International Journal of Hybrid Information Technology
Vol.6, No.5 (2013), pp.255-268
http://dx.doi.org/10.14257/ijhit.2013.6.5.24

A Novel Hybrid Framework for Reconstructing Gene Regulatory Networks

Bin Yang[1], Mingyan Jiang[1] and Yuehui Chen*[2]

[1]School of Information Science and Engineering, Shandong University, Jinan, P.R. China 250100
[2]Computational Intelligence Lab, School of Information Science and Engineering, University of Jinan, 106 Jiwei Road, Jinan, P.R. China 250022

**Corresponding author: yhchen@ujn.edu.cn*

Abstract

Much effect has been devoted over the past decade to inference of gene regulatory networks (GRNs). However, the previous methods infer GRNs containing large amount of false positive edges, which could result in awful influence on biological analysis. In this study, we present a novel hybrid framework to improve the accuracy of GRN inference. In our method, network topologies from linear and nonlinear ordinary differential equation (ODE) models are integrated. The additive tree models are proposed for identification of linear/nonlinear models. We also propose a new criterion function that sparse and relevant terms are considered while inferring linear and nonlinear models. Benchmark datasets from Dialogue for Reverse Engineering Assessments and Methods challenge and real biological dataset from SOS DNA repair network in Escherichia coli are used to test the validity of our method. Results reveal that our proposed method can improve the prediction accuracy of GRN inference effectively and performs better than other popular methods.

Keywords: *gene regulatory network, linear/nonlinear models, ordinary differential equation, criterion function, the additive tree models*

1. Introduction

To construct and understand gene regulatory networks has been becoming a major area of interest in the field of systems biology over the last ten years. The advances on high-throughput DNA microarray technologies have enabled the simultaneous measurement of gene expression levels during the developmental processes containing steady state and time-series expression data. With a large of expression levels, it should be theoretically possible to uncover the underlying gene regulatory network [18, 31]. The problem is how to identify the interactions among genes by modeling of gene regulatory networks with complex and noisy gene expression data.

Many models and numerical algorithms have been proposed to infer gene regulatory networks, including the boolean network [1, 4], dynamic bayesian network [25], artificial neural networks [3, 17, 19, 35], the system of differential equations [8, 33, 36], information-theoretic approaches [6, 11, 21], and so on. However, all existing methods of inferring GRNs have their strengths and weaknesses. For example, the boolean network models are very simple, but only consider the expression of a gene on or off, do not consider intermediate expression level, hence having inadequate dynamic resolution. Bayesian networks are represented as graph with the joint probability distribution of genes. This model could capture

ISSN: 1738-9968 IJHIT

the inherent noise and stochastic in gene expression data effectively, but the high computational cost hinders the application of the model with large number of genes in network. The system of differential equations is a powerful and flexible model, which can offer realistic representation of genetic networks due to their continuity. However, this model has been proved to be computationally expensive and very sensitive to imprecise data [32].

Information theoretic approaches are increasingly being used for inferring regulatory network, such as the relevance network [6], the context likelihood of relatedness (CLR) algorithm [11], ARACNE [21] and MINE [23]. These methods start by computing the mutual information (MI) for possible gene pairs, resulting in an MI matrix. The MI matrix is then manipulated to identify regulatory relationships [28]. These methods have two advantages. The one is its capability of characterizing non-linear dependency between genes [5]. The other is able to deal with thousands of genes in the presence of a limited number of samples [38]. But the methods based on MI could not distinguish indirect regulators from direct ones and discover the joint regulations of a target gene by two or more genes.

In this paper, a hybrid framework is proposed for gene regulatory network reconstruction from gene expression profiling. In our method, network topologies from linear and nonlinear ordinary differential equations are integrated. The additive tree models are proposed for identification of linear/nonlinear ODEs. The hierarchical structure of the additive tree model is evolved using genetic programming (GP) like tree structure-based evolutionary algorithm with specific instructions. Particle swarm optimization (PSO) is used to evaluate each additive tree model by optimizing the parameters of the corresponding tree model. In order to improve the accuracy and false positive rate of building gene regulatory network, we also propose a new criterion function that sparse and relevant terms are considered while inferring linear and nonlinear models.

The Dialogue for Reverse Engineering Assessments and Methods (DREAM) project could provide a set of benchmark networks that can be used to compare both advantages and disadvantages of different GRN topology inference methods [20]. The benchmark networks are extracted from actual biological networks and able to represent some most important and typical biological modules [34]. In this study, benchmark datasets from DREAM3 challenge and real biological dataset from SOS DNA repair network in *Escherichia coli* [29] are used to test the validity of our method. Results reveal that our proposed method can improve the prediction accuracy and false positive rate of GRN inference effectively and performs better than other popular methods.

The paper is organized as follows: Section 2 gives the materials and methods. Section 3 presents some experiments for construction of gene regulatory networks. Section 4 shows the discussion about our method. Some concluding remarks are presented in Section 5.

2. Method

2.1. Mathematic model of transcriptional procedure

In previous research, the algorithms based ODEs are proposed for inferring gene regulatory network, which could be applied for describing the genetic regulations among genes by using steady state and time-series expression data [37]. In this study, we present two ODE models for inferring gene regulatory network: linear and nonlinear models. For a target gene *i* (TG), which has *c* transcriptional factors (TFs), interactions between TG and TFs are described by the following ODE model containing linear and nonlinear formals:

$$\frac{dz_i}{dt} = A_i Y - \alpha_i z_i \tag{1}$$

$$\frac{dz_i}{dt} = A_i Y - \alpha_i z_i \tag{2}$$

Where $A_i = (A_i^1, A_i^2, \cdots, A_i^c)$ (representing the influence of TFs on TG), $Y = (Y_{1\times n}^1, Y_{1\times n}^2, \cdots, Y_{1\times n}^c)$ (n is the number of gene expression profiles), α_i represents the rate constant of degradation of the target gene product and f_i is the nonlinear function $\frac{dz_i}{dt} = 0$ for TF i. Assuming that equation (1) and (2) quickly reach an equilibrium state () [38], we could get the two mathematic models as followed.

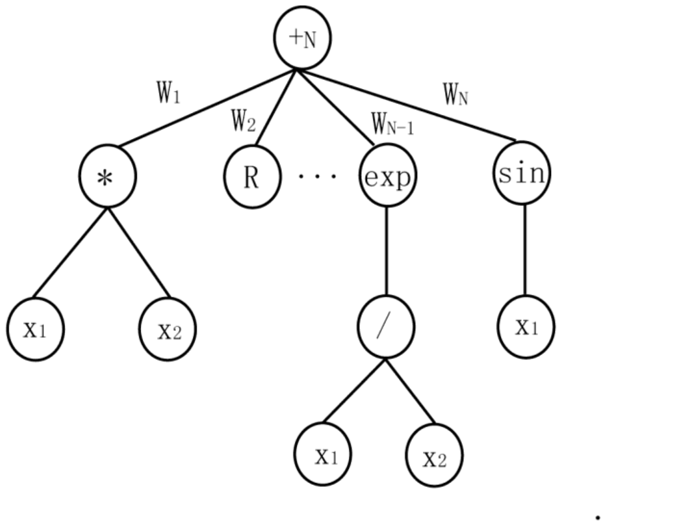
$$z_i = \frac{A_i Y}{\alpha_i} = B_i Y \tag{3}$$

$$z_i = \frac{A_i f_i(Y)}{\alpha_i} = B_i f_i(Y) \tag{4}$$

Where $B_i = (B_i^1, B_i^2, \cdots, B_i^c)$. Thus the problem of inferring gene regulatory network could be decomposed to N dependent subproblems (N is the number of gene).

2.2. Representation of additive tree model

The additive models resulted from our previously published work, are proposed for the system identification especially the reconstruction of polynomials and the identification of linear/nonlinear systems [10]. Thus we encode the right-hand side of linear and nonlinear mathematic models into a additive tree individual (Please refer to Figure. 1) in this subsection.



$$z = W_1 y_1 y_2 + W_2 R + W_{N-1} \exp(y_1/y_2) + W_N \sin(y_1)$$

Figure 1. Example of a mathematic formal in the form of the additive tree model

Two instruction/operator sets I_0 and I_1 are used to generate the additive tree in this approach. For linear method, we set two instruction/operator sets I_0 and I_1 as followed.

$$\begin{aligned} I_0 &= \{+2, +3, ..., +N\} \\ I_1 &= F \bigcup T = \{+, -, x, R\} \end{aligned} \tag{5}$$

Where F = {+, -} and T = {x, R} are function and terminal set. +, x and R denote the addition, subtraction, gene expression of TFs, and random constant number. I_0 is the instruction set and the root node, and the instructions of other nodes are selected from the instruction set I_1.

Many biologically significant nonlinear functions are added into the nonlinear model for gene regulatory network identification, *e.g.*, sigmoid transfer function [33], hill function [22] and hyperbolic tangent function [9]. Thus we add these functions into operator sets I_1 as followed.

$$I_1 = F \bigcup T = \{*, /, \sin, \cos, \frac{1}{1+e^{-x}}, \frac{x^h}{x^h+\alpha}, \frac{1}{x^h+\alpha}, \tanh, x, R\} \tag{6}$$

2.3. Structure optimization of models

Finding an optimal or near-optimal additive tree model is formulated as an evolutionary search process. In this paper, we use the additive tree operators as following:

⑴ Mutation. We choose three mutation operators to generate offsprings from the parents. These mutation operators are as following:

1) Change one terminal node: randomly select one terminal node in the tree and replace it with another terminal node, which is generated randomly.

2) Grow: select a random leaf in hidden layer of the tree and replace it with a newly generated subtree.

3) Prone: randomly select a function node in a tree and replace it with a terminal node selected in the set T.

The additive tree operators are applied to each of the parents to generate an offspring using the following steps. (a) A Poisson random number N, with mean λ is generated. (b) N random mutation operators are uniformly selected with replacement from above four mutation operator set. (c) These N mutation operator are applied in sequence one after the other to the parent to get offspring.

⑵ Crossover. First two parents are selected according to the predefined crossover probability Pc and select one nonterminal node in the hidden layer for each additive tree randomly, and then swap the selected subtree.

⑶ Selection. EP-style tournament selection [7] is applied to select the parents for the next generation. Pairwise comparison is conducted for the union of μ parents and μ offsprings. For each individual, q opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the fitness of individual fitness is no smaller than the one of opponent, it receives a selection. Select μ individuals out of parents and offsprings, which have most wins to form the next generation. This is repeated in each generation until a predefined number of generations or the best structure is found.

2.4. MI

Mathematically, the MI between two discrete random variables *X* (one gene) and *Y* (another gene), taking on values in χ and γ is defined as [2].

$$I(X,Y)=\sum_{x\in\chi}\sum_{y\in\gamma}p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \tag{7}$$

Where $p(x)$ is the marginal probability of each discrete variable *X* in χ, $p(x, y)$ is the joint probability of *x* in χ and *y* in γ .

Here, the marginal and joint probability can be estimated by the Gaussian kernel probability density estimator [24].

$$p(x_i)=\frac{1}{N}\sum_{j=1}^{N}\frac{1}{(2\pi)^{n/2}\sigma_x{}^{n/2}}\exp(-\frac{1}{2}(X_j-X_i)^T C^{-1}(X_j-X_i)) \tag{8}$$

Where *C* is the covariance of variable *X*, σ_x is the standard deviation of matrix *C*, *N* is the number of gene expression and *n* is the number of gene.

With equation (8), the equation (7) can be given as followed.

$$I(X,Y)=\frac{1}{2}\log(\frac{\sigma_X^2\sigma_Y^2}{|C(X,Y)|}) \tag{9}$$

Where $|C(X,Y)|$ is the determinant of the covariance matrix. High MI value indicates that there may be a close relationship between the variables (genes), while low MI value implies their independence. When MI value between genes X and Y is zero, two genes are independent.

2.5. Evaluation of model using Particle Swarm Optimization

The quality of a list of TFs for an individual target gene is evaluated by estimating the parameters of the corresponding additive tree model using particle swarm optimization (PSO) [15]. According to the Figure 1, we check all the parameters contained in each model, and count their number n_i (i = 1, 2, ..., *M*, *M* is the population size of additive tree model).

According to n_i, the particles are randomly generated initially. Each particle x_i represents a potential solution. A swarm of particles moves through space, with the moving velocity of each particle represented by a velocity vector v_i. At each step, each particle is evaluated and keep track of its own best position, which is associated with the best fitness it has achieved so far in a vector $Pbest_i$. The best position among all the particles is kept as Gbest. A new velocity for particle *i* is updated by

$$v_i(t+1)=w*v_i(t)+c_1r_1(Pbest_i-x_i(t))+c_2r_2(Gbest(t)-x_i(t)) \tag{10}$$

where *w* is the inertia weight and impacts on the convergence rate of PSO, which is computed adaptively as *w* = (*max_generation* − *t*)/(2**max_generation*) +0.4 (*max_generation* is maximum number of iterations, and *t* is current iteration), c_1 and c_2 are positive constant and

r_1 and r_2 are uniformly distributed random number in [0,1]. Based on the updated velocities, each particle changes its position according to the following equation:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{11}$$

To evaluate the performance of proposed additive tree model and its corresponding parameters, we proposed new criterion functions as followed. In general the criterion function is to minimize the prediction error for the gene expression profiles, according to

$$F_i = \frac{1}{T}\sum_{k=1}^{T}(z_{it} - z_{it}^{'})^2 \tag{12}$$

Where z_{it} and $z^{'}_{it}$ are the actual and predicted expression levels of gene i at t- th gene expression point.

The candidate regulators for a given target gene is much larger than the number of expression points, and gene regulatory network is sparse network, *i.e.*, only a tiny fraction of the candidate regulators are expected to be true regulators for target gene, so we propose to add a sparse term to the criterion function in order to perform feature selection while inferring the linear and nonlinear model [30]. The criterion function is described as followed.

$$F_i = \frac{1}{T}\sum_{k=1}^{T}(z_{it} - z_{it}^{'})^2 + \alpha\|B_i\| \tag{13}$$

Where $\alpha\|B_i\|$ is the sparse term, $\|B_i\|$ is L_1 regularizer of vector B_i from equation (3) and (4), and α is a parameter that trades off the error and sparse term in the criterion function.

Considering that MI value indicates that there may be a close relationship between the genes in gene regulatory network, we propose to add a relevant term to the criterion function in order to select maximum relevance regulators of target gene [12].

$$F_i = \frac{\frac{1}{T}\sum_{k=1}^{T}(z_{it} - z_{it}^{'})^2}{\left[1 + \frac{1}{m}\sum_{j \notin \Omega_i} I(X_i, X_j)\right]} + \alpha\|B_i\| \tag{14}$$

Where Ω_i denotes the already-selected regulator set containing m regulators for target gene i, $I(X_{i,}X_j)$ is MI value between gene i and gene j.

2.6. Procedure of inferring gene regulatory network

Figure 2 describes the flowchart of our proposed method. The details are addressed as following.

Step 1 Network construction.

1) Initialization. Input the gene expression data Z and initiate adjacency matric $A_{ij} = 0$ and B_{ij} for linear and nonlinear models (i, $j \in N$, N is the number of gene). Estimate mutual information I_{ij} for all gene pairs.

2) For each target gene i, we use the additive tree model to identify linear/nonlinear models. The details are addressed as following.

a) Create the initial population randomly (structures and their corresponding parameters)as described in subsection 2.2;

b) Structure optimization is achieved by the additive tree variation operators as described in subsection 2.3;

c) Evaluate the population using PSO and our proposed criterion functions;

d) If maximum number of generations is reached or a satisfactory solution is found, then stop; otherwise go to step (b).

3) Sign the regulators for target gene *i*. If the gained models contain gene *k*, set A_{ik}=1 or B_{ik}=1. In other words gene *k* regulates gene *i*.

Step 2 Network integration. Considering gene expression data, we could gain the linear network structure *A* and nonlinear network structure *B*. The integration of two network $C_{ij}=A_{ij} \| B_{ij}$, where ‖ is logical or (0 ‖ 0=0, 0 ‖ 1=1 and 1 ‖ 1=1).

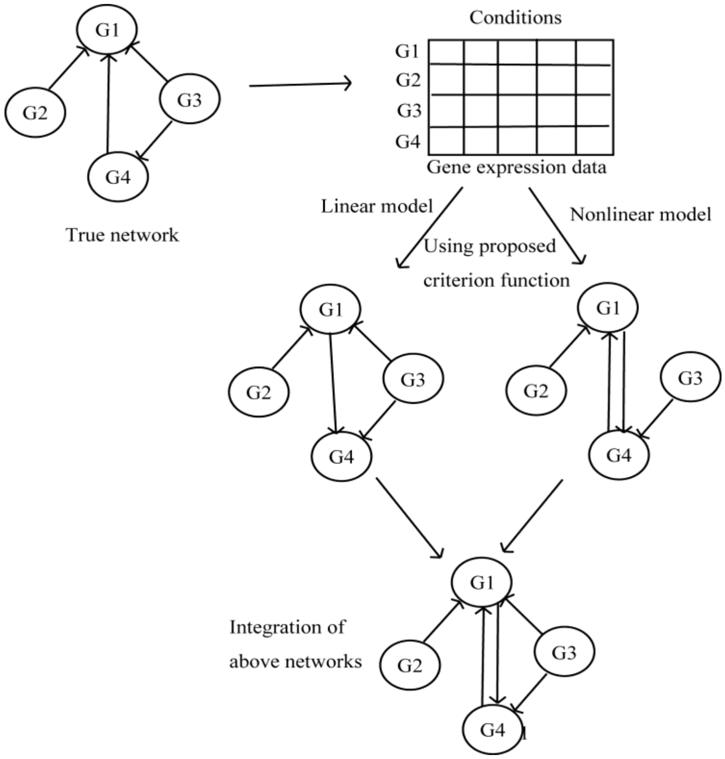


Figure 2. Overview of our proposed method. In the true network, the genes are labelled with G. Using the additive tree models to identify linear/nonlinear model, we could gain linear/nonlinear network structures. Compared with true network, linear network has one false positive edge (*G*1→ *G*4) and *G*4 → *G*1 could not be identified. And nonlinear network also has one false positive edge (*G*1 → *G*4) and *G*3 → *G*1 could not be identified. The overall network integrate linear network with nonlinear one.

3. Experimental Results and Analysis

To test the effectiveness of the proposed method, our method is applied to two synthetic datasets from the DREAM3 (Dialogue for Reverse Engineering Assessment and Methods) challenge [20] and one real gene expression dataset from SOS DNA repair network in Escherichia coli [29]. While inferring the gene regulatory network, two criterion functions are used (equation (12) and equation (14)), labelled with Fitness1 and Fitness2.

Five criterions (sensitivity or true positive rate (TPR), false positive rate (FPR), positive predictive (PPV), accuracy (ACC) and F-score) are used to test the performance of the method. Firstly, we define four variables, *i.e.*, TP, FP, TN and FN are the number of true positives, false positives, true negatives and false negatives, respectively. Five criterions are defined as followed.

$$TPR=TP/(TP+FN),$$

$$FPR=FP/(FP+TN),$$

$$PPV=TP/(TP+FP),$$

$$ACC=(TP+TN)/(TP+FP+TN+FN),$$

$$F\text{-}score=2PPV*TPR/(PPV+TPR)$$

3.1. Simulated data

To illustrate the effectiveness of our method for inferring GRN, tests are performed on benchmark networks with expression datasets from DREAM challenge [20]. In this experiment, the DREAM3 datasets about *Yeast* knock-out genes with size 50 and 100 are used. To evaluate the performance of our method, we compare it with several classic methods including LASSO [13], random forest GENIE3 with parameters'*sqrt*' and '*a*ll' [14], respectively, noise and redundancy reduction NARROMI [38]. The parameters in these methods are set by default.

Firstly, we test our method on the *Yeast* gene expression with network size 50, sample number 50. Table 1 shows the results obtained by different methods with respect to TPR, FPR, PPV, ACC and F-score. From the results, we can see that our method is superior to the other methods except the method NARROMI for ACC.

Table 1. Comparison of different methods on networks with sizes 50 in DREAM3

	TPR	FPR	PPV	ACC	F-score
LASSO	0.351	0.129	0.081	0.855	0.131625
GENIE3-sqrt	0.481	0.078	0.167	0.908	0.247923
GENIE3-all	0.442	0.073	0.164	0.912	0.239234
NARROMI	0.532	0.062	0.217	0.925	0.308262
Fitness1	**0.5584**	0.1024	0.15035	0.86939	0.236915
Fitness2	**0.5584**	**0.05984**	**0.23243**	**0.91061**	**0.328242**

Secondly our method is applied to the Y east gene expression with network size 100, sample number 100. Table 2 shows the results obtained by different methods with respect to five indexes. From the results, we can see that our proposed our method performs better than the other methods except the method GENIE3-all for TPR.

Table 2. Comparison of different methods on networks with sizes 100 in DREAM3

	TPR	FPR	PPV	ACC	F-score
LASSO	0.3494	0.10694	0.052775	0.87808	0.0917
GENIE3-sqrt	0.44578	0.064311	0.10571	0.92	0.1709
GENIE3-all	**0.55422**	0.062461	0.13143	0.92182	0.22247
NARROMI	0.5	0.04732	0.183	0.9351	0.267936
Fitness1	0.45783	0.084652	0.08444	0.9	0.14259
Fitness2	0.53614	**0.03534**	**0.20554**	**0.94848**	**0.297158**

In addition, to assess the effectiveness of our proposed criterion function, the ROC curves obtained and five criterions by Fitness1 and Fitness2 on 50-gene network and 100-gene network from DREAM3 are shown in Figure 3 and Table 1 and 2, respectively. The results shows that our proposed criterion function Fitness2 performs better than general function Fitness1.

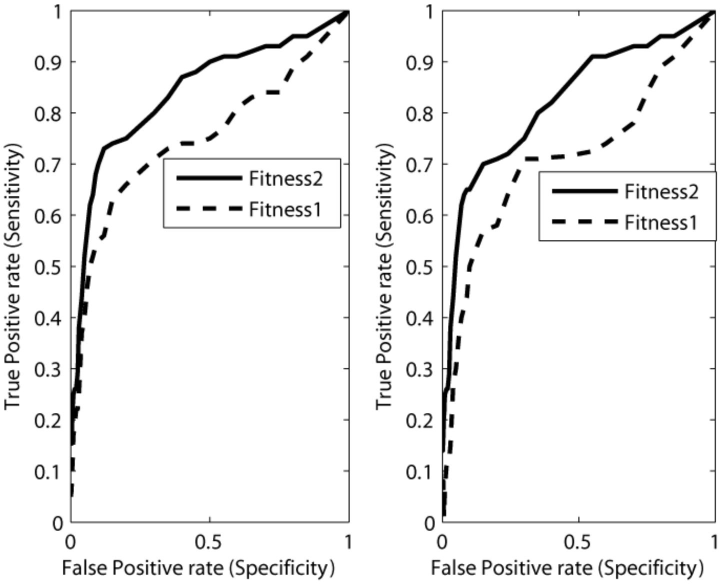


Figure 3. The comparison of ROC curves of criterion function *Fitness*1 and *Fitness*2 on 50-gene network (left part) and 100-gene network (right part) from DREAM3

3.2. Real gene expression data

The real biological datasets result from SOS DNA Repair network of *Ecoli* bacterium [29], which contain four experiments under various light intensities (5Jm^{-2}, 5Jm^{-2}, 20Jm^{-2}, 20Jm^{-2}). Each experiment (http://www. weizmann.ac.il/mcb/UriAlon/Papers/ SOSData/) consists of 50 time points evenly spaced by 6 min, refers to eight genes: uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA and polB. Figure 4 (left) displays the known regulatory interactions among 8 genes. Firstly, the gene expression profiles for each gene are normalized in the interval [0,1] using

$$x_i^{'}(t)=\frac{x_i(t)-\min_i}{\max_i-\min_i} \tag{15}$$

Where $x_i(t)$ is the actual expression level of gene i at time point t, min_i and max_i are the minimum and maximum expression levels of gene i, respectively. Figure 4 (right) shows our predicted topology of SOS network, consisting of 16 edges, eight of which are true positives. But the self-regulating edge *lexA* → *lexA* could not be identified. Table 3 shows the detail comparison of results from different method which are commonly used to infer the gene regulatory network. The results shows that our proposed method performs better than other methods.

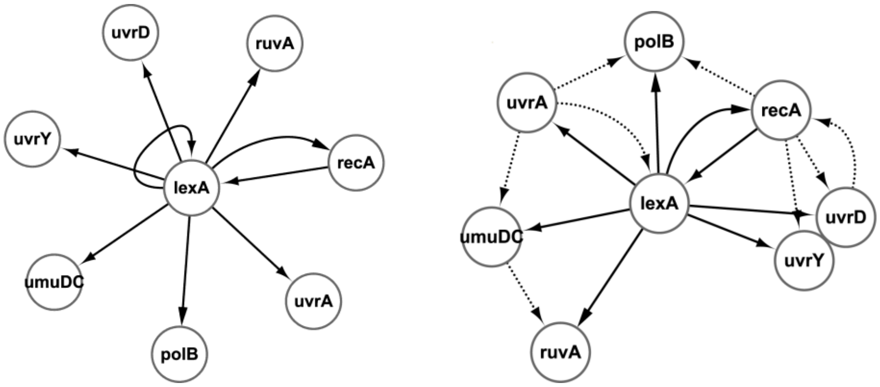


Figure 4. The left part is the true SOS network with eight genes and 9 interactions. The right part is our predicted SOS network. Correctly identified edges are drawn with solid lines and falsely identified edges with dashed lines

Table 3. Comparison of different methods for SOS dataset

Interaction	S-system[26]	DBN[27]	RNN[35]	ODEs[16]	Fitness2
lexA → uvrD	yes	yes	no	yes	yes
recA → lexA	yes	yes	yes	yes	yes
lexA → lexA	yes	yes	no	yes	no
lexA → lexA	yes	yes	yes	no	yes
lexA → recA	no	no	yes	yes	yes
lexA → uvrA	no	no	no	no	yes
lexA → uvrY	yes	no	yes	yes	yes
lexA → ruvA	no	no	no	no	yes
lexA → polB	no	no	yes	yes	yes
TP	5	4	5	6	8
FP	10	5	2	15	8
TPR	0.5556	0.4444	0.55556	0.6667	0.88889
FPR	0.20833	0.10417	0.041667	0.3125	0.16667
PPV	0.3333	0.44444	0.71429	0.28571	0.5
ACC	0.6667	0.75439	0.80702	0.57895	0.70175
F-score	0.41667	0.44444	0.625	0.4	0.64

4. Discussion

In this paper, we propose a novel method to infer gene regulatory network from gene expression data. From results on the DREAM3 benchmark datasets and real biological datasets result from SOS DNA Repair network, our method is effective and superior to other methods significantly. The following factors may lead to the good performance of our method.

First, it integrates network topologies from linear and nonlinear models, *i.e.*, the overall network consisting of linear and nonlinear relations between gene pairs. It overcomes the defects resulting from linear methods losing nonlinear relations and nonlinear methods losing linear relations.

Second, the criterion function we proposed contains sparse and relevant terms. The sparse term satisfies the condition that each target gene has a tiny fraction of the candidate regulators as true regulators, and could weaken the influence that candidate regulators for a given target gene is much larger than the number of expression points. The relevant term utilizes MI from information theory to evaluate the correlation between gene pairs in order to detect the statistical dependence and select maximum relevance regulators of a target gene.

In addition, our method can distinguish the direct interactions from indirect ones, which are important for causality analysis. However, there is some limitation for our method. As same as that of NARROMI, our method cannot infer self-regulating edge, which is also a general limitation of many other methods [11, 21].

5. Conclusion

In this paper, we propose a novel method to improve the accuracy of gene regulatory network inference from gene expression data. In this method, network topologies from linear and nonlinear models are integrated. Moveover the sparse and relevant terms are added to the criterion function while inferring linear and nonlinear models. The method is validated on the benchmark gene regulatory from DREAM3 challenges and SOS DNA repair network of *Ecoli* bacterium. The results are superior to the previous methods.

Acknowledgements

This research was partially supported by the Natural Science Foundation of China (61070130), the Key Subject Research Foundation of Shandong Province, National Natural Science Foundation of Shandong Province of China under Grant No.ZR2010FM040, Special Funding Project for Independent Innovation Achievements transform of Shandong Province under Grant No.2012CX30202 and Foundation of Shandong University for Independent Innovation under Grant No.2012DZ038 and the Shandong Provincial Key Laboratory of Network Based Intelligent Computing.

References

[1] T. Akutsu, S. Miyano and S. Kuhara, "Identification of Genetic Networks from a Small Number of Gene Expression Patterns under the Boolean Network Model", Pac Symp Biocomput, **(1996)**, pp. 17-28.

[2] G. Altay and F. Emmert-Streib, "Inferring the conservative causal core of gene regulatory networks", BMC Syst Biol, vol. 4, **(2010)**, pp. 132.

[3] S. I. Ao and V. Palade, "Ensemble of Elman neural networks and support vector machines for reverse engineering of gene regulatory networks", Applied Soft Computing, vol. 11, no. 2, **(2011)**, pp. 1718-1726.

[4] S. Bornholdt, "Boolean Network Models of Cellular Regulation: Prospects and Limitations", J. R. Soc. Interf, vol. 5, **(2008)**, pp. 85-94.

[5] H. Brunel, J. J. Gallardo-Chacn, A. Buil, M. Vallverd, J. M. Soria, P. Caminal and A. Perera, "MISS: a non-linear methodology based on mutual information for genetic association studies in both population and sib-pairs analysis", Bioinformatics, vol. 26, **(2010)**, pp. 1811-1818.

[6] A. J. Butte and L. S. Kohane, "Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements", Pacific Symposium on Biocomputing, vol. 4, **(2000)**, pp. 418-429.

[7] K. Chellapilla, "Evolving computer programs without subtree crossover", IEEE Transactions on Evolutionary Computation, vol. 1, **(1997)**, pp. 209-216.

[8] T. Chen, H. L. He and G. M. Church, "Modeling Gene Expression with Differential Equations", In: Proc. of Pacific Symposium on Biocomputing, **(1999)**, pp. 29-40.

[9] C. M. Chen, C. Lee, C. L. Chuang, C. C. Wang and G. S. Shieh, "Inferring genetic interactions via a nonlinear model and an optimization algorithm", BMC Syst Biol, vol. 4, **(2010)**.

[10] Y. H. Chen, J. Yang, Y. Zhang and J.W. Dong, "Evolving Additive tree models for System Identification", International Journal of Computational Cognition, vol. 3, no. 2, **(2005)**, pp. 19-26.

[11] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierbowski, G. Cottarel, S. Kasif, J. J. Collins and T. S. Gardner, "Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles", Plos Biology, vol. 5, no. 1, **(2007)**.

[12] Y. F. Gao, B. Q. Li, Y. D. Cai, K. Y. Feng, Z. D. Li and Y. Jiang, "Prediction of active sites of enzymes by maximum relevance minimum redundancy (mRMR) feature selection", Mol Biosyst, vol. 9, no. 1, **(2013)**, pp. 61-9.

[13] G. Geeven, R. E. van Kesteren, A. B. Smit and M. C. de Gunst, "Identification of context-specific gene regulatory networks with GEMULA-gene expression modeling using Lasso", Bioinformatics, vol. 28, **(2012)**, pp. 214-221.

[14] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel and P. Geurts, "Inferring regulatory networks from expression data using tree-based methods", PLoS One, vol. 5, **(2010)**.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization", in: Proceedings of the 1995 IEEE International Conference on Neural Networks, vol. 4, **(1995)**, pp. 1942-1948.

[16] S. Kimura, K. Sonoda, S. Yamane, H. Maeda, K. Matsumura and M. Hatakeyama, "Function Approximation Approach to the Inference of Reduced NGnet Models of Genetic Networks", BMC Bioinformatics, vol. 9, no. 1, **(2008)**.

[17] S. Knott, S. Mostafavi and P. Mousavi, "A neural network based modeling and validation approach for identifying gene regulatory networks", Neurocomputing, vol. 73, **(2010)**, pp. 2419-2429.

[18] S. Liang, S. Fuhrman and R. Somogyi, "Reveal: a general reverse engineering algorithm for inference of genetic network architectures", Pacific Symposium on Biocomputing, **(1998)**, pp. 18-29.

[19] I. A. Maraziotis, A. Dragomir and A. Bezerianosh, "Gene networks reconstruction and time-series prediction from microarray data using recurrent neural fuzzy networks", IET Syst. Biol, vol. 1, no. 1, **(2007)**, pp. 41-50.

[20] D. Marbach, R. J. Prill, T. Schaffter, C. Mattiussi, D. Floreano and G. Stolovitzky, "Revealing strengths and weaknesses of methods for gene network inference", Proc Natl Acad Sci USA, vol. 107, no. 14, **(2010)**, pp. 6286-91.

[21] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, D. R. Favera and A. Califano, "ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context", BMC Bioinformatics, vol. 7, no. Suppl 1, **(2006)**.

[22] L. Marucci, S. Santini, M. di Bernardo and D. di Bernardo, "Derivation, identification and validation of a computational model of a novel synthetic regulatory network in yeast", J Math Biol, vol. 62, no. 5, **(2011)**, pp. 685-706.

[23] P. E. Meyer, F. Lafitte and G. Bontempi, "minet: A R/Bioconductor Package for Inferring Large Transcriptional Networks Using Mutual Information", BMC Bioinformatics, vol. 9, **(2008)**.

[24] P. E. Meyer, K. Kontos and G. Bontempi, "Information-theoretic inference of large transcriptional regulatory networks", EUROSIP journal on bioinformatics and systems biology, **(2007)**.

[25] K. Murphy and S. Mian, "Modeling Gene Expression Data using Dynamic Bayesian Network", Computer Science Division, University of California Berkeley, **(1999)**.

[26] N. Noman and H. Iba, "Reverse Engineering Genetic Networks Using Evolutionary Computation", Genome Informatics, vol. 16, no. 2, **(2005)**, pp. 205-214.

[27] B. E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet and D. F. Buc, “Gene Network Inference Using Dynamic Bayesian Networks”, Bioinformatics, vol. 19, no. Suppl.2, **(2003)**, pp. 138-148.

[28] P. Qiu, A. J. Gentles and S. K. Plevritis, “Fast calculation of pairwise mutual information for gene regulatory network reconstruction”, Comput Methods Programs Biomed, vol. 94, no. 2, **(2009)**, pp. 177-80.

[29] M. Ronen, R. Rosenberg, B. I. Shraiman and U. Alon, “Assigning numbers to the arrows: Parameterizing a gene regulation network by using accurate expression kinetics”, Proc. Natl. Acad. Sci. USA, vol. 99, **(2002)**, pp. 10555-10560.

[30] Y. Shi, Y. Guo and D. Schuurmans, “Kernel-based gene regulatory network in- ference”, International Conference on Computational Systems Bioinformatics, **(2010)**, pp. 156-165.

[31] A. Sirbu, H. J. Ruskin and M. Crane, “Comparison of evolutionary algorithms in gene regulatory network model inference”, BMC Bioinformatics, vol. 11, **(2010)**.

[32] J. Tegner, M. Yeung, J. Hasty and J. Collins, “Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling”, Proc. Nat. Acad. Sci. USA, vol. 100, no. 10, **(2003)**, pp. 5944-5949.

[33] T. T. Vu and J. Vohradsky, “Nonlinear differential equation model for quantification of transcriptional regulation applied to microarray data of Saccharomyces cerevisiae”, Nucleic Acids Res, vol. 35, no. 1, **(2007)**, pp. 279-287.

[34] Y. Wang and T. Zhou, “A relative variation-based method to unraveling gene regulatory networks”, PLoS One, vol. 7, no. 2, **(2012)**.

[35] R. Xu, D. C.Wunsch and R. L. Frank, “Inference of Genetic Regulatory Networks with Recurrent Neural Network Models Using Particle Swarm Optimization”, IEEE/ACM Trans. Computational Biology and Bioinformatics, vol. 4, no. 4, **(2007)**, pp. 681-692.

[36] X. Yang, J. E. Dent and C. Nardini, “An S-System Parameter Estimation Method (SPEM) for biological networks”, J Comput Biol, vol. 19, no. 2, **(2012)**, pp. 175-87.

[37] K. Y. Yip, R. P. Alexander, K. K. Yan and M. Gerstein, “Improved reconstruction of in silico gene regulatory networks by integrating knockout and perturbation data”, PLoS One, vol. 5, no. 1, **(2010)**, pp. e8121.

[38] X. Zhang, K. Liu, Z. P. Liu, B. Duval, J. M. Richer, X. M. Zhao, J. K. Hao and L. Chen, “NARROMI: a noise and redundancy reduction technique improves accuracy of gene regulatory network inference”, Bioinformatics, vol. 29, no. 1, **(2013)**, pp. 106-113.

Authors

Bin Yang

Bin Yang is currently pursuing his Ph.D. in School of Information science and Engineering from Shandong University, Jinan, P.R. China. He received his B.Sc. and Master degree in School of Information Science and Engieering from University of Jinan. His research interests include Hybrid Computational Intelligence and its applications in time-series prediction, system identification and Gene regulatory network**.**

Mingyan Jiang

Mingyan Jiang is a professor in School of Information Science and Engineering of Shandong University, Jinan, P.R. China. He received his M. S. degree in 1992 in electrical engineering and got his Ph. D degree in Communication Engineering from Shandong University in 2005 in China. His research interests include communication signal processing, Soft computing methods, Image processing, communication network, remote control and navigation tracking and positioning technology.

Yuehui Chen

Yuehui Chen was born in 1964 in Shandong Province of China. He received his B.Sc. degree in mathematics/automatics from the Shandong University of China in 1985, and Master and Ph.D. degree in electrical engineering from the Kumamoto University of Japan in 1999 and 2001. During 2001-2003, he had worked as the Senior Researcher of the Memory-Tech Corporation at Tokyo. Since 2003 he has been a member at the Faculty of Electrical Engineering in Jinan University, where he is currently head of the Laboratory of Computational Intelligence. His research interests include Evolutionary Computation, Neural Networks,Fuzzy Logic Systems, Hybrid Computational Intelligence and their applications in time-series prediction, system identification, intelligent control, intrusion detection systems, web intelligence and bioinformatics. He is the author and co-author of more than 70 technique papers.Professor Yuehui Chen is a member of IEEE, the IEEE Systems, Man and Cybernetics Society and the Computational Intelligence Society, a member of Young Researchers Committee of the World Federation on Soft Computing, and a member of CCF Young Computer Science and Engineering Forum of China. More information at: http://cilab.ujn.edu.cn.