ISBN 978-981-11-3671-9
Proceedings of 2017 the 7th International Workshop on Computer Science and Engineering (WCSE 2017)
Beijing, 25-27 June, 2017, pp. 337-343
doi: 10.18178/wcse.2017.06.058

Infer Gene Regulatory Network Using the Novel Nonlinear Differential Equation System

Wei Zhang and Bin Yang[+]

School of Information Science and Engineering, Zaozhuang University, Zaozhuang, China 277160

Abstract. Gene regulatory network (GRN) is very complex and nonlinear dynamics system. In this paper, we present a novel nonlinear ordinary differential equation (ODE) model based on complex-valued flexible neural tree (ODECVFNT) to improve the accuracy of GRN inference. Complex-valued flexible neural tree (CVFNT) model is proposed to model the nonlinear regulation function in an ODE model. The hybrid evolutionary method based on structure-based evolutionary algorithm and cuckoo search (CS) is used to evolve the structure and parameter of ODECVFNT. Benchmark datasets from Dialogue for Reverse Engineering Assessments and Methods challenge are used to test our method. Results reveal that our proposed method can infer more correctly gene regulatory network than the popular method LASSO and real-valued flexible neural tree (RVFNT) model.

Keywords: gene regulatory network, complex-valued, flexible neural tree model, ordinary differential equation, cuckoo search

1. Introduction

Reverse engineering of gene regulatory network (GRN) using expression data could provide new insights into understanding inherent law of life phenomenon and analyzing complex diseases, and has an important role in the field of systems biology [1-2]. Many models have been proposed to identify GRNs, such as Boolean network [3], Bayesian network [4], differential equation [5], Gaussian graphical model [6] and information theory [7].

The system of ordinary differential equation (ODE) belongs to a sophisticated and well established class of methods, which can offer realistic representation of genetic networks due to their continuity. Thus ODEs has become the most common method of describing GRNs [8].Yeung et al. proposed a scheme to reverse-engineer gene networks using singular value decomposition and robust regression based on a linear additive model [9]. However gene regulatory network is a complex system. To accurately capture the properties of true gene network, nonlinear ODE model were proposed. Mazur et al. reconstructed nonlinear differential equation model of gene regulation using stochastic sampling. S-system model, a type power law formalism, has been the most popular nonlinear differential equation model to infer GRN [11, 12].

As the classic modeling method, neural network (NN) models have been successfully applied to gene regulatory network reconstruction and time-series prediction from gene expression profiling in recent years, such as recurrent neural networks (RNN) [13], recurrent Elman neural networks (RENN) [15] and neural fuzzy recurrent network (NFRN) [14]. Compared with the fully connected neural network, the flexible neural tree (FNT) is more flexible and easier to approximate the unknown functions, and supports feature selection and over-layer connections [16]. In the past decade, complex-valued neural network (CVNN) has been proposed to solve prediction and classification problems. Compared with real-valued neural network, CVNN is more flexible and functional. Xiong et al proposed the fully complex-valued radial basis function neural

[+] Corresponding author. Tel.: +18369296801.
E-mail address: batsi@126.com.

networks (FCRBFNNs) to investigate the possibility of forecasting interval time series [17]. Shamima et al. used a Fully Complex-valued Relaxation Network (FCRN) classifier to predict the secondary structure of proteins [18].

The ODE used to infer gene regulatory network could be divided into two parts: regulation function and the self-degradation part. In this paper, complex-valued flexible neural tree (CVFNT) is proposed to approximate the regulation function, namely ODE based on CVFNT (ODECVFNT). Genetic programming (GP) like tree structure-based evolutionary algorithm and cuckoo search are used to evolve the structure and parameter of ODECVFNT, respectively. Simulated time series data obtained by from the DREAM3 challenge about Yeast knock-out genes with size 50 and 100 are test our method. Results show that our method is capable of correctly identifying gene regulatory network.

2. Materials and Methods

2.1. The ordinary differential equation model

The ordinary differential equation model is the common dynamic system and usually used to simulate the evolution of biological macromolecules with time. In order to identify gene regulatory network, one ODE is used to represent regulatory relationships between target gene and regulatory factors. The formal of one ODE is described as follows.

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \ldots, x_n) - \beta_i x_i. \tag{1}$$

where x_i is the express level of gene i, β_i is the self-degradation rate. $f_i(\ )$ is the regulation function containing linear, piecewise linear, pseudo linear (Sigmoid function) and nonlinear functions. The number of parameters and topology in $f_i(\ )$ determine regulation strengths. To better model regulation function, we propose use the complex-valued flexible neural tree model to model the regulation functions $f_i(\ )$. The formal of ODE based on CVFNT (ODECVFNT) described as Eqs.(2) and in Fig. 1.

$$\frac{dx_i}{dt} = CVFNT_i - \beta_i x_i. \tag{2}$$

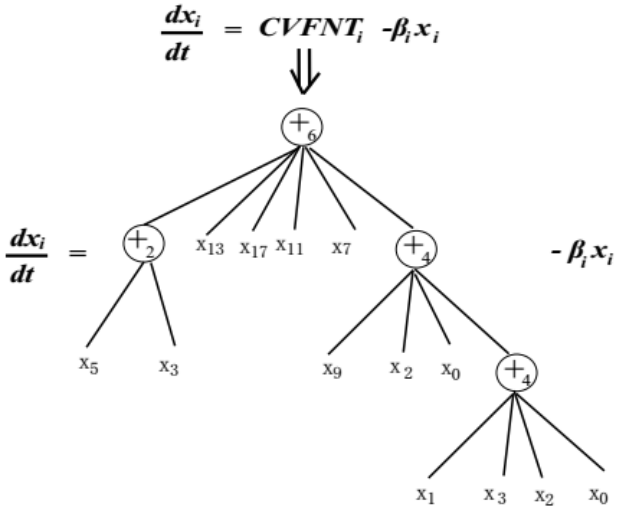

Fig. 1: The formal of the differential equation model based on complex-valued flexible neural tree.

2.2. Complex-valued flexible neural tree model

Complex-valued flexible neural tree (CVFNT) model is the extensions of real-valued FNT model. In a CVFNT, input layer, threshold values and weights are complex numbers. A tree-structural based encoding method with specific instruction set is selected for representing a CVFNT model. The used function set F and terminal instruction set T for generating a CVFNT model are described as follows:

$$S = F \bigcup T = \{+_2, +_3, \ldots, +_N\} \bigcup \{z_1, z_2, \ldots, z_n\}, \tag{3}$$

where $+_i (i = 2,3,\ldots N)$ denotes non-leaf nodes' instructions and taking *I* arguments. $z_1, z_2, \ldots z_n$ ($z_i \in C^n$, $z_i = x_i + jy_i$ and *j* stands for the value of $\sqrt{-1}$.) are leaf nodes' instructions and taking no other arguments. The output of a non-leaf node is calculated as a flexible neuron model (see Fig. 2). From this point of view, the instruction $+_i$ is also called a flexible neuron operator with *i* inputs.

In the creation process of neural tree, the operator is selected randomly from function set *F* and terminal instruction set *T*. If a non-terminal instruction, i.e., $+_i (i = 2,3,\ldots,N)$ is selected, the *i* complex-valued weights ($w_1, w_2, \ldots w_i$) are randomly generated and used for representing the connection strength between the node $+_i$ and its children.

The output of a flexible neuron $+_n$ can be calculated as follows. The total excitation of $+_n$ is

$$net_n = w_0 + \sum_{j=1}^{n} w_j z_j \tag{4}$$

where w_0 is threshold value and $z_j (j = 1,2,\ldots,n)$ are the inputs to node $+_n$. The output of the node $+_n$ is then calculated by

$$out_n = f(c, r, net_n) = \frac{net_n}{c + \frac{1}{r}|net_n|}. \tag{5}$$

where *f*( ) is activation function, *c* and *r* are real variables, and $|net_n|$ is the modulus of complex net_n. The output of flexible activation function is complex.

A typical complex-valued flexible neural tree model is shown as Fig. 3. The overall output of complex-valued flexible neural tree can be computed from left to right by depth-first method, recursively.

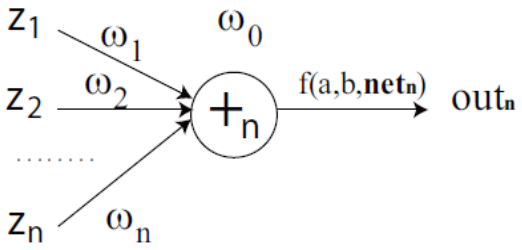

Fig. 2: A flexible neuron operator.

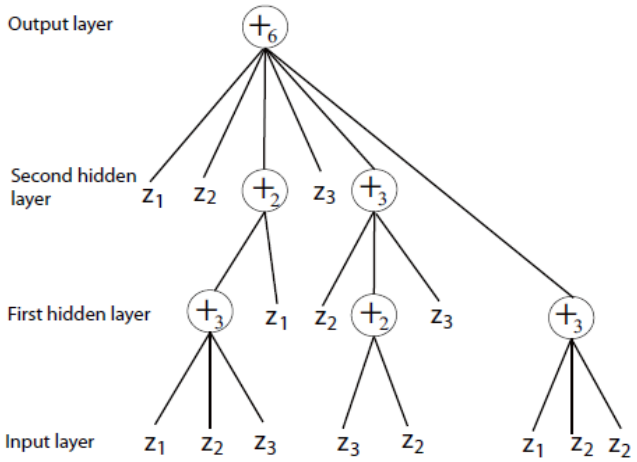

Fig. 3: A typical representation of neural tree with function instruction set $F = \{+_2, +_3, +_4, +_5, +_6\}$, and terminal instruction set $T = \{z_1, z_2, z_3\}$.

2.3. Structure optimization of ODECVFNT

ODECVFNT contains two parts: CVFNT model and the self-degradation part (Eqs. 2). In order to optimize structure of ODECVFNT model, it need only optimize structure of CVFNT model. Finding an optimal or near-optimal CVFNT is formulated as an evolutionary search process. In this paper, we use three kinds of neural tree variation operators: mutation, crossover and selection. Mutation could change the current neural trees using mutation operators. Crossover could exchange two neural trees from current population according to crossover probability. Mutation and crossover could both generate new offsprings for the next generation. Select is applied to select the parents for the next generation according to the fitness value. The three kinds of neural tree variation operators are described in Ref [16].

2.4. Parameters optimization of ODECVFNT

ODECVFNT model has two kinds of parameters: parameters of CVFNT and the self-degradation parameter β. In a CVFNT model, weights w_i and threshold value w_0 are complex valued. In the optimization process, both real and imaginary parts need to be optimized. c and r are real variables. The optimized parameter vector is $[\beta, \mathrm{Re}(w_i), \mathrm{Im}(w_i), \mathrm{Re}(w_0), \mathrm{Im}(w_0), c, r, \ldots]$. Because cuckoo search (CS) is potentially far more efficient than traditional evolutionary algorithms such as particle swarm optimization (PSO), genetic algorithms (GA), CS is proposed to optimize the parameters of S-system model.

The process of CS is described as followed [20].

(1) Create the initial population randomly $X_i (i = 1, 2, \ldots n)$, which represent host nests.

(2) Compute the fitness values of all population. If the maximum number of generations is reached or a satisfactory solution is found, then stop.

(3) Lévy flight is performed to generate new solutions.

$$X_i^{t+1} = X_i^t + \alpha \oplus L\acute{e}vy(\lambda) \tag{6}$$

where X_i^t is the i-th solution at t-th generation, α is step size which could control the scale of random search. In general, $\alpha = 1$. $\oplus$ means entrywise multiplications, $L\acute{e}vy(\lambda)$ abides by Lévy probability distribution:

$$L\acute{e}vy(\lambda) \ \ u = t^{-1-\lambda}, \ 0 < \lambda \le 2 \tag{7}$$

$L\acute{e}vy(\lambda)$ could be computed using the following equation :

$$L\acute{e}vy(\lambda) = \frac{\phi \times \mu}{|v|^{\frac{1}{\lambda}}}$$

$$\phi = \left(\frac{\Gamma(1+\lambda) \times \sin(\Pi \times \frac{\lambda}{2})}{\Gamma((\frac{1+\lambda}{2}) \times \lambda \times 2^{\frac{\lambda-1}{2}})}\right)^{\frac{1}{\lambda}} \tag{8}$$

where μ and v follow Gaussian distributions.

(4) According to predefined probability p_a, discard the worse solutions. Create the same number of new solutions using preference random walk.

$$X_i^{t+1} = X_i^t + r(X_m^t - X_n^t) \tag{9}$$

where r is scaling factor, which is created randomly from [0, 1]. X_m^t and X_n^t are random solutions at t-th generation. Go to step (2).

2.5. Flowchart of our method

In order to infer the regulatory factor of each target gene, the optimal ODECVFNT model need be searched using hybrid evolutionary method. The optimal ODECVFNT of target gene i is found as follows:

(1) Gene expression data of target gene i are used as output data and expression data of other genes are used as input data. Gene expression data are real numbers, so input data need to be transformed to complex

values before search an optimal ODECVFNT. Suppose the input real numbers $[x_1, x_2, \ldots, x_m]$, and tally the maximum and minimum values of real numbers (max and min). i-th real number x_i is transformed into complex number as followed [10].

$$\varphi_i = \frac{x_i - min}{max - min}(2\pi - \delta),$$
$$z_i = e^{i\varphi_i} \tag{10}$$

where δ is the shift angle and i stands for the value of $\sqrt{-1}$.

(2) According to transformed complex numbers, find an optimal or near-optimal ODECVFNT.

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 2.3. Fitness function is calculated by root mean square error (RMSE)

$$RMSE_i = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_i(t) - \overline{y_i}(t))^2}. \tag{11}$$

where N is the number of sample points of gene expression, $y_i(t)$ is actual expression data of gene i at t-th point, and $\overline{y_i}(t)$ is predicted expression data of gene i at t-th point.

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 CS as described in Subsection 2.4. 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).

(3) Train the CVFNT model according to the above complex numbers (input data) and firefly algorithm, which is introduced detailed in Section 2.2 and 2.3. The output of CVFNT is complex number, so the output value needs to be transformed into real number in order to evaluate model effectively. The inverse transformation shall be used:

$$\arg z = \varphi,$$
$$y = \frac{\varphi(max - min)}{2\pi - \delta} + min. \tag{12}$$

where $\arg z$ is the argument of complex value z.

3. Experimental Results and Illustrative Examples

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 about Yeast knock-out genes with size 50 and 100 [23]. To test the validate of our method, LASSO [24] and real-valued flexible neural network (RVFNT) model are also used to infer gene regulatory network with the same data.

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.

$$TPR = TP/(TP + FN),$$
$$FPR = FP/(FP + TN),$$
$$PPV = TP/(TP + FP),$$
$$ACC = (TP + TN)/(TP + FP + TN + FN),$$
$$F - score = 2PPV * TPR/(PPV + TPR) \tag{13}$$

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 ODECVFNT is superior to the popular method LASSO and RVFNT model except for FPR.

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	0.132
RVFNT	0.377	**0.066**	0.156	0.904	0.221
ODECVFNT	**0.455**	0.071	**0.172**	**0.905**	**0.250**

Secondly our method is applied to the Yeast 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 LASSO and RVFNT.

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

	TPR	FPR	PPV	ACC	F-score
LASSO	0.349	0.107	0.052	0.878	0.092
RVFNT	0.398	0.055	0.110	0.929	0.172
ODECVFNT	**0.590**	**0.053**	**0.159**	**0.931**	**0.251**

4. Concluding Remarks

In this research paper, we propose a new ordinary differential equation model based on complex-valued flexible neural tree called ODECVFNT to identify gene regulatory network. In our proposed algorithm, a hybrid evolutionary method is proposed to optimize the structure and parameters of ODECVFNT. From results on the DREAM3 benchmark datasets, ODECVFNT model is effective and superior to LASSO and real-valued FNT model significantly.

5. Acknowledgments.

This research was supported by the Shandong Provincial Natural Science Foundation, China (No. ZR2015PF007).

6. References

[1] J.N. Bazil, K.D. Stamm, X. Li, R. Thiagarajan, T.J. Nelson, A. Tomita-Mitchell, D.A. Beard. The Inferred Cardiogenic Gene Regulatory Network in the Mammalian Heart. *PLoS One*. 2014, 9(6): e100842.

[2] F. Emmert-Streib, M. Dehmer, B. Haibe-Kains. Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. *Front Cell Dev Biol.* 2014, 2: 38.

[3] P. Trairatphisan, A. Mizera, J. Pang, A.A. Tantar, J. Schneider, T. Sauter. Recent development and biomedical applications of probabilistic Boolean networks. *Cell Commun Signal*. 2013, 11: 46.

[4] E. Acerbi, T. Zelante, V. Narang, F. Stella. Gene network inference using continuous time Bayesian networks: a comparative study and application to Th17 cell differentiation. *BMC Bioinformatics*. 2014, 15(1): 387.

[5] Y.K. Wang, D.G. Hurley, S. Schnel, C.G. Print, E.J. Crampin. Integration of SteadyState and Temporal Gene Expression Data for the Inference of Gene Regulatory Networks. *PLoS One*. 2013, 8(8): e72103.

[6] P. Ingkasuwan, S. Netrphan, S. Prasitwattanaseree, M. Tanticharoen, S. Bhumiratana, A. Meechai, J. Chaijaruwanich, H. Takahashi, S. Cheevadhanarak. Inferring transcriptional gene regulation network of starch metabolism in Arabidopsis thaliana leaves using graphical Gaussian model. *BMC Syst Biol*. 2012, 6: 100.

[7] X.J. Zhang, J. Zhao, J.K. Hao, X.M. Zhao, L.N. Chen. Conditional mutual inclusive information enables accurate quantification of associations in gene regulatory networks. *Nucleic Acids Res*. 2015, 43(5): e31.

[8] B. Yang, M.Y. Jiang, Y.H. Chen. A Novel Hybrid Framework for Reconstructing Gene Regulatory Networks.

International Journal of Hybrid Information Technology. 2013, 6(5): 255–268.

[9] M.K.S. Yeung, J. Tegn'er, J.J. Collins. Reverse engineering gene regulatory networks using singular value decomposition and robust regression. *Proc Natl Acad Sci USA*. 2002, 99: 6163–6168.

[10] I. Aizenberg, L. Sheremetov, L. Villa-Vargas. Multilayer Neural Network with Multi-Valued Neurons in time series forecasting of oil production. *Neurocomputing*. 2015, 8495: 61–70.

[11] Y.T. Hsiao, W.P. Lee. Reverse engineering gene regulatory networks: Coupling an optimization algorithm with a parameter identification technique. *BMC Bioinformatics*. 2014, 15(Suppl 15): S8.

[12] E.O. Voit, J. Almeida. Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics*. 2004, 20: 1670–1681.

[13] R. Xu, I.I. Wunsch, R.L. Frank. Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization. IEEE/ ACM Trans. *Comput. Biol. Bioinformatics*. 2007, 4: 681–92

[14] I.A. Maraziotis, A. Dragomir, A. Bezerianosh. Gene networks reconstruction and time-series prediction from microarray data using recurrent neural fuzzy networks. *IET Syst. Biol.* 2007, 1(1): 41–50.

[15] S.I. Ao, V. Palade. Ensemble of Elman neural networks and support vector machines for reverse engineering of gene regulatory networks. *Appl. Soft Comput*. 2011, 11(2): 1718–1726.

[16] B. Yang, Y.H. Chen, M.Y. Jiang. Reverse engineering of gene regulatory networks using flexible neural tree models. *Neurocomputing*. 2013, 99: 458-466.

[17] T. Xiong, Y. Bao, Z. Hu, R. Chiong. Forecasting interval time series using a fully complex-valued RBF neural network with DPSO and PSO algorithms. *Information Sciences*. 2015, 305: 77–92.

[18] B. Shamima, R. Savitha, S. Suresh, S. Saraswathi. Protein secondary structure prediction using a fully complex-valued relaxation network. *The 2013 International Joint Conference on Neural Networks (IJCNN)*. 2013, 1–8.

[19] Y.H. Chen, B. Yang, J. Dong, A. Abraham. Time series forecasting using flexible neural tree model. *Inf. Sci.* 2005, 174(3/4): 219–235.

[20] X.S. Yang, S. Deb. Cuckoo Search: Recent Advances and Applications. *Neural Computing and Application*s. 2014, 24(1): 169–174.

[21] P. Civicioglu, E. Besdok. A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artificial Intelligence Review*. 2013, 39 (4): 315–346.

[22] L.Z. Liu, F.X. Wu, W.J. Zhang. Reverse engineering of gene regulatory networks from biological data. *WIREs Data Mining Knowl Discov*. 2012, 2(5): 365–385.

[23] D. Marbach, R.J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, G. Stolovitzky. Revealing strengths and weaknesses of methods for gene network inference. *Proc Natl Acad Sci USA*. 2010, 107(14): 6286–91.

[24] G. Geeven, R.E. van Kesteren, A.B. Smit, M.C. de Gunst. Identification of contextspecific gene regulatory networks with GEMULA-gene expression modeling using Lasso. *Bioinformatics*. 2012, 28: 214–221.