

# A New Supervised Learning for Gene Regulatory Network Inference with Novel Filtering Method

Bin Yang\*, Wei Zhang, and Jiaguo Lv

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

---

## Abstract

Gene regulatory network (GRN) inference from gene expression data plays an important role in understanding the intricacies of the complex biological regulations for researchers. In this paper, a new hybrid supervised learning method (HSL) is proposed to infer gene regulatory network. In HSL, according to the data imbalance ratio, three different supervised learning methods: direct classification, K-Nearest Neighbor (KNN) method and complex-valued version of flexible neural tree (CVFNT) model are chosen to classify. A novel filtering method based on integration of mutual information (MI) and maximum information coefficient (MIC) is proposed to eliminate the redundant regulations inferred by HSL. Benchmark data from DREAM 5 are used to test the performance of our approach. The results show that our approach performs better than the popular unsupervised Learning methods and supervised Learning methods.

*Keywords:* gene regulatory network; supervised learning; flexible neural network; bioinspired algorithm; information theory

(Submitted on January 29, 2018; Revised on March 2, 2018; Accepted on April 23, 2018)

© 2018 Totem Publisher, Inc. All rights reserved.

---

## 1. Introduction

Transcriptional regulation is the basis of some important biology processes such as development, differentiation, homeostasis and response to stimuli. Inference of gene regularity network (GRN) from expression data is a challenging task to understand the intricacies of the complex biological regulations for researchers [9]. With the development of microarray and next generation sequencing technology, a large amount of gene expression data and gene annotation information have been generated, which promotes the development of GRN inference by computer technology [12]. But, the inference of GRN from massive genome and transcriptome data still remains a big challenge.

Many machine-learning methods have been developed to infer gene regulatory networks ranging from unsupervised to supervised methods [5]. Unsupervised learning methods rely only on gene expression data and mainly contain Boolean network, Bayesian network, Petri network, ordinary differential equation (ODE), information theory model and neural network. The advanced sequencing technology gives a large number of prior biological knowledge along with genomics data, which promotes supervised techniques to infer gene regulatory network. According to prior regulation information, GRN inference is solved as binary classifier problem. Supervised learning methods utilize known regulation interactions to train a classifier in order to predict new unknown regulatory relationships [1]. The main binary classifiers contain support vector machine (SVM), decision tree, neural network and logistic regression [3]. With a set of prior known regulatory connections, supervised methods could achieve lower prediction accuracies than unsupervised methods [6].

However, the existing supervised learning methods are relatively simple and difficult to infer the complex regulatory relationships effectively. For each classifier, training data is unbalanced in general, which may lead to that GRN inferred has many false positive regulations. In this paper, a new hybrid supervised learning method (HSL) is proposed to infer gene regulatory networks. Firstly, the target gene set is divided into two classes with the known regulation relationships. Secondly, according to the level of data imbalance, different supervised learning methods are chosen to apply. If the number of positive samples is zero, it is not necessary to apply classifier and the test sample is directly classified as negative class. If

---

\* Corresponding author.

E-mail address: [batsi@126.com](mailto:batsi@126.com)

the ratio of positive samples is relatively small, K-Nearest Neighbor (KNN) is selected to identify whether two genes regulate or not. Complex-valued neural networks (CVNNs) are powerful mathematical methods that have been successfully applied to predict and classify actual observation data [4]. In terms of flexibility and functionality, CVNN performs better than real-valued neural network (RVNN). In other cases, the complex-valued flexible neural tree (CVFNT) is proposed to solve the binary classification problem.

Because of the unbalanced data, GRN inferred by supervised learning methods may have many false positive regulatory relationships. Because mutual information (MI) is an efficient approach to detect nonlinear dependencies, more methods used MI and conditional mutual information (CMI) to reduce the noised regulations [15]. But, MI method does not work well for continuous multivariate variables. In order to solve this problem, maximum information coefficient (MIC) was developed on the basis of mutual information [8]. MIC could quickly evaluate the relationship of the variables with different functions (linear, exponential and periodic), and search a wide range of relationship types. But when the null hypothesis could not be established, the statistics ability of MIC will be affected. MI and MIC have their strengths and weaknesses when reflecting the relationship between two variables. In order to improve the accuracy of network structure, this paper proposes novel filtering method based on integration of MI and MIC (MIMIC) to eliminate the false-positive regulations inferred by HSL.

## 2. Supervised learning method

### 2.1. Complex-valued flexible neural tree

#### 2.1.1. Structure of CVFNT

Flexible neural tree (FNT) model was first presented in 2005. Because of its automatic features extraction, cross layer connections and flexible activation functions, FNT model performed better than many classical neural networks and has been applied widely for solving forecasting and classification problems. Since CVNN model is more flexible and functional, complex-valued flexible neural tree (CVFNT) model is proposed to solve the classification problem in this paper. CVFNT is an extension of real-valued FNT model. A tree-structural based encoding method with specific instruction set is selected for representing a CVFNT model. In order to create CVFNT models randomly, two operator sets (function set  $F$  and terminal set  $T$ ) are defined in advanced (Equation (1)).

$$\begin{cases} F = \{+_2, +_3, \mathbb{K}, +_n\} \\ T = \{z_1, z_2, \mathbb{K}, z_n\} \end{cases} \quad (1)$$

Where  $+_i$  represents the summation of  $i$  variables.  $z_i$  is the terminal nodes' instruction and takes no operands ( $i = 1, 2, \mathbb{K}, n$ ,  $z_i \in C^n$ ,  $z_i = x_i + jy_i$  and  $j$  stands for the value of  $\sqrt{-1}$ ).

In the process of generating a CVFNT model randomly, the function and terminal operators are selected from instruction sets  $F$  and  $T$ . If a terminal instruction  $z_i$  is selected, the branch is terminated. If a function instruction  $+_n$  is selected,  $n$  leaf nodes and complex-valued weights ( $w_1, w_2, \mathbb{K}, w_n$ ) are randomly created. The output of a non-terminal node can be computed as a complex-valued flexible neuron (CVFN) model.

The output of a CVFN  $+_n$  can be calculated as follows. Equation (2) is the total excitation of  $+_n$ .

$$net_n = w_0 + \sum_{j=1}^n w_j z_j. \quad (2)$$

Where  $w_0$  is threshold value and  $z_j (j = 1, 2, \mathbb{K}, n)$  is the complex-valued input of CVFN operator. The output of the node  $+_n$  is then calculated by complex-valued activation function (CVAF). Generally complex-valued Elliot function is used as CVAF, which is described in Equation (3).

$$out_n = f(a, r, net_n) = \frac{net_n}{a + \frac{1}{r} |net_n|}. \quad (3)$$

Where  $f(\cdot)$  is CVAF and the output is complex-valued.  $a$ ,  $r$  and  $\sigma$  are real variables,  $c$  is complex-valued and  $|net_n|$  is the modulus of complex-valued  $net_n$ . A typical CVFNT model is described in Figure 1. The total output of CVFNT model can be calculated from left to right by depth-first search method.

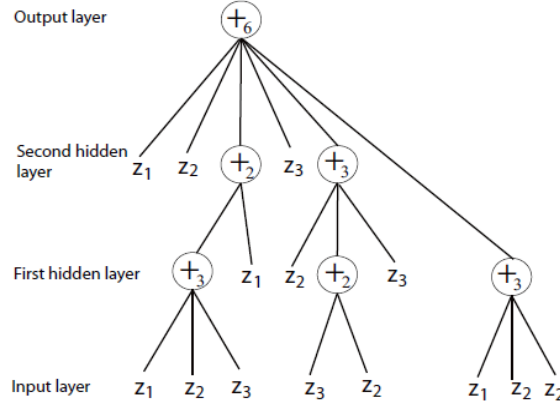


Figure 1. A typical CVFNT model with function set  $F = \{+_2, +_3, +_4, +_5, +_6\}$ , and terminal instruction set  $T = \{z_1, z_2, z_3\}$

### 2.1.2. Structure optimization of CVFNT model

Evolutionary method based on tree structure is proposed to search an optimal or near-optimal CVFNT. In this paper, we use the three neural tree variation operators: select, crossover and mutation, which are described in Ref [11].

### 2.1.3. Parameter optimization of CVFNT model

After optimizing the structure of CVFNT model, the parameters need to be collected. In CVFNT model, weights  $w_i$  and threshold value  $w_0$  are complex-valued. Parameters  $c$  and  $r$  in activation function are real values. In order to gain the optimal parameters of CVFNT model, we need to design an evolutionary method to optimize real-valued and complex-valued parameters simultaneously.

The bat algorithm is a heuristic search method and could be used to search the global optimal solution, which was first presented by Yang in 2010 [13]. Because of its simple structure, few parameters, strong robustness and easy implementation, bat algorithm is proposed to optimize parameters of CVFNT model. Bat algorithm simulates the process that bats use sonar to detect prey and avoid obstacles. Each bat is seen as a solution  $[\text{Re}(w_i), \text{Im}(w_i), \text{Re}(w_0), \text{Im}(w_0), c, r]$  and has its corresponding fitness value. Bat population follows the current optimal bat in the solution space by adjusting frequency, loudness and pulse emission rate. The algorithm is described as follows.

- (1) Initialize random bat positions and create population  $[x_1, x_2, \dots, x_N]$ .
- (2) Evaluate population and give each bat a fitness value. Search the current optimal bat position  $x^*$ .
- (3) If the optimal solution conditions are satisfied or the maximum iterations are achieved, then stop; otherwise go to step (4).
- (4) Update the velocity and position of each bat with pulse frequency using Equations (4), (5), and (6).

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (4)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x^*)f_i \quad (5)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (6)$$

Where  $f_i$  is pulse frequency of  $i-th$  bat, which belongs to  $[f_{\min}, f_{\max}]$ .  $\beta$  is random number from  $[0,1]$ .  $v_i^t$  is the velocity of  $i-th$  bat at time  $t$  and  $x_i^t$  is the position of  $i-th$  bat at time  $t$ .

- (5) Generate a uniformly distributed random number  $r_1$ . If  $r_1 < r_i$  ( $r_i$  is pulse rate of  $i-th$  bat). The stochastic perturbation of the current optimal solution  $x^*$  yields a new solution.
- (6) Generate a uniformly distributed random number  $r_2$ . If  $r_2 < A_i$  &  $f(x_i) < f(x^*)$  ( $A_i$  is pulse loudness of  $i-th$  bat.), accept the new solution population from step (5).
- (7) Update the pulse rate and loudness of each bat using Equations (7) and (8)

$$A_i^{t+1} = \alpha A_i^t \quad x_i^t = x_i^{t-1} + v_i^t \quad (7)$$

$$r_i^{t+1} = r_i^0 [1 - e^{-\gamma t}] \quad (8)$$

Where  $A_i^t$  is pulse loudness of  $i-th$  bat at time  $t$ ,  $r_i^t$  is pulse rate of  $i-th$  bat at time  $t$ ,  $\alpha$  is loudness attenuation coefficient, rate enhancement coefficient and  $r_i^0$  is maximum pulse rate. Go to step (3).

#### 2.1.4. Flowchart of CVFNT optimization

(1) Gene expression data are generally real-valued data, so expression matrix must be transformed to complex-values before optimization process of CVFNT model. The transformation steps are as follows.

- (a) Let gene expression data be  $[g_1, g_2, \dots, g_m]$ , and calculate the maximum and minimum values of expression data ( $g_{\max}$  and  $g_{\min}$ ).
- (b) Expression data  $g_k$  of the  $k$ -th gene is transformed into complex-valued data using Equation (9).

$$\begin{aligned} \varphi_k &= \frac{g_k - g_{\min}}{g_{\max} - g_{\min}}, \\ z_k &= e^{i\varphi_k}, \end{aligned} \quad (9)$$

where  $\delta$  is the shift angle.

(2) Find the optimal CVFNT model, according to the transformed complex-valued gene expression data.

- (a) Initialize the population containing the structure, real-valued and complex-valued parameters of CVFNT.
- (b) Evaluate each individual in the population, using root mean squared error (RMSE).

The output of CVFNT is a complex number, so the output value needs to be transformed into real number using Equation (10).

$$\begin{aligned} \arg z &= \varphi, \\ y &= \frac{\varphi(g_{\max} - g_{\min})}{2\pi - \delta} + g_{\min}. \end{aligned} \quad (10)$$

Where  $\arg z$  is the argument of complex-valued output  $z$ .

If the satisfied solution is found, optimization process is over; otherwise go to (c).

(c) Structure is optimized using three operators: selection, crossover and mutation, which are introduced in Section 2.1.2. At some iterations, select a certain proportion of the individuals to optimize the parameters using bat algorithm, which is introduced in Section 2.1.3.

## 2.2. *K*-Nearest Neighbor

K-Nearest Neighbor (KNN) is one of the simplest machine learning algorithms, which is used to classify by measuring the distance between different feature values. The idea is that if the majority of the most similar  $K$  samples of a sample in the feature space belong to a certain category; this sample also belongs to this category.  $K$  is an odd integer. The process of KNN algorithm is described as follows.

- (1) Calculate the distance between the testing data and the training data. Euclidean distance is used (Equation (11)).

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}. \quad (11)$$

Where  $n$  is the number of sample points,  $x_i$  is training data,  $y_i$  is testing data,  $\bar{x}$  is the averaged value of training data and  $\bar{y}$  is the averaged value of testing data.

- (2) Sort the distances according to ascending order.
- (3) Select the  $K$  minimum distances and get  $K$  training data points.
- (4) Determine the major class over the previous  $K$  points.
- (5) The highest frequency category of  $K$  points is set as the predictive classification category of the testing data.

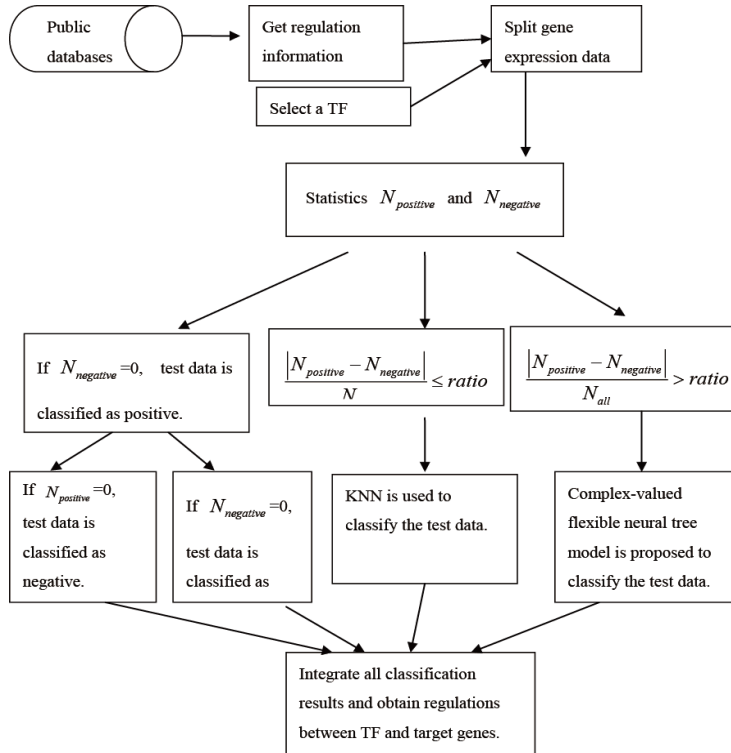


Figure 2. The flowchart of hybrid supervised learning method for each TF

## 2.3. Flowchart of hybrid supervised learning method

The core idea of supervised learning method is to predict whether genes are regulated or not, given TF. For each transcription factor, all genes known to be regulated by this TF form are a set of positive examples. The genes not regulated

are labeled as negative examples. The jackknife method is used for classification. For each TF, according to known regulation information, gene expression data are divided into two classes: positive and negative. Suppose that the number of positive samples is  $N_{positive}$ , the number of negative samples is  $N_{negative}$ , the number of overall samples is  $N_{all}$  and imbalance ratio threshold is *ratio*. For each TF, the flowchart of supervised learning method is depicted in Figure 2 by integrating the regulation relationships between all TFs and target genes for obtaining gene regulatory network.

### 3. Filtering method MIMIC

#### 3.1. Mutual information

For two discrete random variable vectors  $X$  and  $Y$  ( $X \in \chi, Y \in \gamma$ ), the entropy  $I(X)$  and  $I(Y)$  can be computed using Equation (12) and Equation (13) [15].

$$I(X) = -\sum_{x \in \chi} p(x) \log p(x), \quad (12)$$

$$I(Y) = -\sum_{y \in \gamma} p(y) \log p(y), \quad (13)$$

where  $p(x)$  and  $p(y)$  are the marginal probability distribution functions of variables  $X$  and  $Y$ , respectively.

Mutual information (MI) between two variables  $X$  and  $Y$  is defined by Equation (14).

$$I(X, Y) = \sum_{x \in \chi} \sum_{y \in \gamma} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}. \quad (14)$$

Where  $p(x, y)$  is the joint probability density function of  $x$  in  $\chi$  and  $y$  in  $\gamma$ . The Higher the MI value is, the more the dependence of two corresponding genes are.

#### 3.2. Maximum information coefficient

Maximum information coefficient (MIC), as an exploratory analysis tool, can be used to evaluate the relationships between hundreds of variables. Compared with mutual information, MIC can reflect the relationship between two variables better.

A bivariate set is defined as a set where the data elements are ordered tuples  $(a, b)$ . The maximum information gain for all the grids sized of tuple  $(a, b)$  can be computed using Equation (15) [8].

$$I^*(D, a, b) = \max I(D|_G) \quad (15)$$

Where  $G$  is a grid and  $I(D|_G)$  represents the mutual information of  $D|_G$ .

$M(D)$  is the characteristics matrix of  $D$ , which could be represented in Equation (16).

$$M(D)_{a,b} = \frac{I^*(D, a, b)}{\log(\min(a, b))}. \quad (16)$$

$\max(M(D))$  is the MIC of two variables  $a$  and  $b$ . Ideally, if two variables are independent, their MIC value should be about 0.

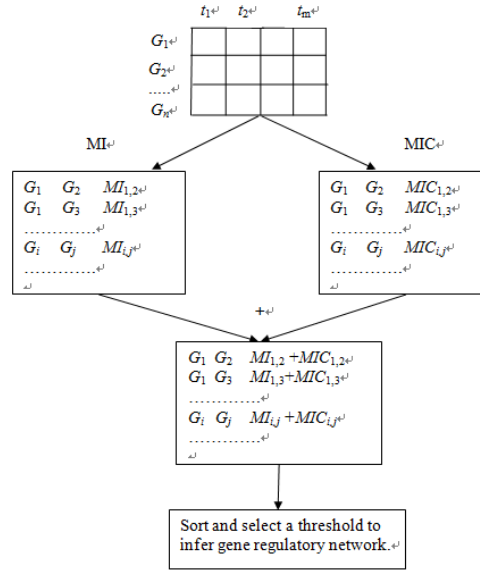


Figure 3. The flowchart of MIMIC

### 3.3. The flowchart of MIMIC

The flowchart of MIMIC is described in Figure 3. The gene expression data contains  $n$  genes ( $G_1, G_2, \dots, G_n$ ) and  $m$  sample points ( $t_1, t_2, \dots, t_m$ ).  $MI_{i,j}$  represents mutual information between gene  $G_i$  and gene  $G_j$ .  $MIC_{i,j}$  represents maximum information coefficient between gene  $G_i$  and gene  $G_j$ .

## 4. Experiments

The inference gene regulatory network is evaluated by five criterions, i.e. true positive rate (TPR), false positive rate (FPR), positive predictive value (PPV), accuracy (ACC) and F-score, which are calculated using Equation (17).

$$\begin{aligned}
 TPR &= \frac{TP}{TP + FN}, \\
 FPR &= \frac{FP}{FP + TN}, \\
 PPV &= \frac{TP}{TP + FP}, \\
 ACC &= \frac{TP + TN}{TP + FP + TN + FN}, \\
 F - score &= \frac{2PPV \times TPR}{PPV + TPR},
 \end{aligned} \tag{17}$$

where TP, FP, TN and FN are presented in Figure 4.

		True output	
		Positive	Negative
Predicting output	Positive	TP	FP
	Negative	PN	TN

Figure 4. Description of TP, FP, PN and TN

#### 4.1. Select the threshold value

In Figure 2, *ratio* is the imbalance ratio threshold. According to *ratio*, HSL selects different classifiers to infer GRN. Threshold *ratio* is critical to the performance of the algorithm. In order to determine the threshold value, we test the performance of our method using different ratios and evaluate its effect on TPR, FPR, PPV, ACC and runtime, in order to select the proper value. The values of ratio are in range [0.03, 0.5] at intervals of 0.03. The test data are collected from *E.coli* network, which includes 150 genes and 202 true regulatory relationships [7]. The results are listed in Table 1.

Table 1. Comparison of HSL with different ratios on *E.coli* network

Ratio	TPR	FPR	PPV	ACC	Runtime (s)	F-score
0.03	0.519802	0.025715	0.152838	0.970265	7350.041	0.23622
0.06	0.514851	0.016328	0.221277	0.979485	3943.289	0.309524
0.09	0.509901	0.012186	0.274667	0.983528	2616.702	0.357019
0.12	0.539604	0.011917	0.290667	0.984061	2889.461	0.377817
0.15	0.475248	0.012455	0.256684	0.98295	2781.075	0.333333
0.18	0.465545	0.008792	0.330882	0.987887	1577.461	0.386829
0.21	0.435644	0.007835	0.335878	0.987149	1419.653	0.379311
0.24	0.430693	0.008371	0.318681	0.986575	1461.866	0.366316
0.27	0.450495	0.008013	0.33829	0.987108	1354.662	0.386412
0.3	0.440594	0.008058	0.33209	0.986974	1252.089	0.378724
0.33	0.435644	0.008014	0.330827	0.986972	1171.586	0.376068
0.36	0.430693	0.008282	0.321033	0.986663	1178.393	0.367865
0.39	0.445545	0.008058	0.334572	0.987019	1180.987	0.382165
0.42	0.440594	0.007879	0.337121	0.98715	1362.712	0.381974
0.45	0.425743	0.008104	0.323308	0.986794	1319.71	0.367521
0.48	0.410891	0.00806	0.316794	0.986702	1175.395	0.357759

From Table 1, we can see that with the increasing of ratio, TPR gradually decreases, which means that the correct regulation relationship predicted is less and less. This is because, when the ratio is small, more regulation relationships of the target gene are predicted by neural network, and the accuracy is higher, but the run time is longer. When the ratio is larger, the regulation relationships of the target genes are determined by the KNN method, and the accuracy is poor, but the running time is faster. Based on the above analysis, in order to predict more regulation relationships with the shorter time, we select a compromise value 0.18 as the ratio in the following experiment.

#### 4.2. Performance of filtering method

In order to test the performance of MIMIC, we use the dataset from *Silico* network in DREAM5, which contains 197 regulatory factors and 4012 regulations [7]. The different filtering methods are used to delete the redundant regulations. The results are listed in Table 2. From the results, it can be seen that filtering methods could remove a large number of false positive regulatory relationships, but may also delete a small portion of true regulations. By comparing the performances of three filtering methods, it can be found that MIMIC performs better than MI and MIC.

Table 2. Comparison of HSL with different filtering methods on *Silico* network

	TPR	FPR	PPV	ACC	F-score
HSL	<b>0.39781</b>	0.029703	0.266	0.95521	0.318817
HSL+MI	0.38036	0.023343	0.305996	0.96095	0.33915
HSL+MIC	0.3674	0.01973	0.335	0.96412	0.35045
HSL+MIMIC	0.38185	<b>0.00906</b>	<b>0.34818</b>	<b>0.96488</b>	<b>0.36424</b>

#### 4.3. Performance of our method

In this part, the expression data generated from *E.coli* network are used to test our method. *E.coli* network is from a well-know dataset DREAM5, which contains 334 transcription factors (TFs), 4511 target genes, 2066 true regulations and 805 chips [7].

Table 3. Performance comparison of several methods on regulatory network of *E.coli*

	TPR	FPR	PPV	ACC	F-score
PCA-CMI	0.3819	0.3238	0.002	0.6762	0.003979
NARROMI	0.278	0.0007	0.065	0.9987	0.105364
LSGPA	0.484	3.62E-04	0.0863	0.9996	0.146482
CLR	0.37222	0.017301	0.001517	0.98266	0.003022
LNN	0.4342	4.03E-04	0.0707	0.9995	0.121652
HSL	<b>0.50242</b>	0.000265	0.118075	0.9997	0.191213
HSL+MIMIC	0.48889	<b>9.51E-05</b>	<b>0.26642</b>	<b>0.9999</b>	<b>0.344886</b>



Through ten runs, Table 3 shows the comparison results of several methods. It could be seen that supervised learning methods (LNN and our method) perform better than unsupervised learning methods (context likelihood of relatedness (CLR) algorithm [2], path consistency algorithm based on conditional mutual information (PCA-CMI) [15], NRROMI [14] and LSGPA [10]). NARROMI combined ODE-based recursive optimization and information theory-based MI to infer GRN. LSGPA combined splitting technology and ODE to improve the performance of large-scale GRN inference. By analysis of reconstructed gene regulatory network, it is found that the regulation relationships of the 40.94% target genes could all be identified.

## 5. Discussions and conclusions

In this paper, a new hybrid supervised learning methods (HSL) is proposed to infer gene regulatory network. In HSL, according to the level of data imbalance, three classifiers could be chosen. After inferring GRN with supervised learning methods, a novel filtering method based on integration of mutual information and maximum information coefficient is proposed to reduce false-positive regulations.

*Silico* network in DREAM5 is used to test the performance of filtering methods. HSL has higher TPR than HSL with filtering methods. HSL with filtering methods has higher F-score than HSL method. It is because that filtering methods could remove a large number of false positive regulatory relationships, but may delete a small portion of true relations. Through comparison performance of three filtering methods, the filtering effect of MIMIC is better than ones of MI and MIC. Gene regulatory network of *E.coli* in DREAM5 is proposed to test the performance of our approach. The experiment results show that our proposed approach performs better than state-of-the-art unsupervised learning methods (CLR, PCA-CMI, NARROMI and LSGPA) and supervised learning methods (LNN).

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61702445), Shandong Provincial Natural Science Foundation, China (No. ZR2015PF007).

## References

1. L. Cerulo, C. Elkan, and M. Ceccarelli, "Learning Gene Regulatory Networks from Only Positive and Unlabeled Data," *BMC Bioinformatics*, vol. 11, no. 1, pp. 228, 2010
2. J. Faith, B. Hayete, J.T. Thaden, I. Mogno, J. Wierzbowski, 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. e8, pp. 0054-0066, 2007
3. Z. Gillani, M.S. Akash, M.D. Rahaman, and M. Chen, "CompareSVM: Supervised, Support Vector Machine (SVM) Inference of Gene Regularity Networks," *BMC Bioinformatics*, vol. 15, pp. 395, 2014
4. T. Kitajima, T. Yasuno, and N. Ikeda, "Wind Speed Prediction System Using Complex-valued Neural Network and Frequency Component of Wind Speed," *Ice Technical Report Neurocomputing*, vol. 113, pp. 35-40, 2013
5. F. Liu, S.W. Zhang, W.F. Guo, Z.G. Wei, and L. Chen, "Inference of Gene Regulatory Network based on Local Bayesian Networks," *Plos Computational Biology*, vol. 12, no. 8, pp. e1005024, 2016
6. S.R. Maetschke, P.B. Madhamshettiwar, M.J. Davis, and M.A. Ragan, "Supervised, Semi-supervised and Unsupervised Inference of Gene Regulatory Networks," *Brief Bioinform*, vol. 15, no. 2, pp.195-211, 2014
7. D. Marbach, J. Costello, R. Kuffner, N. Vega, R. Prill, D. Camacho, K. Allison, the DREAM5 Consortium, M. Kellis, J. Collins, and G. Stolovitzky, "Wisdom of Crowds for Robust Gene Network Inference," *Nat Methods*, vol. 9, no. 8, pp.796-804, 2012
8. D.N. Reshef, Y.A. Reshef, H.K. Finucane, S.R. Grossman, G. McVean, P. Turnbaugh, E. Lander, M. Mitzenmacher, and P. Sabeti, "Detecting Novel Associations in Large Data Sets," *Science*, vol. 334, pp. 1518-1524, 2011
9. O. Satoshi, N. Sarah, Z. Sascha, J.C. Schwamborn, and D.S. Antonio, "A Generalized Gene-Regulatory Network Model of Stem Cell Differentiation for Predicting Lineage Specifiers," *Stem Cell Reports*, vol. 7, no. 3, pp. 307-315, 2016
10. X. Xiao, W. Zhang, and X. Zou, "A New Asynchronous Parallel Algorithm for Inferring Large-Scale Gene Regulatory Networks," *PLoS ONE*, vol. 10, no. 3, pp. e0119294, 2015
11. B. Yang, Y.H. Chen, and M.Y. Jiang, "Reverse Engineering of Gene Regulatory Networks using Flexible Neural Tree Models," *Neurocomputing*, vol. 99, pp. 458-466, 2013
12. B. Yang, M.Y. Jiang, and Y.H. Chen, "A Novel Hybrid Framework for Reconstructing Gene Regulatory Networks," *International Journal of Hybrid Information Technology*, vol. 6, no. 5, pp. 255-268, 2013
13. X.S. Yang and X.S. He, "Bat Algorithm: Literature Review and Applications," *International Journal of Bio-Inspired Computation*, vol. 5, no. 3, pp. 141-149, 2013.
14. 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, pp. 106-113, 2013.
15. X. Zhang, X.M. Zhao, K. He, L. Lu, Y. Cao, J.D. Liu, and L. Chen, "Inferring Gene Regulatory Networks from Gene Expression Data by Path Consistency Algorithm based on Conditional Mutual Information," *Bioinformatics*, vol. 28, pp. 98-104, 2012.

**Bin Yang** is the teacher of Zaozhuang University. His research interests include biology network inference, deep learning.

**Wei Zhang** is the teacher of Zaozhuang University. His research interests include network security, data mining.

**Jiaguo Lv** is the teacher of Zaozhuang University. His research interests include social network inference, data mining.