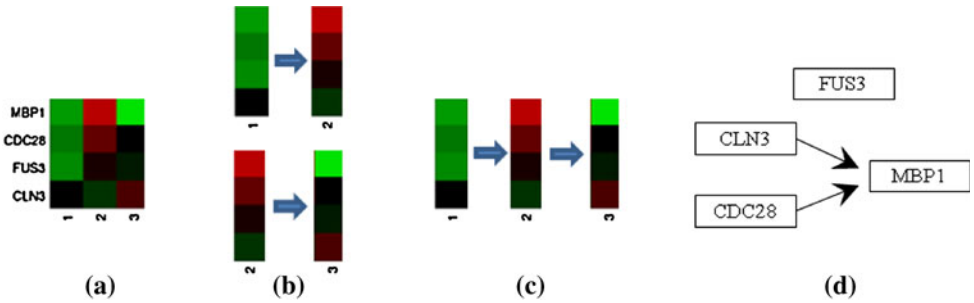

Fig. 1 Demonstration of microarray time-series data and the computational prediction problems investigated in this study. **a** Sample time-series microarray data with 4 genes and 3 time points. *Red*, *green*, and *black* colors denote high, low, and medium expression levels, respectively. **b** Single time point prediction problem showing prediction of expression levels at time t from time $t-1$. **c** Simulation of entire time-series data from the initial expression levels at time $t=1$. **d** An example reconstructed network involving the four genes, where *arrows* indicate transcriptional regulation (color figure online)

network reconstruction problem is often solved independently (Faith et al. 2007; Almansoori et al. 2012), we perform network reconstruction via post-processing of the single time step prediction function.

We model the expression level of each gene as a linear function of the expression levels of the genes in the preceding time step (this model is generalized to consider multiple previous time points below):

$$g_t^j = w^0 + \sum_{i=1..N} w^i g_{t-1}^i$$

where $g_t^j (j = 1, 2, \ldots, N)$ is the expression level of a *response gene* g^j at time t, g_{t-1}^i terms are the expression levels of the candidate *predictor genes* at the preceding time step, N is the number of genes being studied, and w^0 is a constant bias term.

We identify the coefficient weights w^i using stepwise multiple linear regression, with a forward selection strategy (Hadi 2006; Draper and Smith 1998). The predictors for a given gene are identified starting with the inclusion of the constant term w^0. In each forward selection step, individual predictor variables are considered for addition based on their statistical significance in the regression fitting. The p value of an F statistic for each variable is calculated to test the model including and excluding that variable using the null hypothesis that its weight coefficient is zero, using the following equation (Hadi 2006; Draper and Smith (1998):

$$F = \frac{\mathrm{SSE}^* - \mathrm{SSE}}{\mathrm{SSE}/(n-p-1)}$$

where *SSE* is the sum of squared error according to the expanded model using $p+1$ predictor variables, and SSE* is the sum of squared error according to the reduced model using only p predictor variables as follows:

$$\mathrm{SSE} = \sum (y_i - \hat{y}_i)^2$$

$$\mathrm{SSE}^* = \sum (y_i - \hat{y}_i^*)^2$$

where $\hat{y}_i$ and $\hat{y}_i^*$ are the values predicted by the expanded and reduced models, respectively.

If the F statistic is significant, the null hypothesis is rejected, and that particular predictor variable is included in the model. Our forward selection procedure considers the full set of predictor variables, returning a p value for each one. If any predictor variable had a p value less than an entrance tolerance, it was added to the model. This ensures that variables with marginal contributions (with a coefficient close to zero) are omitted from the model.

Since the data were already normalized the constant term w^0 can be set to zero, and without loss of generality, the expression levels of all the genes at time t can be written as:

$$G_t = G_{t-1} \times M$$

where G is an $N \times 1$ vector of gene expression values and M is an $N \times N$ matrix of weight coefficients. The coefficient matrix M can be converted into a sparse matrix, replacing insignificant interactions with zeros.

The model described above utilizes only the most preceding time point as input. This single time point provides only a static snapshot of the changing gene expression levels. It is not, for instance, directly possible to infer whether the expression level of a gene was going up or down during the preceding time point. We therefore consider the more general case of utilizing prior τ time points, where the expression level of a gene g^j is now modeled as a linear function of all the genes from the preceding τ time points:

$$g_t^j = w^0 + \sum_{q=t-\tau \ldots t-1} \sum_{i=1 \ldots N} w_q^i g_q^i$$

Correspondingly, the expression levels of all the genes at time t can now be written as:

$$G_t = [G_{t-\tau}; G_{t-\tau+1}; \ldots; G_{t-2}; G_{t-1}] \times M_\tau$$

where G_t is again an $N \times 1$ vector of predicted gene expression values at time t; the expression levels of the

genes at all previous τ time points are concatenated into a single τN vector, and M_τ is a coefficient matrix of size $\tau N \times N$, containing the coefficients from all genes at the previous τ points. The value of τ can be determined empirically from the mean squared error on the training data, as described in the experiments below. Starting from the first τ time points of a given experiment, the learned coefficient matrix is used to incrementally simulate the rest of the time points.

The weight matrix M (and M_τ) describes the influence of each predictor gene on the response genes. The magnitude of these weights indicates the strength of the interaction and their sign indicates whether the interactions are activating or inhibitory. Each weight is also associated with a p value, indicating the statistical significance of the corresponding interaction. We rank the interactions by their p values and use the top-k most significant interactions in the network reconstruction, where k can be pre-defined from the average number of interactions observed in real networks or discovered empirically, as presented below to minimize the training error. The accuracy of the reconstructed network is evaluated with respect to a reference network, such as the pathways available in the KEGG compendium (Kanehisa and Goto 2000), using the following measures:

$$\text{Precision} \stackrel{\text{def}}{=} \frac{\#\text{ of correctly predicted edges}}{\#\text{ of predicted edges}}$$

$$\text{Recall} \stackrel{\text{def}}{=} \frac{\#\text{ of correctly predicted edges}}{\#\text{ of edges in the known network}}$$

$$F\text{-measure} \stackrel{\text{def}}{=} 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Precision, recall, and F-measure each take values in the range between 0 and 1, with 1 being the best score. The ability to rank the interactions by their significance allows us to control the precision–recall trade-off, which is presented as precision–recall plots below. Note that existing approaches produce or report a single precision and recall result; we use the same number of predicted edges in the network for comparison with earlier studies. In comparisons, we denote our approach as SMLR (stepwise multiple linear regression).

3 Experiments and results

3.1 Datasets

The time-series datasets modeled in this study are from Spellman et al. (1998). These datasets were generated using four different methods to synchronize *Saccharomyces cerevisiae* cell cultures to the same phase of the cell cycle (Spellman et al. 1998; Cho et al. 1998). The experiments utilized multiple strains of yeast and mRNA was harvested from cells extracted from the cultures at predetermined time intervals. The usage of different methods of synchronizing the cultures resulted in four unique datasets, each named after the synchronization method. Each of the datasets consisted of yeast cells whose cell cycles had been arrested at a different phase. This results in the different datasets beginning at different phases of the cell cycle.

One dataset (ALPHA) utilized the alpha factor to arrest the cell cycle and consisted of 18 time points separated by intervals of 7 min. A second dataset separated cells by elutriation (ELU dataset). By separating cells of different sizes the investigators were able to extract cells of similar size that were likely to be in the same phase of the cell cycle. They collected daughter cells that were not budding into new cells. This dataset consisted of 14 time points separated by intervals of 30 min. These first two datasets were collected by Spellman et al. 1998 (Zhu et al. 2000). Spellman et al. included two further datasets from Cho et al. (1998) in their analysis. The third dataset used CDC15 strain of yeast cells, where the cell cycle was arrested by raising the temperature of the culture. This dataset had 24 time points separated by 10- or 20-min time intervals. We excluded the time points that were separated by 20-min intervals from our analysis. The fourth dataset consisted of the strain of yeast possessing CDC28 and also was synchronized by temperature change. This dataset had 17 time points separated by 10-min intervals. All the expression data were normalized so that the mean log2 ratio of the data was 0 (Cho et al. 1998).

3.2 Identification of parameter values

The performance of the linear model was first investigated for the next time step prediction problem. To do so, all "predictor–responder" pairs (i.e., all input–output pairs in Fig. 1b) were extracted from the four datasets and combined into a single set. In a fourfold cross-validation scheme, three-fourths of these pairs were randomly selected for training and the remaining pairs were used for testing. The performance was evaluated in terms of the mean squared error (MSE) of the predicted testing data compared with the real data.

Since we used the p values calculated from the multiple linear regression to determine which genes would be used as predictors of the response gene under consideration, finding a proper cutoff p value was important and prevented us from over-fitting our model to the training data by excluding many insignificant predictors. As demonstrated in Fig. 2a, the average number of predictors per response gene was directly related to the cutoff p value, but there was no clear plateau for the number of predictors with

respect to the p value cutoff. By examining the MSE versus the average number of predictors (Fig. 2b), we were able to identify an average number of predictors giving a minimum MSE value. Optimum MSE values on the test dataset are obtained for the average number of predictors ranging from 2 to 3, which is in line with the number of interactions observed or estimated by others (Andrecut et al. 2008; Thieffry et al. 1998; Guelzim et al. 2002; Luscombe et al. 2004; Andrecut and Kauffman 2006). Using fewer than 2 predictors was insufficient to capture the expression pattern, while using more than 3 predictors resulted in overfitting. Our experiments using multiple preceding time points also showed similar behavior. Thus, in subsequent experiments, we chose a p value cutoff of 0.025, which provided 3 predictors for each gene on average.

We followed a similar approach for determining the optimal number of preceding time points, τ, to consider in the model. Figure 3 shows the MSE for various number of time points used in prediction. The fourfold cross-validation experiment was repeated 1,000 times and the error bars indicate the standard error of the mean for the average MSE in these 1,000 runs. The MSE obtained when 2 preceding time points were used was significantly better than the MSE for other values of τ (p value of two sample t test between the MSE for $\tau = 2$ and $\tau = \{1, 3, 4, 5\}$ are 0.0008, 0.0003, 0.0002, 0.0002, respectively). When 2 time points were used, 53 and 47 % of the predictors were from the first and second preceding time points, respectively.

In order to compare our results with those reported in Abul et al. (2006), we also used CDC15 for training and CDC28 and ALPHA datasets for testing. The result of this comparison is shown in Table 1. We note that the training error obtained by our approach is significantly better than those from FuzzyNet for all but two genes. Overall, SMLR is able to provide lower error rates for 50 % of the predictions. SMLR incurs very high error rates in the testing datasets for two of the genes, namely the transcription factor MBP1 and the S-phase entry cyclin-6 gene CLB6. We attribute the high error rate in the testing datasets for

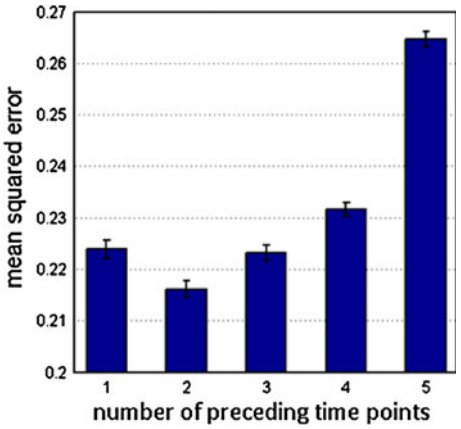

Fig. 3 The mean squared error versus the number of preceding time points used for prediction. *Bars* show the average MSE from of 1,000 fourfold cross-validation experiments. *Error bars* show the standard error of the mean

MBP1 to the fact that the gene expression pattern for MBP1 in the training dataset does not show a clear cyclic expression pattern like the other genes do (Fig. 5), whereas in the testing datasets, such an expression pattern is observed (Figs. 6, 7). This may be due to MBP1 being under different regulatory pressures for different cell-cycle synchronization methods.

CLB6 has three regulators in the KEGG pathway for the genes used in this study, and we suspect that the nonlinear interaction between these regulators is not sufficiently captured by our linear model. Although we anticipated such cases, we leave inclusion of higher order terms into our model under limited data availability conditions as future work. Despite the high MSE values, SMLR is able to detect the gene expression pattern for CLB6 (Figs. 6, 7).

3.3 Time-series data simulation

Taking the next time step prediction function, we iterated the prediction over the entire time course. Only the first τ time points were given as input and the predicted

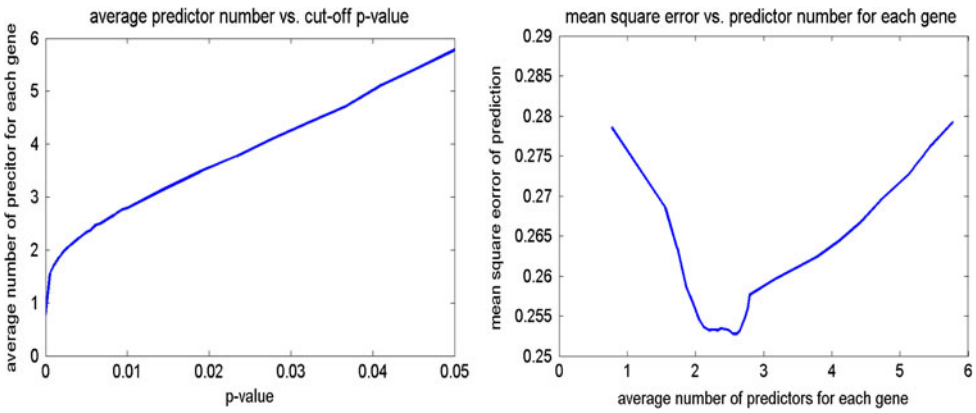

Fig. 2 **a** The average number of predictors versus the cutoff p value calculated including only the most preceding time step. **b** The mean square error versus the average number of predictors. Similar results were obtained when multiple preceding time steps were considered

Table 1 Mean squared error comparison with FuzzyNet for the next time step prediction problem

Dataset	CDC15		CDC28		ALPHA		Average	
Gene	FuzzyNet	SMLR	FuzzyNet	SMLR	FuzzyNet	SMLR	FuzzyNet	SMLR
CLB5	0.17	**0.06**	0.18	**0.08**	0.45	**0.05**	0.27	**0.06**
SWI4	0.36	**0.09**	0.49	**0.08**	0.12	**0.05**	0.32	**0.07**
SIC1	0.45	**0.08**	0.41	**0.31**	0.74	**0.24**	0.53	**0.21**
CDC20	0.55	**0.48**	0.37	**0.07**	0.62	**0.09**	0.51	**0.21**
SW16	0.28	**0.07**	**0.33**	0.36	0.50	**0.13**	0.37	**0.19**
CLN2	0.56	**0.08**	0.58	**0.26**	0.73	**0.23**	0.62	**0.19**
CLN3	0.25	**0.06**	**0.25**	0.27	**0.15**	0.46	**0.22**	0.26
CDC28	**0.13**	0.40	**0.07**	0.41	**0.06**	0.47	**0.09**	0.43
CLN1	**0.19**	0.30	**0.36**	0.61	**0.67**	0.89	**0.41**	0.60
CDC6	0.37	**0.05**	**0.34**	0.97	**0.42**	0.98	**0.38**	0.67
MBP1	0.27	**0.10**	**0.43**	1.91	**0.70**	2.13	**0.47**	1.38
CLB6	0.40	**0.07**	**0.36**	2.86	**0.25**	1.71	**0.34**	1.55

For both FuzzyNet and SMLR, CDC15 dataset was used for training and CDC28 and ALPHA were used for testing. The average MSE calculated for each gene were compared. Superior MSE values for each dataset and gene are shown in bold

expression levels are fed into the next iteration of the simulation. In each simulation experiment, one of the datasets was left out for testing and the model parameters were trained on the remaining datasets.

We have observed that the simulated expression patterns match that of the real data (Fig. 4, top row), but with an increase or decrease in the frequency of the expression patterns. We attribute this change in the periodicity to the fact that the datasets were generated with different time intervals, causing the trained function to output an expression level that is not in-sync with the testing dataset. Specifically, the ELU dataset had a time interval of 30 min, which is larger than the others (7 or 10 min). Testing a model trained for the other three datasets on the ELU dataset would give predictions with an increased period compared to the real data. Conversely, including ELU in training data would give predictions that are beyond the time interval of other datasets, effectively giving accelerated cell cycle for the predicted test dataset. We confirm this by repeating the training with the exclusion of the ELU dataset. As expected, this exclusion corrects the phase shift in the predictions (Fig. 4, bottom row).

Excluding the ELU dataset, we performed three additional experiments, taking each of the remaining datasets for testing (Figs. 5: CDC15, 6: ALPHA, and 7: CDC28). The simulations covered the gene expressions of 83 genes, which were known to be participating in the yeast cell cycle. We present the simulated results of only 14 genes that are later used for the regulatory network reconstruction. For each simulation, we show the predictions for models with $\tau = 1$ (red) and $\tau = 2$ (green). We observe that the overall expression patterns of the predictions are very well matched with the real data. However, the predictions tend to be conservative in their amplitude compared to the real data (especially see CDC6 and CLB5 in CDC15 dataset; SWI4, FAR1, CDC6, SIC1, and CLN2 in ALPHA dataset; SWI4, CDC20, and CLB6 in CDC28 dataset).

In general, the simulated expression levels follow a smoother trend compared to the real data. This is expected, considering that the real microarray measurements contain fluctuations due to biological variations or noise from the data collection technology. The predictions for $\tau = 1$ and $\tau = 2$ have a high degree of overlap. Using two preceding time points as input results in slightly better predictions (see for instance FAR1, CDC6, SIC1, and CLN2 in Fig. 6).

In order to examine the large-scale behavior of the gene expressions, we generated heat-maps for the real and simulated data (Fig. 8). Two clusters of expression patterns have emerged from the heat-map for the real data. The simulated data using both 1 and 2 time preceding time points are able to preserve these expression clusters. A cluster of genes show up-regulation from second to seventh time points and begin to be up-regulated in the next cycle starting from 14th time point. A second cluster of genes show up-regulation between the fifth and tenth time points. The genes CDC20, MBP1, SWI6 show expression patterns different from the other genes. The highly fluctuating behavior of MBP1 explains the high mean squared error reported for MBP1 in Table 1.

3.4 Network reconstruction

Having created a model of the expression of each gene as a linear function of the expression levels of the genes at preceding time points, we were able to directly apply this

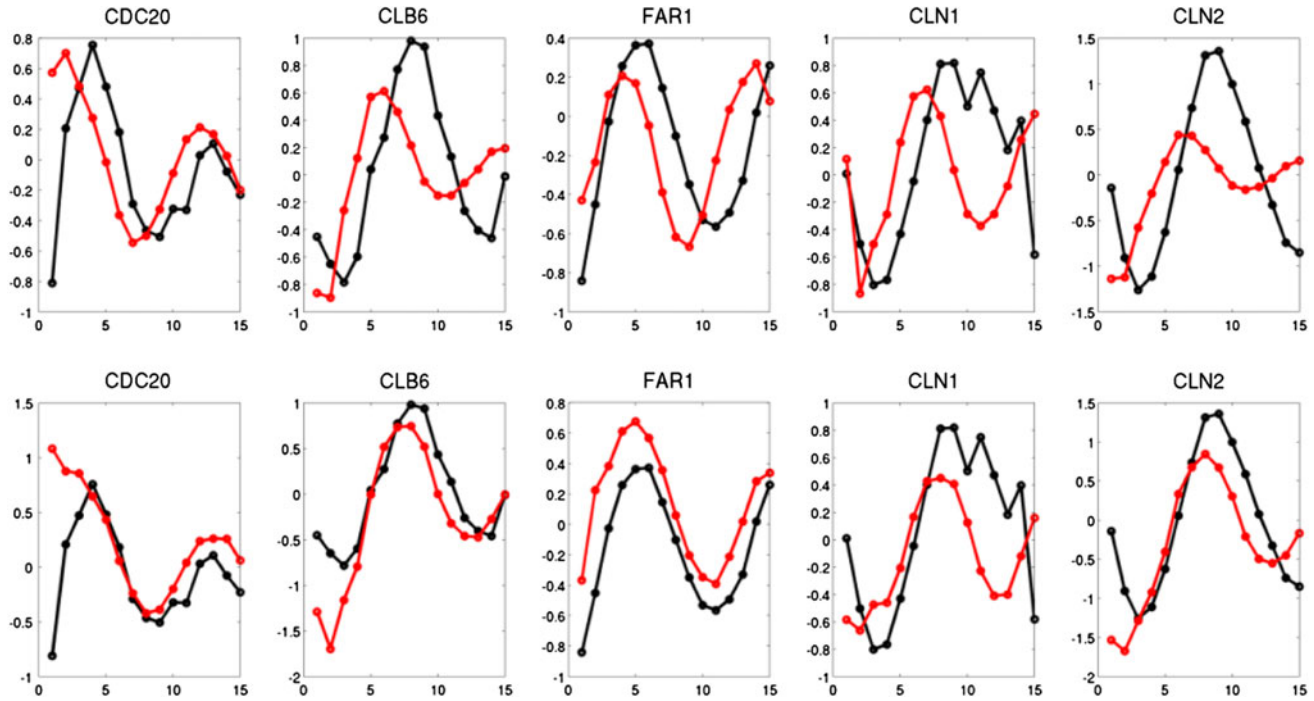

Fig. 4 Including ELU dataset in training causes error in predicted periodicity. The models were tested on the CDC15 dataset. *Upper* training with the ELU, CDC28, and ALPHA datasets. *Lower* training with the CDC28 and ALPHA datasets. Real CDC15 data are shown in *black*, simulated expression levels are shown in *red*. Expression patterns for only 5 of the genes that best illustrate the error in the periodicity are shown (color figure online)

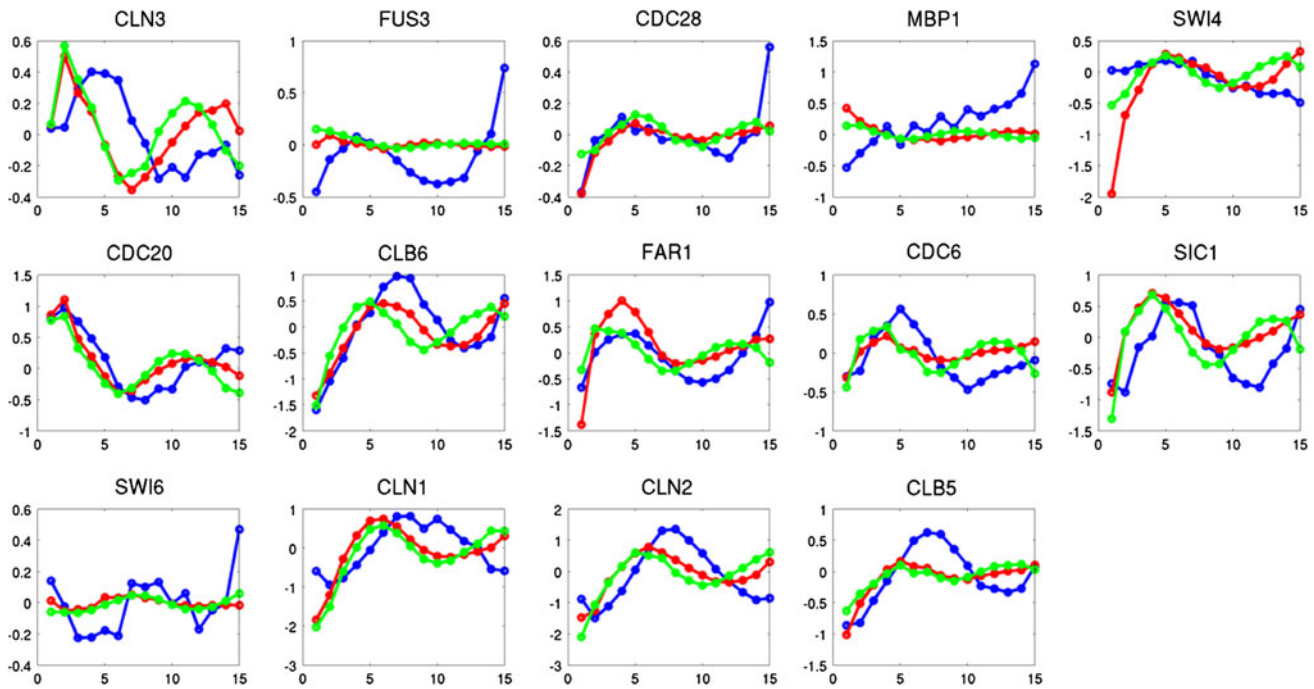

Fig. 5 Simulated data of CDC15 from models trained from ALPHA and CDC28 datasets using one previous time point (*red*) or two previous time points (*green*). The real data are shown in *blue* (color figure online)

model to the gene regulatory network reconstruction problem. A central intuitive assumption in this application is that the coefficients of the predictor genes directly reflect the strength of their influence on the respective target response genes in the gene interaction network. The predictors of all genes were compiled into a single list of

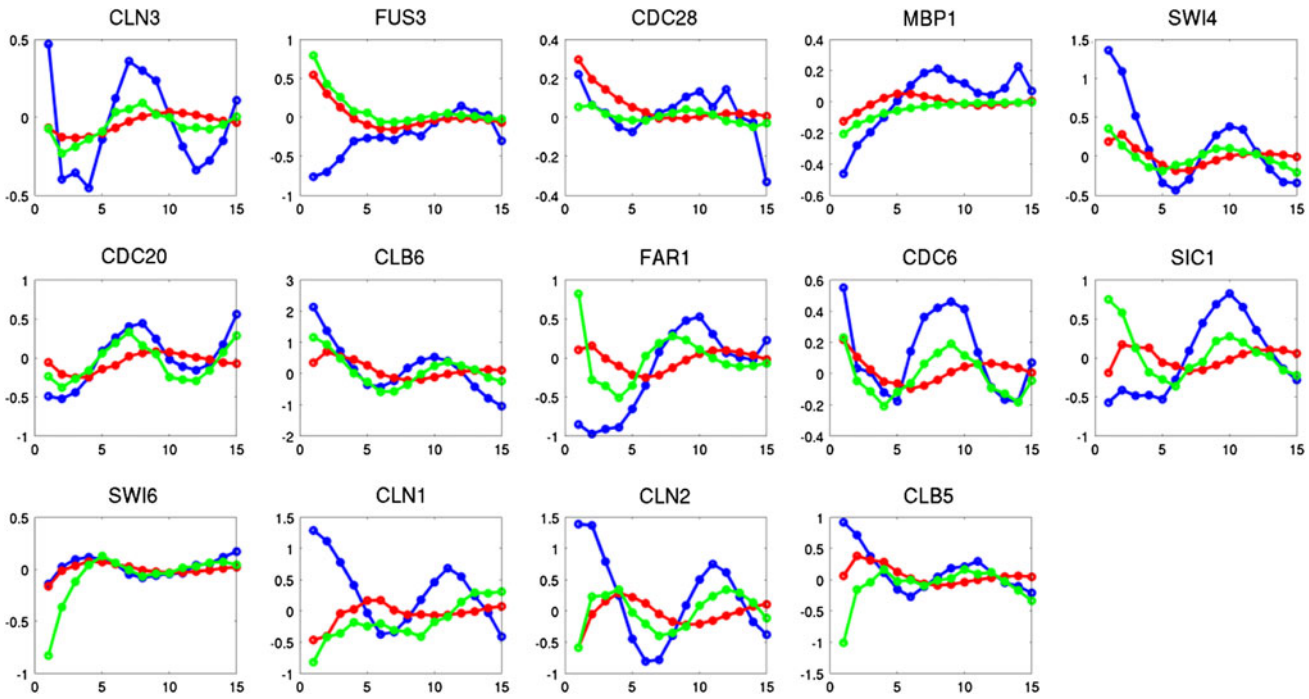

Fig. 6 Simulated data of ALPHA from models trained from CDC15 and CDC28 datasets using one previous time point (*red*) or two previous time points (*green*). The real data are shown in *blue* (color figure online)

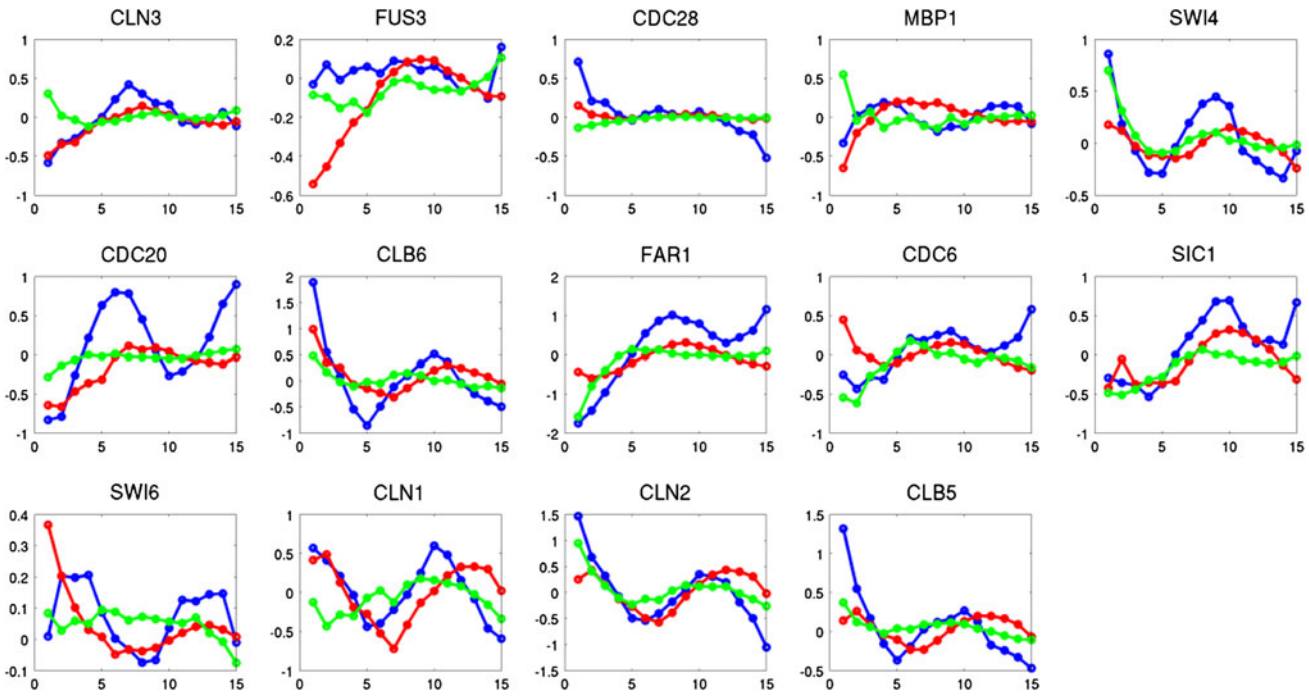

Fig. 7 Simulated data of CDC28 from models trained from CDC15 and ALPHA datasets using one previous time point (*red*) or two previous time points (*green*). The real data is shown in *blue* (color figure online)

predicted regulatory interactions, ranked by their p values. These p values were corrected for false discovery rate using the Benjamini–Hochberg method (Benjamini and Hochberg 1995). For a given p value cutoff, the interactions with greater statistical significance were used to reconstruct the regulatory network.

The regulatory network was reconstructed by connecting the selected predictors with their response gene using a

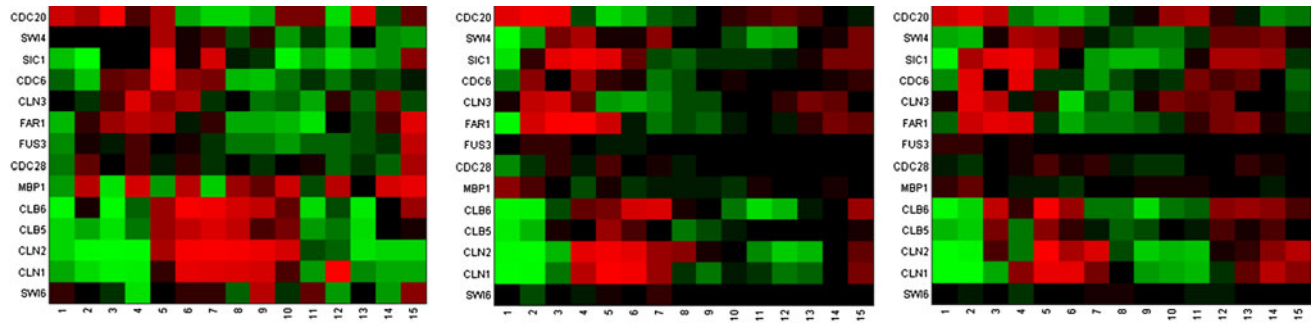

Fig. 8 The heat-maps show the periodic behavior of the genes over time steps. *Left* real data. *Middle* simulated data using one time points. *Right* simulated data using two time points

directed edge. The magnitude of the weights in the model represents the strength of the regulatory interaction, and their sign determines whether it is an activating or inhibitory regulation. For comparison with existing methods that only determine the presence or absence of the interactions, we constructed the regulatory network as an unweighted, directed graph. We compared the gene regulatory networks reconstructed from our model to the networks reconstructed using DBN (Kim et al. 2004) and FuzzyNet (Abul et al. 2006) methods. The target network contained 14 genes, as shown in Fig. 9a. Using all of the datasets for training, the DBN model predicted 15 edges consisting of 4 correct, 8 half-correct, and 3 incorrect edges (Fig. 9b), where the correct and incorrect edges are the edges present or absent, respectively, in the KEGG pathway and half-correct edges are those that either capture indirect effects or the reverse direction of interaction. For the same number of edges predicted from the CDC28 dataset alone, our model is able to predict 7 correct, 5 half-correct, and 3 incorrect edges (Fig. 9c).

Since each edge in our model is associated with a p value, a straightforward method of integrating the results from all of the datasets is to pool the predicted edges from different datasets and re-rank them by their p values. Integrating the interactions predicted from each of the datasets in this fashion increases the number of correctly predicted edges to 8 and decreases the number of half-correct predictions to 4 (Fig. 9d). Each dataset provided support for a different but overlapping set of interactions, where three of the interactions (SWI6 → SWI4 SWI6 → MBP1, and SWI4 → CLN2) were determined highly significant across all datasets. The performance of our method when trained on individual datasets and when trained on all four datasets is summarized in Table 2. Excluding ELU and training on the remaining three datasets did not affect the network reconstruction performance.

Next we compare the performance of our method (SMLR) to those of other methods. DN, DBN, and FuzzyNet have reported 14, 15, and 36 predicted interactions, respectively. For direct comparison, we generated three networks by varying the cutoff p value in SMLR, such that the same numbers of edges are obtained. SMLR achieves better precision, recall, and F-measure values when compared with these methods (Fig. 10). Particularly, the predictions made by SMLR are at least twice more precise and complete when compared with the same number of predictions made by BN and DBN. FuzzyNet makes a larger number of predictions than BN and DBN and performs slightly worse than SMLR for the same number of predictions.

Note that our method is additionally able to rank the predicted interactions using their associated statistical significance values, such that any desired number of interactions can be generated. The precision–recall curves of the predictions made by our method for varying p values are shown in Fig. 11. Integrated predictions outperform predictions from individual datasets in precision, up to a recall of 20 %. We attribute this partially to our integration strategy, which focuses on collecting predictions with high statistical significance from individual datasets, biasing the improvement to the top predictions. The performance of our method is slightly better than that of FuzzyNet for comparable precision and recall values.

In addition to the comparison of SMLR to the methods that are suitable for time-series data, we also compared SMLR to the methods that use steady-state microarray data, including ARACNE (Margolin et al. 2006a, b), which is a state-of-art method based on mutual information (MI) calculation. Here, ARACNE was used to reconstruct the regulatory network using the same four datasets, where the microarray samples at each time point in the time series were regarded as different steady-state samples. The edges predicted by ARACNE were sorted by their associated MI scores. Besides ARACNE, we also attempted to reconstruct the network by calculating the gene expression correlation between each pair of genes. The edges representing the gene pairs were sorted by the p value of the correlation. Since the results of ARACNE and correlation calculation

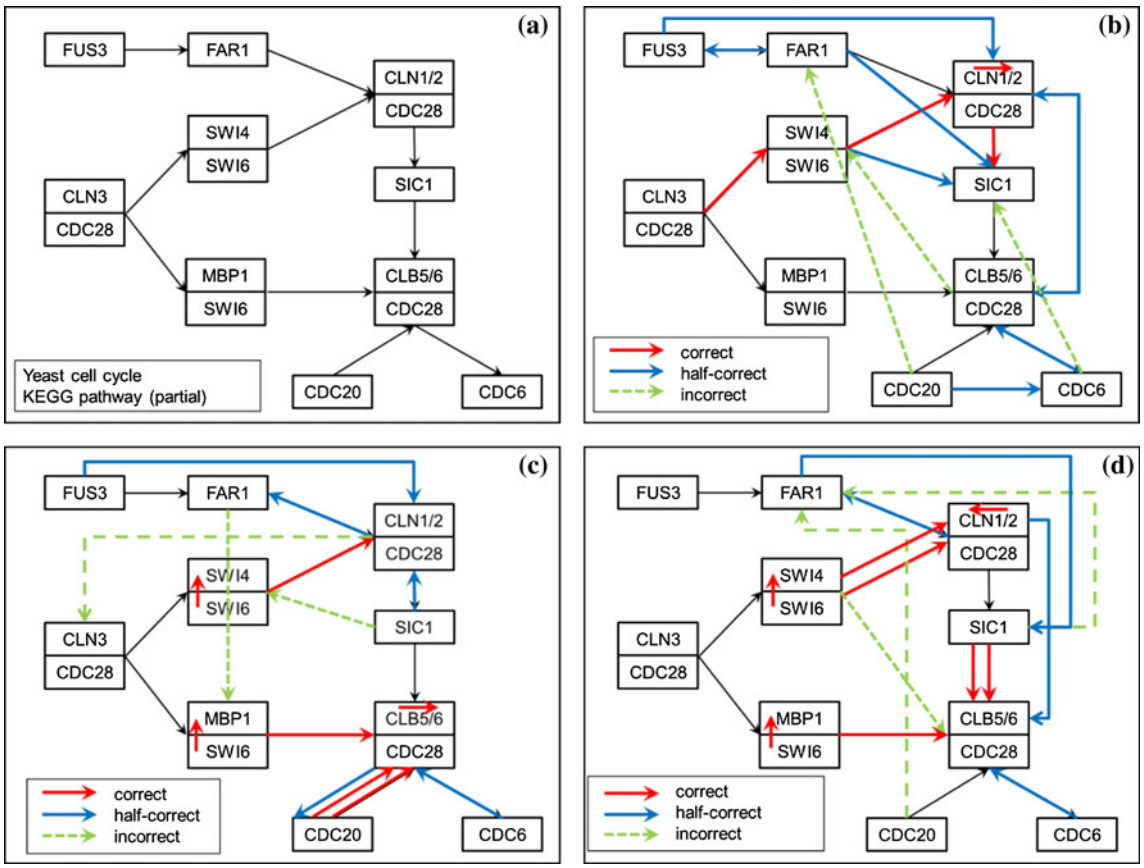

Fig. 9 Regulatory network reconstruction. **a** Sub-network extracted from yeast cell-cycle pathway obtained from KEGG. The KEGG pathway contains 51 edges in total; multiple edges between covarying modules are not displayed here. **b** Regulatory interactions predicted by the DBN model (Kim et al. 2003, 2004). **c** Interactions predicted by a model trained on the CDC28 dataset. **d** Integration of predictions from the four datasets

Table 2 Prediction performance of our method for each of the four different datasets separately and for integrating the results from all of the training datasets

Dataset(s)	Precision (%)	Recall (%)	*F*-measure (%)
CDC28	46.7	13.7	21.2
CDC15	33.3	9.8	15.1
ALPHA	33.3	9.8	15.1
ELU	26.7	7.8	12.1
Integrated	53.3	15.6	29.9

Evaluations in this table are based on the top 15 most significant edges predicted from each dataset

lack the edge directionality, for the purpose of comparison we consider the presence of an edge as correct if the edge is observed in the known network, without regarding its direction. Figure 12 demonstrates that the performance of SMLR using time-series data is superior to that of both ARACNE and correlation-based reconstruction. This indicates that utilizing the time-series data as a dynamic and dependent set of measurements instead of static independent samples results in a more reliable reconstructed network.

In order to further evaluate how well the predicted network is statistically supported from the data, we performed random permutations of the time points and analyzed the resulting predicted interactions (Fig. 13). The integrated predictions perform consistently better than randomly permuted datasets, at two standard deviations better precision than randomized datasets. This shows that the predictions made by our method are not simply due to spurious expression patterns in the dataset due to noise or systematic errors. On the other hand, predictions from individual datasets degrade quickly, and one can be confident of their accuracy only for the top few best predictions. Figure 13 also demonstrates the effectiveness of using the p value for ranking the predictions, as concluded from the general trend of the overall monotonicity in the reduction of the precision as more edges are predicted. Surprisingly, the performance of the DBN model is close to the results obtainable by our method for randomized data indicating that the results of DBN may not be statistically supported from the datasets.

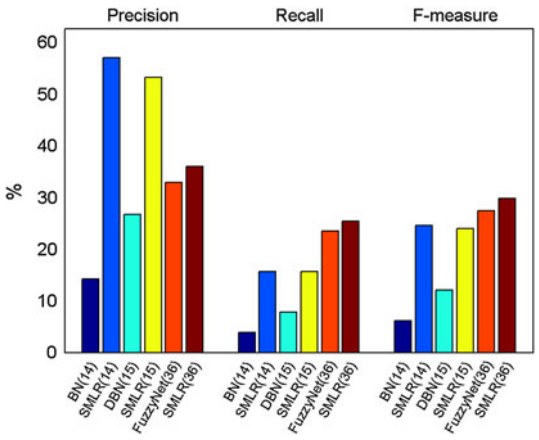

Fig. 10 Comparison of network reconstruction performance for SMLR and other methods. The number of estimated interactions reported by each method is indicated in the *parentheses*. The p value threshold of SMLR was adjusted to generate three networks, such that the same number of edges is reported with the method it is being compared to

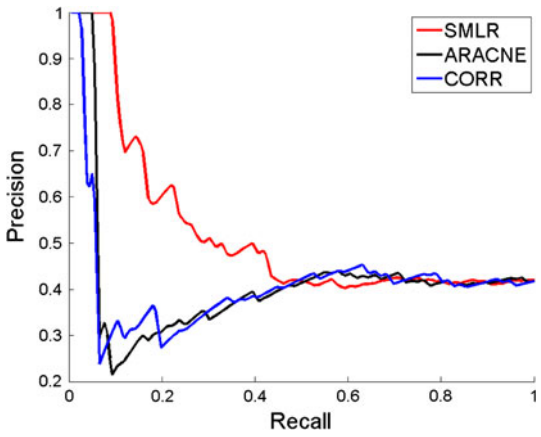

Fig. 12 Comparison of our method (SMLR) to ARACNE and the correlation-based reconstruction (CORR). Note that unlike the results reported in Fig. 11. The direction of the edges is disregarded and the interactions predicted by SMLR in either direction were considered as correct. ARACNE and CORR only report un-directed interactions

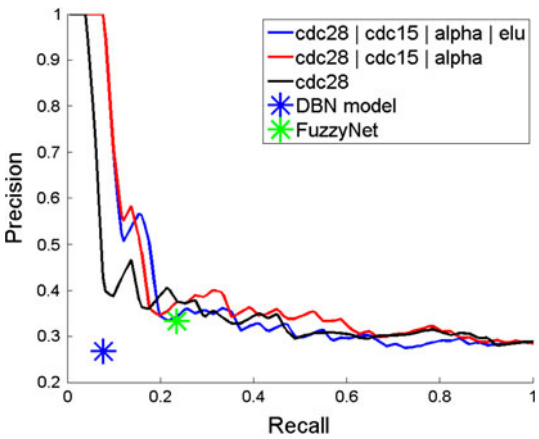

Fig. 11 Comparing the precision–recall curves for our method with that of others. Results from our method on integrating all datasets, excluding ELU, and using only CDC28 are shown; other individual datasets are omitted for clarity

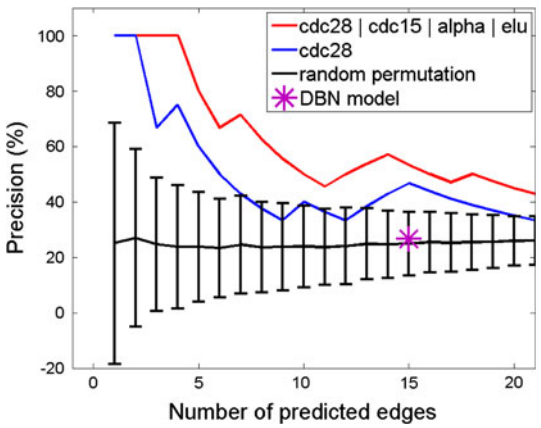

Fig. 13 Comparison of predictions of our method to its predictions from randomized data. *Error bars* for the recall of the randomly permuted datasets show its standard deviation in the 100 random trials

Another important advantage of our approach over existing methods is the interpretability of the inferred coefficients as the strength of the interactions. We have listed the coefficients for the top 15 predicted interactions in Table 3. There are currently no quantitatively annotated datasets for regulatory networks, so we are not able to validate the magnitude of these coefficients directly. On the other hand, the KEGG pathway contains information regarding whether an interaction activates or inhibits the target gene. We observe that the signs of the correctly predicted coefficients match for some of the top predictions. The positive sign of the half-correct interaction FAR1 $\rightarrow$ SIC1 maps to two consecutive inhibitory interactions in the KEGG pathway (FAR1 $\rightarrow$ CLN1/2, CDC28 $\rightarrow$ SIC1), which effectively makes it an activating interaction.

4 Discussion

In this paper, we have employed a multiple linear regression model to predict and simulate time-series microarray data and also to reconstruct gene regulatory networks from this model. Linear models provide a compelling alternative to other existing approaches due to their simplicity, robustness against noise, and low computational requirements. Our approach introduces two additional parameters, in addition to the coefficients estimated in the linear model.

Table 3 Coefficients and p values of the predicted interactions from integrated 4 datasets

Source gene	Target gene	Accuracy	p value (log10)	Coefficient	Sign correct
SWI4	CLN2	Correct	−7.20	1.36	Yes
SIC1	CLB6	Correct	−5.79	0.92	No
SWI4	CLN1	Correct	−5.73	1.02	Yes
SWI6	SWI4	Correct	−5.12	−1.83	No, co-regulated
FAR1	SIC1	Half-correct	−5.04	0.77	Yes (indirect)
SWI4	CLB6	Incorrect	−4.95	2.24	–
SWI6	MBP1	Correct	−4.70	0.94	Yes, co-regulated
CDC6	CDC28	Half-correct	−4.29	0.40	Yes
CLN2	FAR1	Half-correct	−4.24	−0.65	Yes
SIC1	FAR1	Incorrect	−4.20	0.69	–
CDC20	FAR1	Incorrect	−4.06	0.58	–
SIC1	CLB5	Correct	−4.03	0.46	No
CLN2	CLN1	Correct	−4.03	0.64	Yes, co-regulated
SWI6	CLB5	Correct	−3.99	−2.29	No
CLN1	CLB6	Half-correct	−3.94	−1.55	No

The sign of the coefficients is compared against the interactions available in the KEGG yeast cell-cycle pathway. Incorrect predictions naturally do not have corresponding information in KEGG. For co-regulated genes, we considered an activating relationship to be correct

Specifically, we have shown that the number of prior time points used to train the model and the p value cutoff of genes to include in the gene expression prediction function can be determined empirically from the training data. We have demonstrated that the proposed model is able to make correct predictions for the yeast cell-cycle pathway, and simulate the expressions of the genes involved. The predicted gene expressions showed similar cyclic behavior and similar clustering, when compared with the real data. The linear model presented here is able to model the presence, directionality, and the strength and sign of the interactions in a reconstructed regulatory network. This is an important advantage over most of the existing methods that at best predict the directionality of the interactions.

The statistical significance associated with each predicted interaction provides a convenient way of assessing the reliability of the prediction. Given that most computational prediction approaches to biological problems aim to produce new hypotheses that can be validated with further biological experiments, the prioritization of the predictions becomes an invaluable feature for these time- and labor-intensive and low-throughput downstream experiments. The statistical significance also provides a straightforward means of integrating multiple time-series datasets, collected under different experimental conditions and time scales. Whereas very short time intervals mean that consecutive time points may not reveal regulatory interactions, longer time points risk missing the regulatory window of action. While each regulatory interaction is likely to operate at different time scales, the integration of the datasets with varying time intervals would be able to collect such interactions into a single predicted network.

Although the network reconstruction was robust to the heterogeneity of the training datasets, the simulation of the time-course data was sensitive to the time intervals of these datasets. Of the four datasets used in this study, the elutriation dataset (ELU) was collected at a 30-min time interval, which was three times longer than any of the other datasets. Inclusion of this data did not prevent the model from capturing the cyclic behavior of genes; however, our simulation contained a phase shift compared to the real data. When the elutriation dataset was included in the training (or testing) set, our model predicted changes in the gene expression to occur at earlier (or later) times than they actually occurred in the real data. We conclude that the model should be trained with data collected at similar time intervals to the testing data in order to achieve better performance. Approaches to interpolate the expression levels and thus artificially generate new datasets with the same time interval may be pursued as a potential solution when dataset exclusion is not desirable. In particular, the datasets can be re-sampled from a continuous representation using linear interpolation (Aach and Church 2001) or spline interpolation (Bar-Joseph et al. 2003a, b). These continuous representations additionally allow re-alignment of datasets to minimize the effects of varying phase and periodicity of the datasets. Such dataset integration methods will be especially useful pre-processing steps when the method introduced in this paper is applied to large-scale, heterogeneous datasets.

In order to identify the predictor genes and fit the model parameters to the data, we have used a stepwise multiple linear regression with a forward selection strategy. This greedy stepwise optimization strategy may not discover a globally optimal solution. Using more comprehensive sampling approaches such as Monte Carlo methods (Berg 2004), or utilizing related model fitting methods, such as ridge regression (Hoerl and Kennard 1970; Marquardt and Snee 1975) and partial least squares regression (Lindgren et al. 1993) may improve the model fitting and consequently increase the accuracy of the reconstructed regulatory network, at the cost of increased training time. Known regulatory interactions can also be incorporated as constraints in the search and sampling of predictors during the model fitting stage. Incorporation of known transcription factors improves network reconstruction (Yao et al. 2010); consequently, the predictors in our model fitting can be limited to the set of known transcription factors to improve the reconstruction accuracy.

It may be argued that using a linear model for representing regulatory interactions is incorrect or limited.

While in this study we do not claim that a linear model should represent the kinetics of regulatory interactions, we have shown that in the context of expression prediction, time-course simulation, and network reconstruction problems, the linear model provides a sufficient approximation to the otherwise complex regulatory interactions. Furthermore, using more complex functional forms would incur a larger number of parameters that need to be estimated from the data, bringing the sufficiency of the available data into question.

In evaluating the accuracy of different methods, we used the interactions available in the KEGG pathways as the ground truth. We acknowledge that future discoveries may change the known interactions in the cell-cycle pathway investigated in this study, and alter the evaluations presented in this paper. We also expect that the discrepancies between our predictions and currently known interactions may guide such new discoveries. Furthermore, the view that interactions between pairs of genes should be an either always or never phenomena is limiting, since gene regulation is dynamic and certain interactions may be present only under certain temporal and experimental conditions. The investigation of interactions as emerging or disappearing relationships and the predictions of these dynamic behaviors have attracted recent attention (Almansoori et al. 2012).

To conclude, we demonstrated our approach on a relatively small dataset and compared its results to those from Bayesian Network, dynamic Bayesian Network (Kim et al. 2004) and Fuzzy Neural Network (Abul et al. 2006) models. Our method generally produced a lower mean squared error for the simulated data than the neural fuzzy network method. We also achieved better accuracy than these methods in reconstructing the yeast cell-cycle pathway. These early comparisons are promising; however, a large-scale evaluation using a more comprehensive set of synthetic and real datasets and different types of reconstruction methods as well as handling differences in sampling rates is left for future work. Finally, we note that it may be possible to develop a meta-method that combines the predictions of various methods into a single improved regulatory network.

References

Aach J, Church GM (2001) Aligning gene expression time series with time warping algorithms. Bioinformatics 17(6):495–508

Abul O, Alhajj R, Polat F (2006) Asymptotical lower limits on required number of examples for learning Boolean networks computer and information sciences. In: Levi A, Savas E, Yenigün H, Balcisoy S, Saygin Y (eds) ISCIS 2006, Springer, Berlin, vol 4263, pp 154–164

Almansoori W, Gao S, Jarada T, Elsheikh A, Murshed A, Jida J, Alhajj R, Rokne J (2012) Link prediction and classification in social networks and its application in healthcare and systems biology. Netw Model Anal Health Inform Bioinformatics, 1–10

Andrecut M, Huang S, Kauffman SA (2008) Heuristic approach to sparse approximation of gene regulatory networks. J Comput Biol 15(9):1173–1186

Bansal M, Della Gatta G, di Bernardo D (2006) Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. Bioinformatics 22(7):815–822

Bar-Joseph Z, Gerber GK, Gifford DK, Jaakkola TS, Simon I (2003a) Continuous representations of time-series gene expression data. J Comput Biol 10(3–4):341–356

Bar-Joseph Z, Gerber G, Simon I, Gifford DK, Jaakkola TS (2003b) Comparing the continuous representation of time-series expression profiles to identify differentially expressed genes. Proc Natl Acad Sci 100(18):10146–10151

Basso K, Margolin AA, Stolovitzky G, Klein U, Dalla-Favera R, Califano A (2005) Reverse engineering of regulatory networks in human B cells. Nat Genet 37(4):382–390

Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc: Ser B (Methodol) 57(1):289–300

Berg BA (2004) Markov Chain Monte Carlo simulations and their statistical analysis (with Web-based Fortran code). World Scientific, Hackensack

Chen T, He HL, Church GM (1999) Modeling gene expression with differential equations. Pac Symp Biocomput 1999:29–40

Cho RJ, Campbell MJ, Winzeler EA, Steinmetz L, Conway A, Wodicka L, Wolfsberg TG, Gabrielian AE, Landsman D, Lockhart DJ et al (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. Mol Cell 2(1):65–73

di Bernardo D, Thompson MJ, Gardner TS, Chobot SE, Eastwood EL, Wojtovich AP, Elliott SJ, Schaus SE, Collins JJ (2005) Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. Nat Biotechnol 23(3):377–383

Draper N, Smith H (1998) Applied regression analysis (Wiley Series in Probability and Statistics). Wiley, Hoboken

Faith JJ, Hayete B, Thaden JT, Mogno I, Wierzbowski J, Cottarel G, Kasif S, Collins JJ, Gardner TS (2007) Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. PLoS Biol 5(1):e8

Fan X, Shi L, Fang H, Cheng Y, Perkins R, Tong W (2010) DNA microarrays are predictive of cancer prognosis: a re-evaluation. Clin Cancer Res 16(2):629–636

Friedman N, Linial M, Nachman I, Pe'er D (2000) Using Bayesian networks to analyze expression data. J Comput Biol 7(3–4): 601–620

Gardner TS, Faith JJ (2005) Reverse-engineering transcription control networks. Phys Life Rev 2(1):65–88

Gardner TS, di Bernardo D, Lorenz D, Collins JJ (2003) Inferring genetic networks and identifying compound mode of action via expression profiling. Science 301(5629):102–105

Golub TR, Slonim DK (1999) Molecular classification of cancer: class discovery and class prediction by gene expression. Science 286(5439):531

Guelzim N, Bottani S, Bourgine P, Kepes F (2002) Topological and causal structure of the yeast transcriptional regulatory network. Nat Genet 31(1):60–63

Hadi SCAS (2006) Regression analysis by example, 4th edn. Wiley, New York

Hartemink AJ, Gifford DK, Jaakkola TS, Young RA (2001) Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. Pac Symp Biocomput 2001:422–433

Hecker M, Lambeck S, Toepfer S, van Someren E, Guthke R (2009) Gene regulatory network inference: data integration in dynamic models—a review. Biosystems 96(1):86–103

Hoerl AE, Kennard RW (1970) Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12(1):55–67

Jeffery IB, Higgins DG, Culhane AC (2006) Comparison and evaluation of methods for generating differentially expressed gene lists from microarray data. BMC Bioinformatics 7:359

Kanehisa M, Goto S (2000) KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res 28(1):27–30

Kauffman SA (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. J Theor Biol 22(3):437–467

Andrecut MA, Kauffman SA (2006) Mean-field model of genetic regulatory networks. New J Phys 8(148)

Kim SY, Imoto S, Miyano S (2003) Inferring gene networks from time series microarray data using dynamic Bayesian networks. Brief Bioinformatics 4(3):228–235

Kim S, Imoto S, Miyano S (2004) Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data. Biosystems 75(1–3):57–65

Lindgren F, Geladi P, Wold S (1993) The kernel algorithm for PLS. J Chemom 7(1):45–59

Luscombe NM, Madan Babu M, Yu H, Snyder M, Teichmann SA, Gerstein M (2004) Genomic analysis of regulatory network dynamics reveals large topological changes. Nature 431(7006): 308–312

Margolin A, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera R, Califano A (2006a) ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics 7(Suppl 1):S7

Margolin AA, Wang K, Lim WK, Kustagi M, Nemenman I, Califano A (2006b) Reverse engineering cellular networks. Nat Protoc 1(2):662–671

Marquardt DW, Snee RD (1975) Ridge regression in practice. Am Stat 29(1):3–20

Mutch DM, Berger A, Mansourian R, Rytz A, Roberts MA (2002) The limit fold change model: a practical approach for selecting differentially expressed genes from microarray data. BMC Bioinformatics 3:17

Nachman I, Regev A, Friedman N (2004) Inferring quantitative models of regulatory networks from expression data. Bioinformatics 20(Suppl 1):i248–i256

Opgen-Rhein R, Strimmer K (2007) From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data. BMC Syst Biol 1(1):37

Quackenbush J (2002) Microarray data normalization and transformation. Nat Genet 32(Suppl):496–501

Rangel C, Angus J, Ghahramani Z, Lioumi M, Sotheran E, Gaiba A, Wild DL, Falciani F (2004) Modeling T-cell activation using gene expression profiling and state-space models. Bioinformatics 20(9):1361–1372

Sakamoto E, Iba H (2001) Inferring a system of differential equations for a gene regulatory network by using genetic programming. In: Proceedings of the 2001 congress on evolutionary computation, 2001, vol 721, pp 720–726

Segal E, Shapira M, Regev A, Pe'er D, Botstein D, Koller D, Friedman N (2003) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nat Genet 34(2):166–176

Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, Eisen MB, Brown PO, Botstein D, Futcher B (1998) Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. Mol Biol Cell 9(12):3273–3297

Stuart JM, Segal E, Koller D, Kim SK (2003) A gene-coexpression network for global discovery of conserved genetic modules. Science 302(5643):249–255

Thieffry D, Huerta AM, Pérez-Rueda E, Collado-Vides J (1998) From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in *Escherichia coli*. BioEssays 20(5): 433–440

Tusher VG, Tibshirani R, Chu G (2001) Significance analysis of microarrays applied to the ionizing radiation response. Proc Natl Acad Sci USA 98(9):5116–5121

van de Vijver MJ, He YD, Van't Veer LJ, Dai H, Hart AAM, Voskuil DW, Schreiber GJ, Peterse JL, Roberts C, Marton MJ et al (2002) A gene-expression signature as a predictor of survival in breast cancer. N Engl J Med 347(25):1999–2009

van Someren EP, Wessels LF, Reinders MJ (2000) Linear modeling of genetic networks from experimental data. Proc Int Conf Intell Syst Mol Biol 8:355–366

Weaver DC, Workman CT, Stormo GD (1999) Modeling regulatory networks with weight matrices. Pac Symp Biocomput 1999: 112–123

Wong DJ, Chang HY (2005) Learning more from microarrays: insights from modules and networks. J Investig Dermatol 125(2): 175–182

Yao F, Jarboe LR, Dickerson JA (2010) Gene regulatory network reconstruction based on gene expression and transcription factor activities. In: BIOCOMP: 2010, pp 113–119

Zhu G, Spellman PT, Volpe T, Brown PO, Botstein D, Davis TN, Futcher B (2000) Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth. Nature 406(6791):90–94