International Journal of Hybrid Information Technology
Vol.8, No.7 (2015), pp.225-238
http://dx.doi.org/10.14257/ijhit.2015.8.7.21

Using Additive Expression Programming for Gene Regulatory Network Inference

Bin Yang

School of Information Science and Engineering, Zaozhuang University, Zaozhuang, P.R. China 277160
E-mail: batsi@126.com

Abstract

Gene regulatory networks depict the interactions among genes in the cell and construction of networks is important in uncovering the underlying biological process of living organisms. In this paper, a non-linear differential equation model is used for gene regulatory network reconstruction and time-series prediction. A new model, called additive expression tree (AET) model is proposed to encode ordinary differential equations (ODEs). A new structure-based evolutionary algorithm and artificial bee colony (ABC) are used to optimize the architecture and parameters of the additive expression tree model, respectively. A synthetic data and two real time-series expression datasets are used to test the validity of our proposed model and hybrid approach. Experimental results demonstrate that our model could improve accuracy of microarray time-series data effectively.

Keywords: *gene regulatory network, additive expression tree, artificial bee colony, hybrid approach.*

1. Introduction

Interactions among genes, transcription factor (regulator), mRNA and gene product (proteins) almost control all biological activity, which constitute a gene regulatory network. Reconstruction of gene regulatory network (GRN) have been playing an important role in understanding the complexity, function and pathways of biological system, and formation of new drugs about disease [1-2]. With the rapid development of microarray technology and second-generation sequencing technology, a large number of expression data has been produced [3]. Due to the high sequencing costs, most of gene expression data only contain a few time points. Thus prediction of expression data could speed up biotechnological projects and reduce costs of lab experiments. Since then, combined with the available expression data, how to predict and infer gene regulatory network has been becoming a major area of interest in the field of systems biology [4-5].

In the past several years, many models have been proposed to tackle time series expression data and infer gene regulatory network, such as Boolean network [6-7], dynamic Bayesian network [8], neural network [9], Petri net [10-11], information theoretic approaches [12], the system of differential equation. The system of differential equations is powerful and flexible model to describe complex relations among components. We have used the ODE model to predict the small-scale traffic measurements data and stock index, and experimental results revealed that the ODE method was feasible and efficient for forecasting time-series data [13]. Many methods were proposed for inferring a system of differential equations for the gene regulatory network during the last few years. Tominaga D. used the Genetic Algorithms (GA) to optimize the parameters of the fixed form of system of differential equations [14]. Palafox et al. [15] implemented a variation of particle swarm optimization (PSO), called

ISSN: 1738-9968 IJHIT

dissipative PSO (DPSO) to optimize the parameters of the popular non-linear differential equation model named S-System in order to infer small gene regulatory networks. Xu et al. used PSO to search the best weights of recurrent neural network (RNN) model for gene network inference from gene expression data [16]. Noman et al. used the evolutionary algorithm called differential evolution to find the best RNN model for inferring the underlying network structure as well as the regulatory parameters [17]. However most methods focused on the fixed structure of equation and only need design optimization algorithm to identify the parameters of the ODEs.

We have proposed a new representation scheme of the additive tree models for the system identification especially identification of linear/nonlinear ODE system [13, 18]. Compared with genetic programming (GP) and gene expression programming (GEP), this model was powerful than the GEP and GP models, both in the aspects of accuracy and runtime. But like GP individuals, additive tree models are also represented and manipulated as nonlinear tree entities, which leads to some problems such as the inefficiencies of handling tree structure and the difficulties of program implementation.

In this paper, as linear variant of additive tree model, additive expression tree (AET) model is first proposed to encode non-linear ODEs. We propose a hybrid evolutionary method, in which a new structure-based evolutionary algorithm is used to optimize the architecture of system and select input variables, and corresponding parameters are evolved by artificial bee colony. A synthetic data obtained by biochemical pathway and two real time-series expression datasets from Human cell time-series dataset and E. coil dataset are used to test the validity of our proposed model and hybrid approach. Experimental results demonstrate that our model could improve accuracy of microarray time-series data effectively.

2. Representation of Additive Expression Tree Model

We use a structure-based evolutionary algorithm to evolve the architecture of the additive expression models for the nonlinear ODE system. For this purpose, we encode the nonlinear expression into an additive expression tree model as illustrated in Figure. 1. Figure. 1(a) is nonlinear expression of symbolic tree structure, which need be created randomly, and Figure. 1(b) is the corresponding expression tree structure.

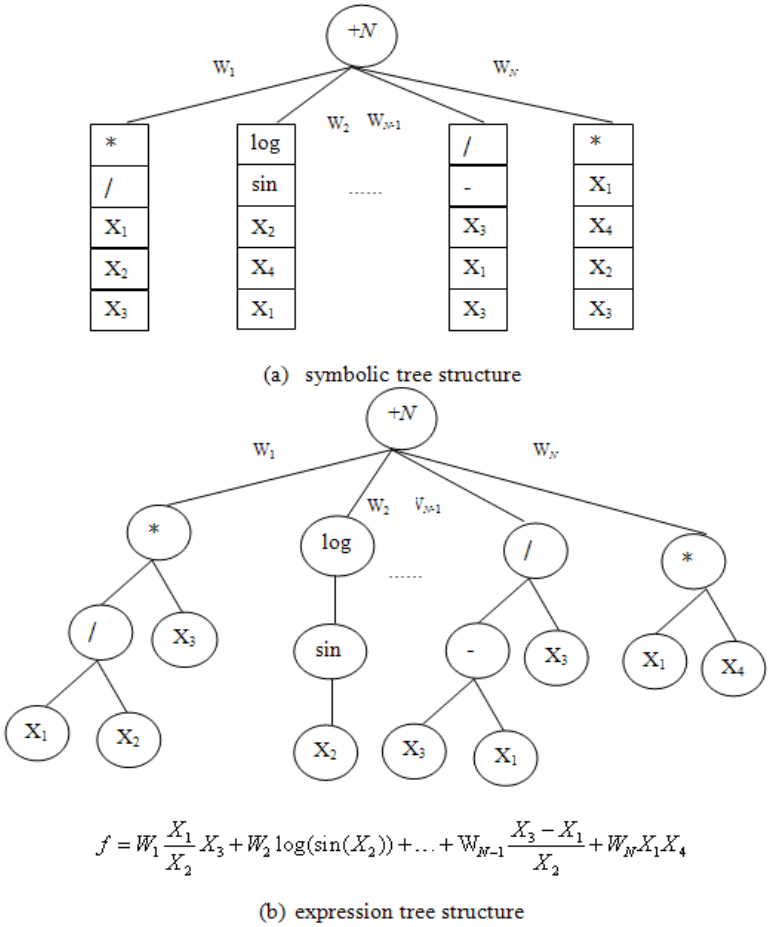

Figure 1. Example of a System in the Form of Additive Expression Tree Model

Two instruction/operator sets I_0 and I_1 are used to generate the additive expression tree model.

$$\begin{cases} I_0 = \{+_2, +_3, \ldots, +_N\} \\ I_1 = F \bigcup T = \{*, /, \sin, \cos, \exp, \text{rlog}, \text{x}, \text{R}\} \end{cases} \quad (1)$$

N is an integer number (the maximum number of linear/nonlinear terms). Each term is encoded as GEP gene. I_0 is the instruction set and the root node, which returns the weighted sum of a number of linear/nonlinear terms according to the GEP gene expressions. I_1 is the operator set, where $F = \{*, /, \sin, \cos, \exp, r\log\}$ and $T = \{x, R\}$ are function and terminal sets. *, /, *sin*, *cos*, *exp*, *rlog*, *x*, and *R* denote the multiplication, protected division ($\forall x, y \in R$: when $y = 0$, $x/0 = 1$), sine, cosine, exponent, protected logarithm ($\forall x \in R$, $x \neq 0$: $rlog(x) = log(abs(x))$ and $rlog(0) = 0$), system inputs, and random constant number, taking 2, 2, 1, 1, 1, 1, 0 and 0 arguments respectively [18]. According to the specific problems, we could add some complex functions into operator set I_1, for example

$$I_1 = \{*, /, \sin, \cos, \log, \tan, \frac{x^2}{1+x^2}, \frac{1}{1+x^2}, \frac{1}{1+e^{-x}}, *3, *4, x, R\} \quad (2)$$

Where *3 represents that three variables are multiplied.

A GEP gene is a string of function and terminal symbols, which is composed of a head and a tail [19-20]. The head part contains both function and terminal symbols, whereas the tail part contains terminal symbols only. The head could be created through selecting symbols randomly from the set I_1. The symbols of tail are selected from set F only. For each problem, user must determine the head length (h). The tail length (t) is computed as:

$$t = (n-1) \times h + 1 \tag{3}$$

Where n is the maximum number of arguments of functions. According to set F, n is set as 2.

3. The Proposed Hybrid Method

In this section, it is explained the proposed hybrid method. The method description can be made in several sub-chapters [21].

3.1. Structure Optimization Methods

To search an optimal or near-optimal additive expression tree model is formulated as an evolutionary finding process. We use the structure operators as following:

(1) Mutation. We use three mutation operators to generate offsprings from the parents, which are described as following:
   a) One-point mutation. Select one point in the tree randomly, and replace it with another symbol, which is selected from set I_1. Notice that in the head any symbol could be changed, but in the tail the terminal symbols are allowed to be changed only.
   b) One-gene mutation. Randomly select one GEP gene in the tree, and replace it with another newly generated gene.
   c) Change all terminal symbols. Select every terminal symbol in the additive expression tree model, and replace it with another terminal symbol.

(2) Crossover. First two parents are selected according to the predefined crossover probability P_c and select one GEP gene for each additive expression tree randomly, and then swap the selected gene.

(3) Selection. EP-style tournament selection [22] 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.

3.2. Fitness Definition

Mean square error (MSE) and root mean square error (RMSE) are used as fitness functions to evaluate the performance of candidate model.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\mathrm{x}_1^i - \mathrm{x}_2^i)^2$$
$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathrm{x}_1^i - \mathrm{x}_2^i)^2} \tag{4}$$

Where N is the number of samples, x_1^i and x_2^i are the actual and model output of i-th sample.

3.3. Parameter Optimization of Models

In the parameter learning stage, there are many learning methods, such as PSO, GA, EP and SA. In this paper, artificial bee colony algorithm is selected because its fast convergence, more accurate solution and stability [23-24].

According to 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 expression tree models). According to n_i, the parameter vector could be created randomly. ABC algorithm can be summarized as follows.

1. Choose the initial values of algorithm parameters including maximum number of generation (*max_gen*), number of employed bees and onlooker bees (*num_employ* and *num_onlooker*), the maximum times that food sources do not change their locations (*limit*).

2. Generate *num_employ* and *num_onlooker* solutions, calculate the fitness value, and select the best half of solutions as foods.

3. In order to produce a candidate food position from the old one in memory, the ABC uses the following equation:

$$v_{ij} = x_{ij} + R_{ij}(x_{ij} - x_{kj}) \tag{5}$$

Where v_{ij} is a new position of food, R_{ij} is a random number in the range [-1,1], $k \in \{1,2,3,\mathrm{K},N\}$ and $k \neq i$. Calculate the fitness of v_i, and when the fitness of v_i is better than x_i, replace x_i with v_i. If x_i is not changed, $failure_i = failure_i + 1$.

4. The probabilities for onlookers are calculated as follows.

$$p_i = \frac{fit_i}{\sum_{n=1}^{N} fit_n} \tag{6}$$

Where fit_i is the fitness value of i-th bee.

5. According to p_i, onlooker bees search the area as employed bees.

6. If $failure_i > limit$, then scout bee randomly generates a new solution according to the following equation:

$$x_{ij} = x_{min\ j} + rand(0,1)(x_{max\ j} - x_{min\ j}) \tag{7}$$

7. Record the best solution at each generation.

8. If *max_gen* is reached or a satisfactory solution is found, then stop; otherwise go to step 3.

3.4. Summary of Our Proposed Algorithm

(1) Create the initial population randomly, containing structures and their corresponding parameters.
(2) Structure optimization is achieved by the structure operators as described in Subsection 3.1. Fitness function is calculated by mean square error or root mean square error.
(3) According to fitness value, sort the population. At some interval of generations, select certain percentage of population to optimize parameters. Parameter

optimization is achieved by ABC as described in Subsection 3.3. During this process, the structure of model is fixed.

(4) If the maximum number of generations is reached or a satisfactory solution is found, then stop; otherwise go to step (2).

4. Experimental Results and Analysis

To test the effectiveness of the proposed method, our method is applied to a synthetic data obtained by biochemical pathway and two real time-series expression datasets from Human cell time-series dataset and E. coil dataset. The parameters setting in this experiment is shown in Table 1.

Table 1. Parameters for Experiment

Parameters	Exp. 1	Exp. 2	Exp. 3
Population size	20	30	30
Generation	30	100	100
Crossover rate	0.7	0.7	0.7
ABC employed bees size	15	15	15
ABC onlooker bees size	15	15	15
ABC generation	200	200	200
ABC limit	20	20	20
Step size	0.01	0.05	0.05
Data points	100	45	35

4.1. Experiment with Biochemical Pathway

A biochemical pathway is used in this part, which is described as followed [25]:

$$\begin{aligned} X_1 + X_2 &\xrightarrow{k_1} X_3 \\ X_3 &\xrightarrow{k_2} X_2 + X_4 \end{aligned} \tag{8}$$

The corresponding rate equations for all four species are described as followed:

$$\begin{cases} \dfrac{dX_1}{dt} = -2X_1X_2 \\ \dfrac{dX_2}{dt} = -2X_1X_2 + 1.2X_3 \\ \dfrac{dX_3}{dt} = 2X_1X_2 - 1.2X_3 \\ \dfrac{dX_4}{dt} = 1.2X_3 \end{cases} \tag{9}$$

The initial conditions are created randomly and the synthetic time series are generated by solving the differential equations. In this experiment, the time series is from 0 to 99, including 100 time point. The top eighty percent of the data are used for training, and the rest are used to evaluate the performance of the model. The used instruction sets to create an optimal additive expression tree model is $I_1 = F \bigcup T = \{+2, +3, +4\} \bigcup \{*, x_0, x_1, x_2, x_3\}$. Experimental parameters are shown in Table 1.

The evolved biochemical pathway as the best solution is obtained as following:

$$\begin{cases} \frac{dX_1}{dt} = -2.00000X_1X_2 \\ \frac{dX_2}{dt} = -2.00000X_1X_2 + 1.200387X_3 \\ \frac{dX_3}{dt} = 1.992833X_1X_2 - 1.195695X_3 \\ \frac{dX_4}{dt} = 1.199763X_3 \end{cases} \quad (10)$$

The time series generated from the above ODEs is shown in Figure. 2 along with the target data. It can be seen that our predicted time-series data is very close to the target one.

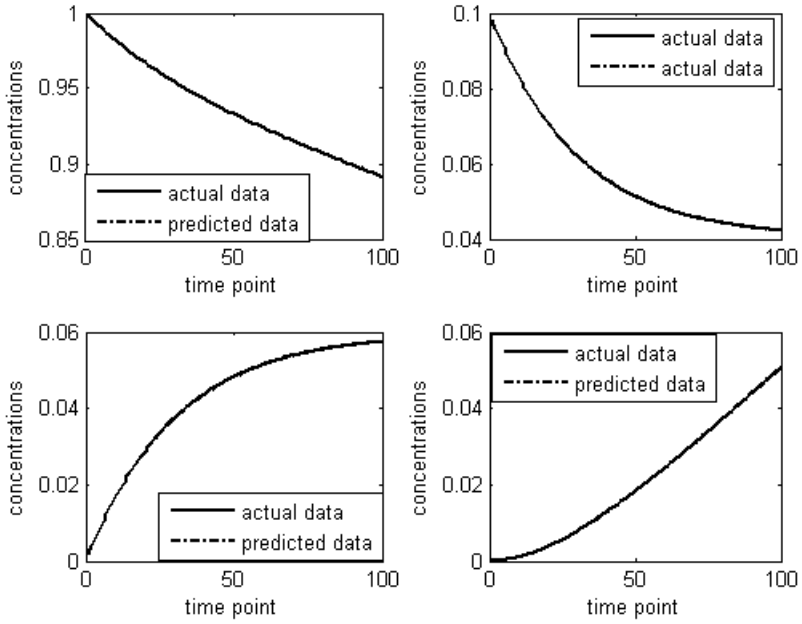

Figure 2. Time Series Of Acquired Model With All Four Species $\{X_1, X_2, X_3, X_4\}$, The Last 20 Points Dataset Is Used To Testing The Model

In order to investigate the model with different time points to the effect of results, we choose 20, 40, 60, 80 and 100 time points to make a comparative experiment. Experimental parameters in Table 1 are also used. The MSE performances of applied different time points are given in Table 2. From the results, we can see that time points are more, the evolved model is closer to the target one and the MSE is smaller. But with the smallest time point, the predicted time-series data of our method is also quite close to the target one. Gene expression data is usually with a few time points, so our method has ability to infer gene regulatory network with expression data.

Table 2. MSE of Time-Series Prediction with Different Time Points

Number of time points	Dataset	X_1	X_2	X_3	X_4
20	Training	3.4×10^{-6}	9.1×10^{-7}	2.2×10^{-6}	3.05×10^{-6}
	Testing	6.24×10^{-5}	2.6×10^{-5}	9.3×10^{-5}	5.6×10^{-5}

40	Training	7.71×10^{-7}	6.98×10^{-7}	8.36×10^{-7}	7.0×10^{-7}
	Testing	1.04×10^{-6}	2.31×10^{-6}	4.52×10^{-6}	8.3×10^{-7}
60	Training	7.68×10^{-7}	1.25×10^{-7}	1.69×10^{-7}	8.34×10^{-7}
	Testing	2.58×10^{-6}	3.12×10^{-7}	7.48×10^{-7}	4.12×10^{-6}
80	Training	1.46×10^{-7}	6.45×10^{-8}	7.68×10^{-8}	2.78×10^{-7}
	Testing	3.81×10^{-7}	9.21×10^{-8}	8.69×10^{-8}	5.67×10^{-7}
100	Training	1.2×10^{-7}	5.78×10^{-8}	6.84×10^{-8}	2.36×10^{-8}
	Testing	3.24×10^{-7}	7.87×10^{-7}	4.51×10^{-7}	3.28×10^{-7}

To test the validity of the additive expression tree model, we also compare inference of synthetic data using genetic programming and additive tree model. We choose 40 time points to infer the biochemical pathway. After searching the best solution, the results are listed in Table 3. From the empirical results, it is evident that the proposed method is powerful than genetic programming and additive tree mode, both in the aspects of accuracy and runtime.

Table 3. Comparison Results Using the Proposed Method, Genetic Programming and Additive Tree Model

	Our method	Genetic programming	Additive tree model
runtime	63.5s	132.2s	89.0s
MSE	4.21×10^{-6}	4.85×10^{-4}	3.29×10^{-5}

4.2. Experiment with the Human Cell Time-Series Data

The time series about Human cell are from the experiments performed by Whitfield [26]. The purpose is to identify genes which are periodically expressed in cell cycles. The experiment contains 1134 genes and the time series are divided into five subsets. We choose the first 5 genes expression time series with 46 time points for our purpose.

The used instruction sets to create an optimal additive expression tree model is $I_1 = F \bigcup T = \{+2,+3,+4,+5,+6,+7,+8\} \bigcup \{*, x_0, x_1, x_2, x_3, x_4\}$. Experimental parameters are shown in Table 1. The first 40 points are used for training, the rest 5 points are used to evaluate the performance of the model. By applying our method, we have acquired the following ODEs:

$$\begin{aligned}
\frac{dx_1}{dt} &= 121.132x_2x_3 - 20.954x_5x_5 - 2.815x_4x_5 - 48.126x_1 - 43.029x_2x_5 + 23.192x_4 + 22.186x_3 \\
\frac{dx_2}{dt} &= -68.097x_2x_5 + 7.427x_1x_3 + 0.684x_1 - 36.707x_2 - 14.145x_3 + 18.136x_5 + 45.390x_4x_4 \\
\frac{dx_3}{dt} &= -23.155x_2 - 90.015x_2x_2 + 49.041x_1x_2 - 31.479x_3 + 51.168x_4x_4 + 21.164x_1 - 8.629x_5x_1 \\
\frac{dx_4}{dt} &= 40.879x_3x_1 - 43.639x_4 + 41.395x_5 + 14.960x_1 + 63.439x_2x_5 \\
\frac{dx_5}{dt} &= 10.692x_2 + 28.482x_4 - 9.451x_3x_4 - 37.375x_5 + 3.006x_3x_1 - 2.505x_1 + 19.618x_5x_3
\end{aligned} \tag{11}$$

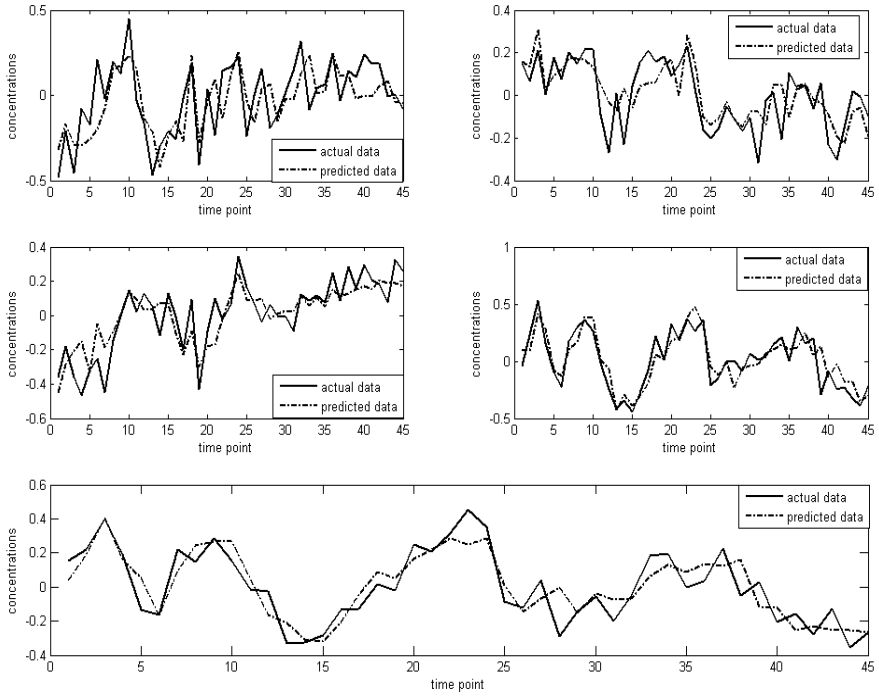

Figure 3. Dynamic Of Expression Level Of The Five Genes $\{x_1, x_2, x_3, x_4\}$ In Human Cell By 45 Times, The Rest 5 Points Are Used To Evaluate The Performance Of The Model

Figure. 3 shows dynamic of expression level of the five genes in Human cell. It can be seen that time-series data from our evolved model is quite close to the target one. Table 4 lists the comparison of five genes among genetic programming, additive tree model and our model. From Table 4, we can see that the prediction of our model is more accurate than the previously published methods.

Table 4. MSE of Time-Series Prediction with Three Methods

Methods	Dataset	x_1	x_2	x_3	x_4	x_5
Genetic programming	Training	0.101	0.082	0.063	0.086	0.058
	Testing	0.25	0.14	0.121	0.206	0.087
Additive tree model	Training	0.0293	0.0234	0.0208	0.0302	0.0197
	Testing	0.0594	0.0769	0.0268	0.0814	0.0442
Our method	Training	0.0272	0.0121	0.0135	0.0154	0.0106
	Testing	0.0341	0.0315	0.0129	0.0306	0.0308

4.3. Experiment with the E. Coli Database

In this part, to test the performance of reconstruction of our method in the real microarray data containing noise, we use the gene-expression data from E. coil dataset.

The measurement data is original from polymicrobial probe database (Many Microbe Microarrays M3D database), version 4 build 6 [27]. The data has 907 experiments and 4297 genes. We choose the first 35 experiments and 8 genes (**Crp**, **araC**, **nagC**, **chbC**, **araE**, **araA**, **chbA** and **chbF**) for our experiment.

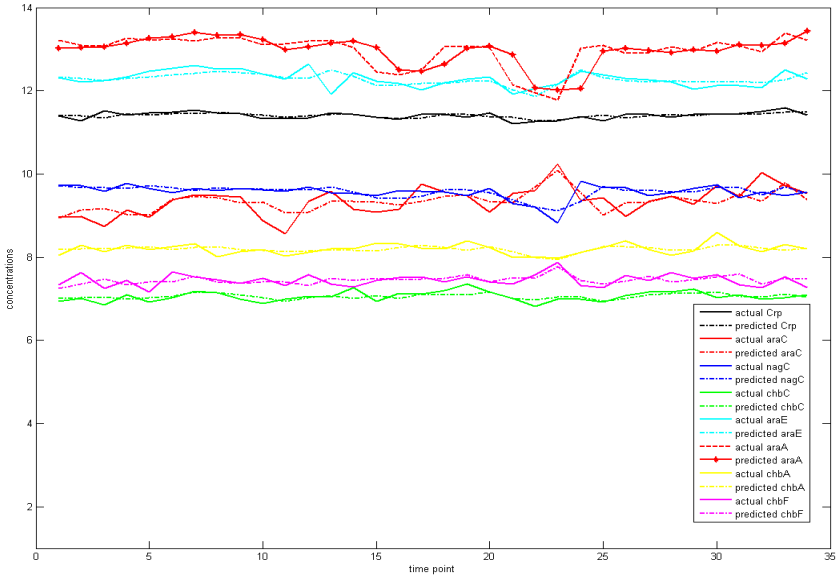

Figure 4. Dynamic Of Expression Level Of The 8 Genes In E. Coli Database, The Rest 5 Points Are Used To Evaluate The Performance Of The Model

The used instruction sets to create an optimal additive expression tree model is

$$I_1 = F \bigcup T = \{+2,+3,+4,+5,+6,+7,+8\} \bigcup \{*,/,\sin,\cos,\log,\tan,\frac{x^2}{1+x^2},\frac{1}{1+x^2},\frac{1}{1+e^{-x}},x_0,x_1,\ldots,x7\}$$

. Experimental parameters are shown in Table 1. The first 30 points are used for training, and the rest 5 points are used to evaluate the performance of the model. By applying our method, we have acquired the following

$$\frac{dx_1}{dt} = 69.745\frac{1}{1+e^{x_7}} - 30.836\tan(x_1) - 65.032\frac{x_1}{x_6} - 2.897\cos(x_2)$$

$$\frac{dx_2}{dt} = 37.130\cos(x_2) + 95.400\cos(x_8) + 44.717\cos(x_3) + 91.204\cos(x_1)$$

$$\frac{dx_3}{dt} = 14.696\cos(x_3) - 0.002e^{x_3} + 0.277x_1x_7 + 16.766\cos(x_6)$$

$$\frac{dx_4}{dt} = 34.325\tan(x_4) + 42.496\cos(x_1) + 2.219x_3 - 17.205\tan(x_8)$$

$$\frac{dx_5}{dt} = 15.783\cos(x_6) - 100.177\tan(x_5) + 9.313x\frac{x_6}{x_1} + 76.925\frac{1}{1+e^{x_3}}$$

$$\frac{dx_6}{dt} = 12.394\frac{1}{1+x_6^2} + 35.379\tan(x_1) - 91.374\log(x_6) + 87.824\log(x_5)$$

$$\frac{dx_7}{dt} = -2.071x_6 - 0.007e^{x_7} + 0.0016e^{x_3} + 0.277x_1x_2$$

$$\text{ODEs: } \frac{dx_8}{dt} = -25.292\tan(x_7) + 30.609\cos(x_3) + 19.295\log(x_4) - 0.0239e^{x_8} \quad (12)$$

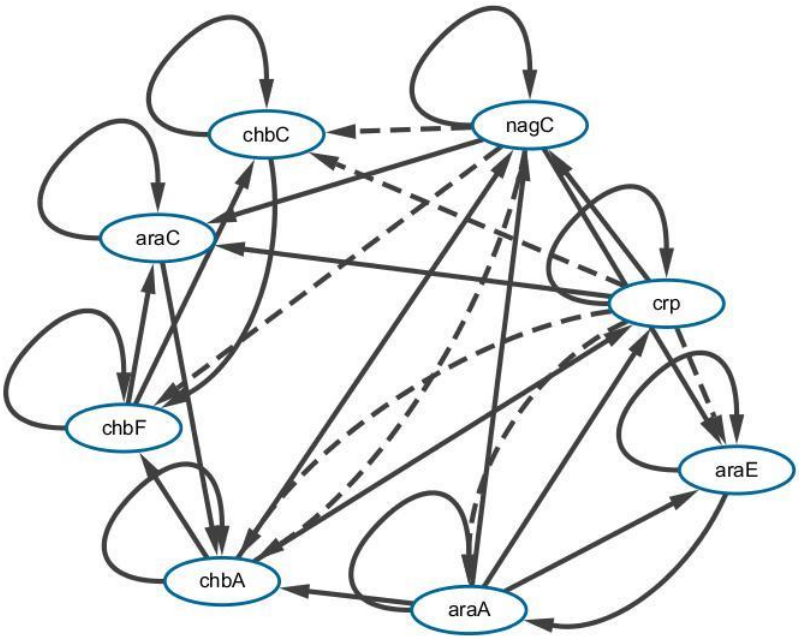

Figure 5. Inferred Regulatory Relationships Among 8 Genes. Correctly Identified Edges Are Drawn With Dashed Lines And Falsely Identified Edges With Solid Lines

Figure. 4 shows dynamic of expression level of the 8 genes in E. coli database. It can be seen that time-series data from our evolved model is quite close to the target one. By our method, the detail regulatory relationships among 8 genes are given in Figure. 5. from Figure. 5, we can see that as quite important regulatory factor, **Crp** regulates **chbC**, **araE**, **araA** and **chbA**, while **nagC** regulates **chbC**, **chbF** and **chbA**. These relationships are in agreement with biological experimental finding from RegulonDB [28].

5. Conclusion and Discussion

In this paper, an additive expression tree model and its design and evolved method are proposed to infer gene regulatory networks. The additive expression tree model could be used to encode the ordinary differential equations. A new structure-based evolutionary algorithm is used to optimize the architecture of system and corresponding parameters are evolved by artificial bee colony. The proposed method has been verified by synthetic data and two real time-series expression datasets. Experiment results reveal that the proposed method performs well, both in the aspects of accuracy and runtime.

The following factors may lead to the good performance of our method. First, the additive expression tree model contains linear entities. Thus user may reduce the difficulties of program implementation. Due not to handle the nonlinear structure, runtime reduces sharply. Second, the form of additive expression tree model is very near to the representation of the ODE system which we need identify. Third, the additive root node could make the nonlinear ODE system divided into several blocks, which reduces the problem complexity.

Acknowledgements

This work was supported by the PhD research startup foundation of Zaozhuang University (No.1020702).

References

[1] M. Maienschein-Cline, J. Zhou, K. White, R. Sciammas and A. Dinner, "Discovering Transcription Factor Regulatory Targets Using Gene Expression and Binding Data", Bioinformatics, vol. 28, **(2011)**, pp. 206-213.
[2] N. Chemmangattuvalappil, K. Task and I. Banerjee, "An integer optimization algorithm for robust identification of non-linear gene regulatory networks", BMC Syst. Biol., vol. 2, no. 6, **(2013)**, p. 119.
[3] S. A. Thomas and Y. C. Jin, "Reconstructing biological gene regulatory networks: where optimization meets big data", Evol. Intel., vol. 7, **(2014)**, pp. 29-47.
[4] L. Windhager, J. Zierer and R. Küffner, "Refining Ensembles of Predicted Gene Regulatory Networks Based on Characteristic Interaction Sets", PloS ONE, vol. 9, no. 2, **(2014)**, pp. e84596.
[5] J. Ruyssinck, V. A. Huynh-Thu, P. Geurts, T. Dhaene, P. Demeester and Y. Saeys, "NIMEFI: Gene Regulatory Network Inference using Multiple Ensemble Feature Importance Algorithms", PLoS ONE, vol. 9, no. 3, **(2014)**, pp. e92709.
[6] A. Graudenzi, R. Serra, M. Villani, C. Damiani, A. Colacci and S. A. Kauffman, "Dynamical Properties of a Boolean Model of Gene Regulatory Network with Memory", Journal of Computational Biology, vol. 18, no. 10, **(2011)**, pp. 1291-1303.
[7] H. J. Ouyang, J. Fang, L. Z. Shen, E. R Dougherty and W.B. Liu, "Learning restricted Boolean network model by time-series data", EURASIP Journal on Bioinformatics and Systems Biology, vol. 2014, **(2014)**, pp. 10.
[8] W. C. Young, A. E. Raftery and K. Y. Yeung, "Fast Bayesian inference for gene regulatory networks using Scan BMA", BMC Systems Biology, vol. 8, **(2014)**, pp. 47.
[9] B. Yang, Y. H. Chen and M. Y. Jiang, "Reverse engineering of gene regulatory networks using flexible neural tree models", Neurocomputing, vol. 99, **(2013)**, pp. 458-466.
[10] C. Chaouiya, "Petri net modelling of biological networks", Brief Bioinform., vol. 8, no. 4, **(2007)**, pp. 210-219.
[11] C. Chaouiyaa, E. emyb and D. Thieffrya, "Petri net modelling of biological regulatory networks", Journal of Discrete Algorithms, vol. 6, no. 2, **(2008)**, pp. 165-77.
[12] O. P. Tabbaa and C. Jayaprakash, "Mutual information and the fidelity of response of gene regulatory models", Phys. Biol., vol. 11, no. 4, **(2014)**, pp. 046004.
[13] Y. H. Chen, B. Yang, Q. F. Meng, Y. O. Zhao and A. Abraham, "Time-series forecasting using a system of ordinary differential equations", Information Sciences, vol. 181, no. 1, **(2010)**, pp. 106-114.
[14] D. Tominaga, N. Koga and M. Okamoto, "Efficient Numerical Optimization Algorithm Based on Genetic Algorithm for Inverse Problem", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO00), **(2000)**, pp. 251-258.
[15] L. Palafox, N. Noman and H. Iba, "Reverse Engineering of Gene Regulatory Networks Using Dissipative Particle Swarm Optimization", IEEE Transactions on Evolutionary Computation, vol. 17, no. 4, **(2013)**, pp. 577-587.
[16] R. Xu, I. I. D. Wunsch and R. Frank, "Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization", IEEE/ACM Trans Computat. Biol. Bioinform., vol. 4, no. 4, **(2007)**, pp. 681-692.
[17] N. Noman, L. Palafox and H. Iba, "Reconstruction of Gene Regulatory Networks from Gene Expression Data Using Decoupled Recurrent Neural Network Model", Natural Computing and Beyond, vol. 6, **(2013)**, pp. 93-103.
[18] 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.
[19] O. Mihai, "A Comparison of Several Linear Genetic Programming Techniques", Complex Systems, vol. 14, no. 4, **(2003)**, pp. 285-313.
[20] C. Ferreira, "Gene Expression Programming: A New Adaptive Algorithm for Solving Problems", Complex Systems, vol. 13, no. 2, **(2001)**, pp. 87-129.
[21] R. Frigg and S. Hartmann, "Models in Science", in: The Stanford Encyclopedia of Philosophy, Stanford University, Stanford, CA, USA (**2006)**.
[22] K. Chellapilla, "Evolving computer programs without subtree crossover", IEEE Transactions on Evolutionary Computation, vol. 1, **(1997)**, pp. 209-216.
[23] X. Y. Cheng and M. Y. Jiang, "An improved artificial bee colony algorithm based on Gaussian mutation and chaos disturbance", Lecture Notes in Computer Science, vol. 7331, **(2012)**, pp. 326-333.
[24] D. Karaboga, B. Gorkemli, C. Ozturk and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications", Artificial Intelligence Review, vol. 42, no. 1, **(2014)**, pp. 21-57.

[25] J. Srividhya, S. Schnell, E. J. Crampin and P. E. McSharry, "Reconstructing biochemical pathways from time course data", Proteomics, vol. 7, no. 6, **(2007)**, pp. 828-838.
[26] M. L. Whitfield, G. Sherlock, A. J. Saldanha, J. I. Murray, C. A. Ball, K. E. Alexander, J. C. Matese, C. M. Perou, M. M. Hurt, P. O. Brown and D. Botstein, "Identification of genes periodically expressed in the human cell cycle and their expression in tumors", Molecular Biology of the Cell, vol. 13, **(2002)**, pp. 1977-2000.
[27] J. Faith, M. Driscoll, V. Fusaro, E. Cosgrove, B. Hayete, F. Juhn, S. Schneider and T. Gardner, "Many Microbe Microarrays Database: uniformly normalized Affymetrix compendia with structured experimental metadata", Nucleic Acids Res, vol. 36, **(2008)**, pp. D866-D870.
[28] S. Gama-Castro, H. Salgado, M. Peralta-Gil, A. Santos-Zavaleta, L. Muñiz-Rascado, H. Solano-Lira, V. Jimenez-Jacinto, V. Weiss, J. S. García-Sotelo, A. López-Fuentes, L. Porrón-Sotelo, S. Alquicira-Hernández, A. Medina-Rivera, I. Martínez-Flores, K. Alquicira-Hernández, R. Martínez-Adame, C. Bonavides-Martínez, J. Miranda-Ríos, A. M. Huerta, A. Mendoza-Vargas, L. Collado-Torres, B. Taboada, L. Vega-Alvarado, M. Olvera, L. Olvera, R. Grande, E. Morett and J. Collado-Vides, "RegulonDB version 7.0: transcriptional regulation of Escherichia coli K-12 integrated within genetic sensory response units (Gensor Units)", Nucleic Acids Res, vol. 39, no. suppl 1, **(2011)**, pp. D98-D105.

Author

Bin Yang is the teacher of Zaozhuang University. He has pursued 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 Engineering from University of Jinan. His research interests include hybrid computational intelligence and its applications in time-series prediction, system identification and gene regulatory network**.**