Inferring Gene Regulatory Networks from Time-Series Expressions using Random Forests Ensemble

D.A.K. Maduranga[1], Jie Zheng[1,2], Piyushkumar A. Mundra[1], and Jagath C. Rajapakse[1,3,4]

[1] Bioinformatics Research Center, School of Computer Engineering,Nanyang Technological University, Singapore 639798.
[2] Genome Institute of Singapore, Biopolis Street, Singapore 138672.
[3] Singapore-MIT Alliance, Singapore.
[4] Department of Biological Engineering, Massachusetts Institute of Technology, USA
asjagath@ntu.edu.sg

Abstract. Reconstructing gene regulatory network (GRN) from time-series expression data has become increasingly popular since time course data contain temporal information about gene regulation. A typical microarray gene expression data contain expressions of thousands of genes but the number of time samples is usually very small. Therefore, inferring a GRN from such a high-dimensional expression data poses a major challenge. This paper proposes a tree based ensemble of random forests in a multivariate auto-regression framework to tackle this problem. The efficacy of the proposed approach is demonstrated on synthetic time-series datasets and Saccharomyces cerevisiae (Yeast) microarray gene expression data with 9-genes. The performance is comparable or better than GRN generated using dynamic Bayesian networks and ordinary differential equations (ODE) model.

Keywords: Gene regulatory networks, time-series gene expression data, gene regulation, Random forests, multivariate auto-regression, regression trees

1 Introduction

A set of genes, transcription factors (regulators), mRNAs, and gene products (protein) interact among themselves to control almost all biological activities and form a gene regulatory network (GRN). Therefore, reverse engineering of GRN from gene expression data becomes an important problem. Reconstruction of regulatory networks plays a vital role in understanding of complexity, functionality and pathways of the biological systems and plays a crucial role in developing novel drugs for disease. With recent advancements of microarray technology and next generation sequencing, a vast amount of expression data

has been produced. Thereafter, developments of novel computational models to infer the GRN from gene expression measurements have been more feasible.

Microarray technology enables us to gather both steady-state and time series gene expression data. Gene regulatory interactions among genes are not instantaneous, but they are dynamic events which occur throughout a period of time [1]. Therefore, time-series expression data are vital in studying the dynamics of the underlying biological systems. A typical time series data contains only a few time samples compared to the number of genes, and hence, inference of regulatory interaction of large number of genes from a few time points is one of the biggest challenges faced by computational biologists.

Several computational techniques have been proposed to infer GRN by using time course gene expression data. Boolean networks are the simplest and earliest models of gene networks [2, 3]. Some of biological characteristics of actual GRN are illustrated by the Boolean network models [4]. On the other hand, ordinary differential equations (ODE) [5] are able to describe dynamic changes of the regulatory network and capture complex regulatory dependencies among the expression data. However, their major disadvantage is having a high-dimensional parameter space. Therefore, they require a large amount of experimental data to infer the accurate regulatory network. Dynamic Bayesian networks (DBN) based models are also popular in reconstructing GRN as they are capable of learning causal interactions among the temporal gene expressions [1],[6],[7]. Another approach is the usage of information theoretical measures such as mutual information (MI) to model the time course expression data. TimeDelay-ARACNE [8] is one of the recently proposed algorithms using MI among gene expressions. Also, several linear multivariate vector auto-regression (MVAR) techniques such as lasso regression, elastic net and ridge regression have been introduced in literature to infer GRN [9, 10].

However, the performance of GRN inference techniques is still poor because the current approaches are unable to capture the complex regulatory interactions among the genes and many of these approaches are incapable of handling high-dimensional microarray expression data. Within this context, we propose an effective approach to infer GRN from time-course expression data with ensemble of random forest. Random forest method has become popular in handling high-dimensional problems [11], [12], [13], [14]. Huynh-Thu et al initially applied random forests technique to build GRN [15]. Their proposed method, namely GENIE3, showed the significant improvement in accuracy of GRN inference and it was the best performer in the DREAM4 *In Silico* Multifactorial challenge [15]. However, experiments were only performed with steady-state gene expression data(static data). Also the structure of the GRN was not built, but only provided the ranking of gene regulatory links. On the other hand, sparse linear regression based MVAR approaches has inherent limitations in modeling non-linear regulations. In this paper, to tackle the limitation of these previous approaches, we develop a random forests based MVAR approach to infer a GRN from time-series gene expression data. Using variable importance criterion de-

rived from training random forest model and subsequently using adjusted R^2, a structure of GRN is obtained using time-series gene expression data.

The rest of the paper is organized in three sections. First, Section 2 describes the inference of GRN from time-course expression data using the tree based ensemble method of Random forests. Section 3 provides details on both synthetic and real datasets, performance metrics used in the evaluation, present the results and time complexity of the proposed approach. Finally, Section 4 concludes the paper with a discussion on obtained results along with future research directions.

2 Method

Let $(x_t^j)_{j=1}^q$ be a vector containing the gene expressions of q genes at the tth time point. Let x_t^{-j} is a vector containing gene expressions at time t of all the genes except gene j. By assuming that the expression level of given gene (j) at next time point $(t+1)$ is a function (g_j) of the expression values of other genes at current time (t), we can write

$$x_{t+1}^j = g_j(x_t^{-j}) + \epsilon_t, \forall t \tag{1}$$

where ϵ_t denotes the random noise. The static version of GRN inference with random forest assumes that the expression value of each gene depends on expression values of other genes for a given experiment(k) [15]:

$$x_k^j = f_j(x_k^{-j}) + \epsilon_k, \forall k \tag{2}$$

where x_k^{-j} is a vector containing all static gene expression data except expression data of gene j in the k^{th} experiment. The network inference procedure first decomposes the problem of recovering network structure of q genes into q different sub-problems. The j^{th} sub-problem is equivalent to finding regulators for j^{th} gene. Each sub problem has its own learning sample (LS_T^j) which is consists of input-output pairs for gene, $LS_T^j = (x_t^{-j}, x_{t+1}^j)_{t=1}^{T-1}$. Here, T denotes the total number of time points in the time series. Each sub-problem can be solved by finding an optimal function for g_j that minimizes the square error loss between the actual expression level and the predicted expression level by the function as follows:

$$\sum_{t=1}^{T-1} (x_{t+1}^j - g_j(x_t^{-j}))^2 \tag{3}$$

Each of these sub-problems can be categorized as supervised regression problem [15]. Regression problem which is defined by Eq. (3) can be solved by constructing tree models such as regression trees [16]. Accuracy of the single tree is further improved by ensemble methods where prediction outcomes of several individual trees are merged. Ensemble methods provide a combine prediction by considering all individual predictions in the ensemble. Therefore, the tree based ensemble method of random forest [11] is suitable for solving above problem because it can handle high dimensional expression data [13], and is capable of

learning non-linear relationships as well as dealing with interacting features [15]. So, each sub-problem is solved by building an ensemble consists of regression trees using random forest method. On the other hand, proposed method can be identified as another way of solving sparse autoregressive model where function g_j is assumed to be a linear function of the regression coefficients (β) [9, 10].

First step of the random forest is generation of bootstrap samples from the initial input data. Then, each tree is constructed by using these samples. But tree building process is little bit different than the normal process because at each node, N numbers of predictors are randomly selected from the bootstrap sample to determine the optimal split for the node. The value of N is the tuning parameter because it determines the level of randomization of the trees. All the trees of an ensemble are built by applying above process.

Function g_j is learned from the learning sample LS_T^j using random forest ensemble. Following [15], weight for having a regulatory link from any gene i to j ($w_{i,j}$) are obtained by computing variable importance measure using following equation:

$$I = \#S.Var(S) - \#S_t.Var(S_t) - \#S_f.Var(S_f) \tag{4}$$

where S indicates the input data sample that reach the node, # shows the cardinality of data sample, S_f and S_t shows the subset of samples out of input data sample (S) that the test is false and true, respectively. For each subset of samples (S_f and S_t), the variance of the target variable is indicate by $Var(.)$. Variable importance measure provides an indication about the relevance of an input variable for the prediction of the output. After that, regulatory links are ranked based on their weights for each learning sample. Regulatory links that have higher weights are more likely to be actual regulatory interactions. Therefore, we apply adjusted coefficient of determination (Adjusted R^2) which is given by Eq. (5) to each sub problem to determine the actual regulators.

$$\text{Adjusted coefficient of determination } = 1 - (1 - R^2)\frac{n-1}{n-p-1} \tag{5}$$

where n denotes the size of the learning sample, p is the number of regressors in the model and R^2 is the coefficient of determination. In our case, n equals to q. An important property of adjusted R^2 is that when a regression variable is added into the model , adjusted R^2 increases if added variable improves the prediction ability of the model, otherwise the value of adjusted R^2 decreases [17]. So, for each sub-problem, we add regulators into the model from highest weight to lower one and each time the value of adjusted R^2 is computed. If added regulator increases adjusted R^2, we consider it as an actual regulator. We continue adding more regressor until adjusted R^2 starts to decrease. This way, we determine the actual regulators for each sub problem.

3 Experiments and Results

Several synthetic gene expression datasets were generated and used to evaluate the performance of the proposed method. Many gene regulatory network infer-

ence studies with synthetic datasets were done using scale-free synthetic networks that were obtained using Barabasi-Albert model [18]. But in this study, we used GeneNetWeaver (GNW) [19] software package to extract sub-networks from global Escherichia coli (E. Coli) network. Sub-networks of having 10, 30, 50 and 100 genes were extracted from E. Coli network. Topology or the structure of the gene regulatory network which has q number of genes is depicted by the connectivity matrix $M = \{M_{ij}\}_{q \times q}$ where $M_{ij} = 1$ for the presence of connection between gene i and j, and $M_{ij} = 0$ for the absence. These network topologies were used in the section 3.1 to generate synthetic gene expression data. Other than synthetic data, real time-course gene expression dataset were also used to evaluate the performance of the proposed method.

3.1 Synthetic expression data generation

First-order multivariate vector autoregressive model (MVAR) [10],[9] is used to generate synthetic time-series gene expression data. Sub-networks extracted from GNW were used as network topologies in MVAR model to simulate the expression data. Gene expression at time t were obtained by using the first order MVAR model as follows:

$$x_t = x_{t-1} \times M_{weight} + \epsilon_t \tag{6}$$

where $x_t = (x_t^j)_{j=1}^q$ indicates the expressions of q number of genes at time t and ϵ_t denotes the added Gaussian random noise to the gene expression at time t. Matrix M_{weight} is obtained by assigning weights randomly to all the connection (where $M_{ij} = 1$) in the connectivity matrix M. These weights were assigned by getting the values from uniform distribution on the interval [-1,-0.6] and [0.6, 1]. Two intervals are chosen to maintain the amount of negative and positive weights nearly equal [10]. Gene expression vector at $t = 0$ ($x_{t=0}$) is initialized by obtaining the samples from the uniform distribution on the interval [0, 1] and subsequent time points are simulated using Eq. (6). For each network topology, three synthetic datasets which have 10, 30 and 50 time points were generated. For each combination of genes and time points, 50 different datasets were generated.

3.2 Real Dataset

Performance evaluation of GRN inference techniques on real gene expression data is more difficult because of lack of experimentally verified ground truth gene networks. In this study, we choose an experimentally identified gene regulatory network which is related to yeast Saccharomyces cerevisiae cell cycle [20]. This real gene regulatory network is depicted in figure 1(a) and consists of 9 genes (Fkh2, Swi4, Swi5, Swi6, Ndd1, Ace2, Cln3, Mbp1, Mcm1). Real time-series gene expression data were obtained from Spellman [21] dataset. Spellman dataset contains expression data of yeast cell cycle regulation. We selected time-course gene expression data from cdc28 cell cycle arrest which consists of 17 time points.

3.3 Performance

We generated synthetic datasets using MVAR model with the network topologies which were extracted from GNW software. Therefore, true structure of extracted gene regulatory networks is known. Also in the real data, true structure is available since we used an experimentally verified regulatory network. Hence, we compared GRN which was inferred by the proposed random forest based approach with the ground truth network to evaluate the performance. In synthetic data, there were 50 time series datasets for each combination of genes and time points, resulting in 50 inferred GRNs. Number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) were computed for each predicted network by comparing predicted network with ground truth network. Then performance measures such as precision[5], recall[6], accuracy[7] and F-measure[8] were calculated.

For both synthetic and real dataset, an ensemble of 1000 trees was constructed. The most important parameter of this method is the number of predictors which were selected randomly to find the best split in each node. This parameter was set to $\sqrt{q}$, where q denotes the number of genes in the network. Table 1 shows the performance of the proposed method with synthetic data. In table 1, the mean and the standard deviation of each performance metric over 50 times simulation are shown. The effectiveness of the proposed method is also shown over real gene-expression data. In order to compare with existing techniques, three techniques, namely the random forest static version, dynamic Bayesian networks with Markov chain Monte Carlo (Dbmcmc software package) [1],[22] and the ordinary differential equation based model (TSNI software package)[23] were applied to the same real dataset. All the packages were used with the default settings according to their user manuals. Table 2 shows the performance measures on real data. In figure 1(b), 1(c), 1(d) and 1(e), we illustrate the gene network structures inferred from real data by the proposed method, random forests static version, ODE and DBN methods respectively. In figure 1, we used solid line to represent the true positive (TP) and dash line to represent the false negatives (FN). False positives are not shown in figure 1, though they were considered in calculating performance metrics in table 2.

3.4 Time complexity

Random forest algorithm has time complexity of $O(Tree_{Total} * N * TlogT)$ [15], where $Tree_{Total}$ represents the number of trees in the ensemble, T denotes the number of time point in the learning sample and N denotes the number of genes that are randomly chosen at each node during construction of each tree. The proposed approach divides the infer of GRN with q number of gene into q number

[5] $Precision = \frac{TP}{FP+TP}$
[6] $Recall = \frac{TP}{FN+TP}$
[7] $Accuracy = \frac{TP+TN}{TP+TN+FN+TP}$
[8] $F - measure = 2 \times \frac{Precision \times Recall}{Precision+Recall}$

Table 1: The performance of the proposed method on synthetic data

Number of genes	Number of time points	Precision	Recall	Accuracy	F-measure
10	10	0.40 ± 0.08	0.50 ± 0.10	0.80 ± 0.03	0.45 ± 0.09
	30	0.58 ± 0.07	0.76 ± 0.09	0.88 ± 0.03	0.66 ± 0.08
	50	0.65 ± 0.07	0.86 ± 0.08	0.90 ± 0.04	0.74 ± 0.07
30	10	0.17 ± 0.03	0.43 ± 0.07	0.86 ± 0.01	0.25 ± 0.04
	30	0.32 ± 0.02	0.80 ± 0.05	0.90 ± 0.01	0.46 ± 0.03
	50	0.36 ± 0.05	0.90 ± 0.04	0.91 ± 0.00	0.52 ± 0.02
50	10	0.14 ± 0.02	0.39 ± 0.04	0.87 ± 0.00	0.21 ± 0.02
	30	0.24 ± 0.02	0.66 ± 0.07	0.89 ± 0.01	0.35 ± 0.04
	50	0.28 ± 0.02	0.78 ± 0.05	0.90 ± 0.00	0.42 ± 0.03
100	10	0.11 ± 0.01	0.30 ± 0.04	0.90 ± 0.00	0.13 ± 0.02
	30	0.14 ± 0.03	0.53 ± 0.03	0.91 ± 0.01	0.22 ± 0.04
	50	0.19 ± 0.02	0.71 ± 0.01	0.93 ± 0.01	0.30 ± 0.02

Table 2: The Performance measures on real data

Method	Precision	Recall	Accuracy	F-measure
Random forests static version	0.25	0.29	0.66	0.27
Random forests dynamic version(proposed method)	0.33	0.40	0.70	0.36
TNSI	0.28	0.29	0.69	0.29
DBN-MCMC	0.26	0.38	0.70	0.30

of sub problems. For each sub problem, we computed a value of adjusted R^2 for all regulators from highest weight to lower one. Therefore, time complexity of each sub problem became $O(q*Tree_{Total}*N*TlogT)$. Since there are altogether q number of sub problems, proposed approach has time complexity of $O(q^2 * Tree_{Total} * N * TlogT)$.

4 Discussion

Building GRN from time-series gene expression data is very important since they contain temporal information about the underline regulatory interactions among genes. In this paper, we have proposed an approach to build GRN using ensemble of random forest. The proposed approach first divides the recovering of regulatory network which is having q genes in to q different supervised regression problems. Then each of these sub problems is solved by applying random forest ensemble method. There are two main contributions of this paper. They are, 1) extend the work of [15] to infer GRN from time-series gene expression data

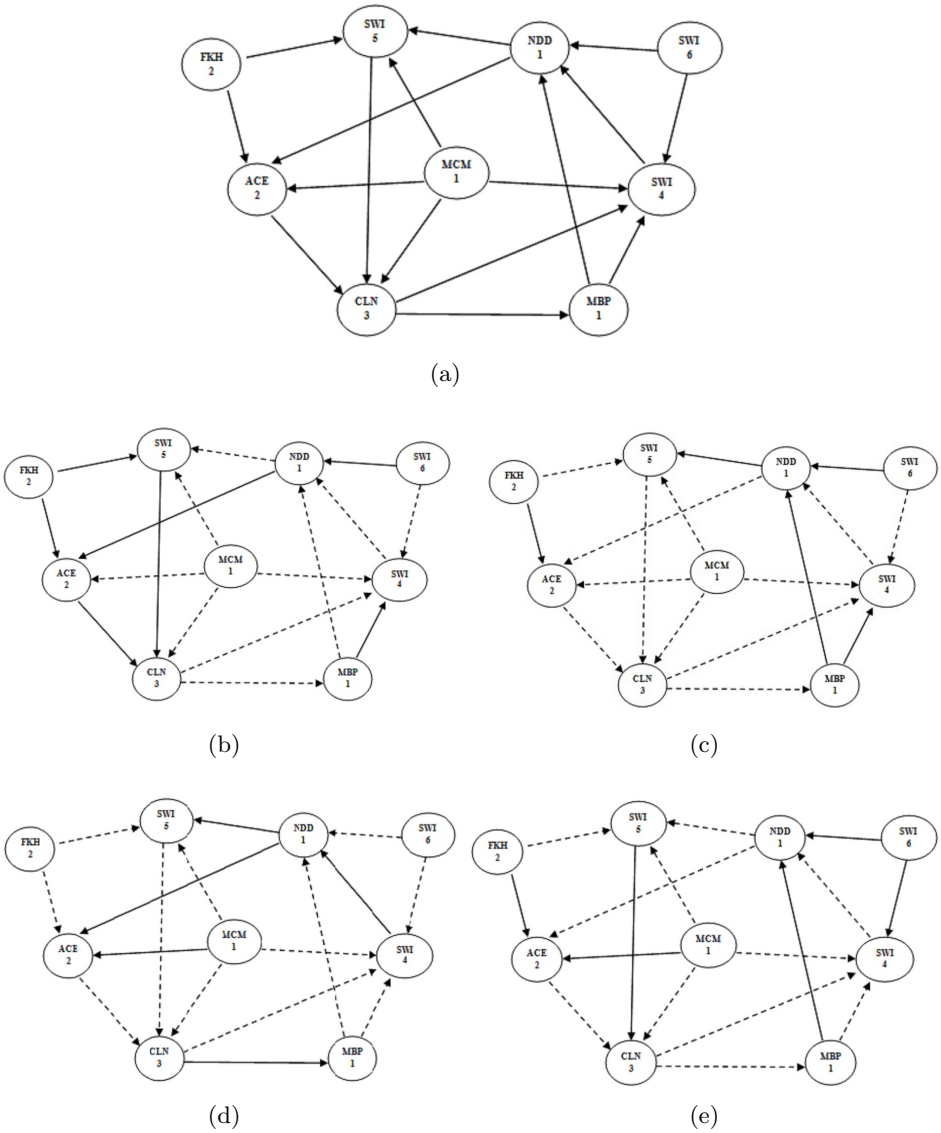

Fig. 1: The GRN identified in Yeast cell cycle and predicted network by various methods. a) is the real GRN related to yeast cell cycle [20]; b) is the predicted network by proposed approach; c) is the predicted network by Random forests static version; d) is the predicted network by TSNI; e) is the predicted network by Dbmcmc.

by developing random forest based MVAR approach and 2) introduce adjusted coefficient of determination to construct the structure of GRN.

The results on synthetic data show that all performance metrics are improved with increase in number of time points and are deteriorated with increase in number of genes. The decrease in the performance of inferred network is due to the inference of large number of false positives than false negatives. Further, the effect of false negatives is corrected quickly than false positive effect with the increased in number of time points in the proposed method. It can also be seen that all the predicted gene networks have more than 80% of accuracy. Figure 1(b) shows the predicted GRN on the real data by the proposed random forest based approach and it is apparent that many true regulatory connections have been identified. As shown in table 2, the proposed method shows better performance on the real data compared to the Random forests static version, DBN with MCMC and ODE method.

Experiments results on both synthetic data and real expression data on a 9-gene network in yeast show the effectiveness of proposed approach. On the other hand, the proposed approach could be improved further. For example, in this study, we assumed that only gene expressions affect the gene regulation. But gene regulation also depends on other mechanisms such as histone modification and transcription factor bindings. Chen et al [24] recently showed that accuracy of DBN can be improved by integrating epigenetic data in to GRN inference. As a future work, similar approaches of data integration with random forest could improve the performance. The proposed approach divides the inference of GRN with q gene into q number of sub-problems. Since each sub-problem is independent of each other, another future work would be to parallelize all these sub-problems to reduce the computation time. Last but not least, similar to [25], the proposed method could be extended to model the time-delayed gene regulations.

Acknowledgments This work is supported by a AcRF Tier 2 grant MOE2010-T2-1-056 (ARC 9/10), Ministry of Education, Singapore.

References

1. Husmeier, D.: Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. Bioinformatics **19**(17) (2003) 2271–2282
2. Bornholdt, S.: Boolean network models of cellular regulation: prospects and limitations. Journal of the Royal Society Interface **5**(Suppl 1) (2008) S85–S94
3. Li, P., Zhang, C., Perkins, E.J., Gong, P., Deng, Y.: Comparison of probabilistic boolean network and dynamic bayesian network approaches for inferring gene regulatory networks. BMC bioinformatics **8**(Suppl 7) (2007) S13
4. Filkov, V.: Identifying gene regulatory networks from gene expression data. Handbook of Computational Molecular Biology (2005) 27–1
5. Liu, B., Thiagarajan, P., Hsu, D.: Probabilistic approximations of signaling pathway dynamics. In: Computational Methods in Systems Biology, Springer (2009) 251–265

6. Kim, S.Y., Imoto, S., Miyano, S.: Inferring gene networks from time series microarray data using dynamic bayesian networks. Briefings in bioinformatics **4**(3) (2003) 228–235
7. Friedman, N., Murphy, K., Russell, S.: Learning the structure of dynamic probabilistic networks. In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc. (1998) 139–147
8. Zoppoli, P., Morganella, S., Ceccarelli, M.: TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach. Bmc Bioinformatics **11**(1) (2010) 154
9. Fujita, A., Sato, J., Garay-Malpartida, H., Yamaguchi, R., Miyano, S., Sogayar, M., Ferreira, C.: Modeling gene expression regulatory networks with the sparse vector autoregressive model. BMC Systems Biology **1** (2007) 39
10. Rajapakse, J.C., Mundra, P.A.: Stability of building gene regulatory networks with sparse autoregressive models. BMC bioinformatics **12**(Suppl 13) (2011) S17
11. Breiman, L.: Random forests. Machine learning **45**(1) (2001) 5–32
12. Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T., Zeileis, A.: Conditional variable importance for random forests. BMC bioinformatics **9**(1) (2008) 307
13. Cutler, A., Cutler, D.R., Stevens, J.R.: Tree-based methods. High-Dimensional Data Analysis in Cancer Research (2009) 1–19
14. Boulesteix, A.L., Janitza, S., Kruppa, J., König, I.R.: Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. (2012)
15. Huynh-Thu, V.A., Irrthum, A., Wehenkel, L., Geurts, P.: Inferring regulatory networks from expression data using tree-based methods. PLoS One **5**(9) (2010) e12776
16. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and regression trees. Chapman & Hall/CRC (1984)
17. Pagano, M., Gauvreau, K., Pagano, M.: Principles of biostatistics. Duxbury Pacific Groveˆ eCA CA (2000)
18. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. science **286**(5439) (1999) 509–512
19. Marbach, D., Schaffter, T., Mattiussi, C., Floreano, D.: Generating realistic in silico gene networks for performance assessment of reverse engineering methods. Journal of Computational Biology **16**(2) (2009) 229–239
20. Simon, I., Barnett, J., Hannett, N., Harbison, C.T., Rinaldi, N.J., Volkert, T.L., Wyrick, J.J., Zeitlinger, J., Gifford, D.K., Jaakkola, T.S., et al.: Serial regulation of transcriptional regulators in the yeast cell cycle. Cell **106**(6) (2001) 697–708
21. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. Molecular biology of the cell **9**(12) (1998) 3273–3297
22. Husmeier, D.: Inferring dynamic bayesian networks with mcmc. http://www.bioss.ac.uk/ dirk/software/DBmcmc/index.html (2003)
23. Bansal, M., Della Gatta, G., Di Bernardo, D.: Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. Bioinformatics **22**(7) (2006) 815–822
24. Haifen, C., Maduranga, D., Mundra, P., Zheng, J.: Integrating epigenetic prior in dynamic bayesian network for gene regulatory network inference. In: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology. (2013) (Accepted).

25. Mundra, P., Niranjan, M., Welsch, R., Zheng, J., Rajapakse, J.: Inferring time-delayed gene regulatory networks using cross-correlation and sparse regression. In: 9th International Symposium on Bioinformatics Research and Applications. (2013) (Accepted).