# Gene Regulatory Relationship Mining Using Improved Three-Phase Dependency Analysis Approach

Jianxiao Liu, Zonglin Tian, Yingjie Xiao, Haijun Liu, Songlin Hao, Xiaolong Zhang, Chaoyang Wang, Jianchao Sun, Huan Yu and Jianbing Yan

**Abstract**—How to mine the gene regulatory relationship and construct gene regulatory network (GRN) is of utmost interest within the whole biological community, however, which has been consistently a challenging problem since the tremendous complexity in cellular systems. In present work, we construct gene regulatory network using an improved three-phase dependency analysis algorithm (*TPDA*) Bayesian network learning method, which includes the steps of *Drafting*, *Thickening* and *Thinning*. In order to solve the problem of learning result is not reliable due to the high order conditional independence test, we use the entropy estimation approach of Gaussian kernel probability density estimator to calculate the (conditional) mutual information between genes. The experiment on the public benchmark data sets show the improved method outperforms other 9 kinds of Bayesian network learning methods when to process the data with large sample size, with small number of discrete values, and the frequency of different discrete values is about same. In addition, the improved *TPDA* method was further applied on a real large gene expression data set on RNA-seq from a global collection with 368 elite maize inbred lines. Experiment results show it performs better than the original *TPDA* method and other 9 kinds of Bayesian network learning algorithms significantly.

**Index Terms**—Gene regulatory, Bayesian network, Mutual information, Maize

## 1 Introduction

Gene regulatory networks (GRN) could be inferred from expression profiles and interactions between regulatory targets. It can help to explain the new function of gene and the mechanism of the life process, and provide important information for drug design or medical-related fields.

At present, there exists a lot of research work about inferring gene regulatory network from gene expression data [1], [2]. Several methods are introduced into the research of gene regulatory network construction, such as differential equation method, boolean model, regression method [3], linear programming [4], algebra-based method [5], distance correlation, motif activity response analysis [6], majority rule method, etc. The above approaches have certain advantages and limitations respectively. For example, the method of boolean model is simple and straightforward, but it can lead to information loss, thus to affect the accuracy of gene network construction. Differential equation method can better mine the continuous dynamic relationships between genes, but it cannot deal with the noise of the experiment data [1]. The information-theoretic approaches are increasingly used for constructing GRNs, such as *ARACNE* [7], *PCA-CMI* [8], *CMI2NI* [9], etc. However, the information-theoretic based method cannot determine the direction of regulatory relationship, and it cannot calculate the regulatory relationship among multiple genes. Recently, machine learning methods are more and more used to mine gene regulatory relationships [10].

At present, Bayesian network is more and more used in the research of gene regulatory relationship mining. The commonly used Bayesian network structure learning method mainly includes the dependence analysis based method and search scoring based method. The related research work mainly includes Greedy Search (*GS*) [11], [12], Markov Chain Monte Carlo (*MCMC*) [13], Hill Climbing (*HC*) [14], *K2* [15], etc. In addition, the dynamic Bayesian network is also used to analyze the temporal gene expression data [2], [16]. Recently, the prior knowledge is used to improve the learning efficiency and accuracy of Bayesian network method, such as using the cocitation in PubMed and schematic similarity in Gene Ontology [13]. In all, the existing research work mainly uses the search scoring based Bayesian network learning method. This method often uses the local or random search strategy, and it is a combinatorial explosion problem with the increase of the node number, resulting in a long time to build the network.

Compared with the search scoring based Bayesian network learning method, the learning efficiency of dependence analysis based method is relatively high, and it can obtain the global optimal solution. The three-phase dependency analysis algorithm (*TPDA*) is a commonly used dependence analysis based Bayesian network structure learning method [17]. This algorithm uses the global search strategy, and it can quickly determine the relationship between nodes by computing the mutual informa-

- *Jianxiao Liu, Zonglin Tian, Songlin Hao, Xiaolong Zhang, Chaoyang Wang, Jianchao Sun and Huan Yu are with the College of Informatics, Huazhong Agricultural University, Wuhan, China, 430070. E-mail: liujianxiao321@163.com.*
- *Yingjie Xiao, Haijun Liu and Jianbing Yan are with the National Key Laboratory of Crop Genetic Improvement, Huazhong Agricultural University, Wuhan, China, 430070. E-mail: jianbingyan@163.com.*

*Please note that all acknowledgments should be placed at the end of the paper, before the bibliography (**note that corresponding authorship is not noted in affiliation box, but in acknowledgment section**).*

xxxx-xxxx/0x/$xx.00 © 2018 IEEE Published by the IEEE Computer Society

tion or conditional mutual information. Its learning efficiency is higher than the search scoring based methods. Therefore, we use the three-phase dependency analysis Bayesian network learning method to construct the gene regulatory network in this work. However, this method uses the conditional mutual information and open path judgment to do the conditional independence test, the learning result is often not reliable due to the high order conditional independence test. At the same time, in the view of the transcriptome data is often subjected to the normal distribution [8], [9], we use the entropy estimation approach of Gaussian kernel probability density estimator to calculate the mutual information and conditional mutual information between genes in the *TPDA* method. It can effectively solve the problem of result is unreliable caused by high order conditional independence test. For the discrete values, the experiment on the public benchmark data sets show our improved *TPDA* method performs best when to process the data with large sample size, with small number of discrete values, and the frequency of different discrete values is about same. In addition, experiments on real maize data set show the improved *TPDA* method outperforms other 9 kinds of Bayesian network learning algorithms. We develop the source code about the improved *TPDA* method using *R*.

# 2 Methods

## 2.1 Bayesian Network

Bayesian probability denotes the confidence of an event occurrence. It is the foundation of Bayesian network, which is called as belief network. A complete Bayesian network includes three parts: nodes, edges between nodes and the conditional probability of all the nodes.

The three-phase dependency analysis algorithm (*TPDA*) is a commonly used dependence analysis based Bayesian network structure learning method. The concrete process of *TPDA* method mainly includes three steps: *Drafting*, *Thickening* and *Thinning* [17].

## 2.2 Gene Regulatory Relationship Mining

**Algorithm 1**. Gene regulatory relationship mining

Input: $G=\{ge_i, 1≤i≤gnum\}$, $M=\{m_j, 1≤j≤mnum\}$, $GM=\{gm_{ij}, 1≤i≤gnum, 1≤j≤mnum\}$, *NOL*

Output: *graph*

```
1: $i, j, v_{ij}, cv_{ij}$←0, $S, R, Cutset, Cutset_{min}$←∅
2: Node[] nodes←new Node [gnum]
3: graph←new Graph(nodes, gnum)
4: for $i$=1 to gnum do
5:   graph.nodes[$i$]←$ge_i$
6: end for
7: for $i$=1 to gnum do
8:   for $i$=1 to gnum do
9:     $v_{ij}$←$MI(gm_i, gm_i)$
10:    if($v_{ij}$ >ε) then
11:      $S$←$S$∪<$ge_i$, $ge_j$, $v_{ij}$>
12:    end if
13:  end for
14:$S$←Sort($S$)//Sort $S$ according to $MI(gm_i, gm_j)$
15:for all <$ge_i$, $ge_j$, $MI(gm_i, gm_j)$> in $S$ do
16:  if(ExistsPath($ge_i$, $ge_j$)) then
17:    $R$←$R$∪<$ge_i$, $ge_j$>
18:  else graph.insert(new Edge($ge_i$, $ge_j$)) with NOL
19:end for
20: for all <$ge_i$, $ge_j$> in $R$ do
21:   Cutset←FindCutSet(graph, $ge_i$, $ge_j$)
22:   $v_{ij}$=CMI($ge_i$, $ge_j$ | Cutset)
23:     if($v_{ij}$>ε) then
24:       graph.insert(new Edge($ge_i$, $ge_j$)) with NOL
25:     end if
26:   Cutset←∅
27: end for
28: for all Edge($ge_i$, $ge_j$) in graph do
29:   delete Edge($ge_i$, $ge_j$)
30:   $Cutset_{min}$←FindMinCutSet(graph, $ge_i$, $ge_j$)
31:   $cv_{ij}$=CMI($ge_i$, $ge_j$ | $Cutset_{min}$)
32:     if($cv_{ij}$>ε) then
33:       graph.insert(new Edge($ge_i$, $ge_j$)) with NOL
34:     end if
35:  $Cutset_{min}$←∅
36: end for
37: return graph
```

In the *Input*, *G* represents the gene set, *M* represent the sample set, *GM* represents the expression data of all the genes in *G*, *NOL* represents the order of all the nodes. In the algorithm, step 7-19 is used to construct the initial Bayesian network (*Drafting*), step 20-27 is used to judge the conditional independence (*Thickening*), and step 28-36 is used to do the network optimization (*Thinning*).

(1) Initial Bayesian network construction (*Drafting*)

The mutual information $MI(ge_i, ge_j)$ between any two genes $ge_i$ and $ge_j$ will be calculated firstly. The edges whose mutual information is larger than the threshold will be inserted into an edge set named *S*. Then we sort all the node pairs in *S* according to the value of mutual information. All the node pairs in *S* are judged whether there exists an open path between the corresponding nodes or not. If there exists an open path, the node pair will be inserted into another edge set named *R*. Otherwise, we will insert the corresponding edge into the graph, thus to construct the initial network. The direction of the edge is determined by the node order in *NOL*.

For the discrete variables in *X* and *Y*, we can define *MI* in term of probability and entropies, as shown in Eq.(1).

$$\begin{aligned} MI(X,Y) &= H(X)+H(Y)-H(X,Y) \\ &= -\sum_{x\in X,y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \end{aligned} \tag{1}$$

As has been described above, we use the entropy estimation approach of Gaussian kernel probability density estimator to calculate the mutual information between genes to solve the above problem [8], [9]. We can get the entropy of variable X, as shown in Eq.(2).

$$H(X)=\log[(2\pi e)^{n/2}|C|^{1/2}]=\frac{1}{2}\log(2\pi e)^n|C| \tag{2}$$

For gene $ge_i$ and $ge_j$, we use Eq.(3) to calculate the mutual information *MI*($ge_i$, $ge_j$).

$$MI(ge_i, ge_j) = \frac{1}{2}\log\frac{|C(ge_i)|\cdot|C(ge_j)|}{|C(ge_i, ge_j)|} \tag{3}$$

(2) Conditional independence judgement (*Thickening*)

On the basis of constructing the initial network through *Drafting*, we judge the node pair of *R* in turn using the conditional independence test. From the aspects of non-transfer connection, serial (transfer) connection and convergence connection, we get the minimum cut set *cutset* which can *D*-separate the node pair of *R* in turn.

It calculates the conditional mutual information (*CMI*) between the node pair in *R* and the corresponding minimum cut set, and thus to judge whether the node pair is conditional independent or not. The *CMI* of variables *X* and *Y* given *Z* is defined using Eq.(4).

$$CMI(X,Y|Z) == H(X,Z) + H(Y,Z) - H(Z) - H(X,Y,Z) \tag{4}$$

Similarly, we can use *CMI* to measure the conditional dependence between two variables (genes) given other variables (genes), as shown in Eq.(5).

$$CMI(ge_i, ge_j \mid cutset) = \frac{1}{2}\log\frac{|C(ge_i, cutset)|\cdot|C(ge_j, cutset)|}{|C(cutset)|\cdot|C(ge_i, ge_j, cutset)|} \tag{5}$$

In Eq.(5), $ge_i$ and $ge_j$ represent two genes, *cutset* represents the minimum cut set, *C* represents the covariance matrix of gene expression, |*C*| represents the determinant of matrix *C*.

(3) Network optimization (*Thinning*)

This stage will check each edge *e* in graph to achieve the further optimization in the network. Supposing two genes of *e* is ($ge_i$, $ge_j$), if there exists an open path which connects $ge_i$ and $ge_j$ except for *e*, we remove *e* temporarily and find the minimum cut set that can *D*-separate $ge_i$ and $ge_j$. Then we use Eq.(5) to judge whether the node pair is conditional independent or not. If it is independent, then we delete *e*.

## 3 Results

The *bnlearn* is an *R* package of learning the structure of Bayesian networks, estimating the parameters and performing Bayesian inference. This package does not include the three-stage dependency analysis algorithm. We use *R* to develop the source code of the improved *TPDA* algorithm in this work. We use other 9 kinds of Bayesian network learning methods (including *gs*, *hc*, *iamb*, *mmpc*, *rsmax*, *tabu*, *fastiamb*, *interiamb*, *mmhc*) to construct gene regulatory network, and thus to compare these methods with our improved *TPDA* method. The experiment is carried out on the computer with the configuration of dual Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz, and 128G memory.

### 3.1 Learning effect comparison of different methods

We use the benchmark data set to validate the Bayesian network learning methods, including *Alarm*, *Child*, *Insurance*, *Barley*, *Mildew*, *etc*(http://www.dsl-lab.org/supplements/mmhc_paper/mmhc_index.html) [18]. The *Alarm*, *Child*, *Insurance* and *Hailfinder* contain the data sets of *Type*-1, *Type*-3 and *Type*-5. Each data set contains 10 versions of data at each sample size (500, 1000, 5000), such as *Alarm*-1-500v1~*Alarm*-1-500v10. We use Eq.(6) to calculate the learning precision, $Num_{total}$ denotes the total number of edges in the Bayesian network that has been learned. $Num_{match}$ denotes the common number of edges between the learning Bayesian network and the standard network in the benchmark data set.

$$\Pr ecision = \frac{Num_{match}}{Num_{total}} \tag{6}$$

In the case of setting the threshold in the *Drafting* stage to 0.06, and setting the threshold of *Thickening* and *Thinning* to 0.04, we use 10 kinds of Bayesian network learning methods (*gs*, *hc*, *iamb*, *mmpc*, *rsmax*, *tabu*, *fastiamb*, *interiamb*, *mmhc* and *TPDA*) to construct the gene regulatory network of the above benchmark data sets. Fig. 1-11 show the learning precision comparison of 10 kinds of Bayesian network learning methods about all the benchmark data sets, including *Alarm*-1, *Alarm*-3, *Alarm*-5, *Child*-1, *Child*-3, *Child*-5, *Insurance*-1, *Insurance*-3, *Insurance*-5, *Barley* and *Mildew*. In order to ensure the learning precision of the experiment comparison, we take the average precision of 10 versions of each data set to do the comparison and analysis.

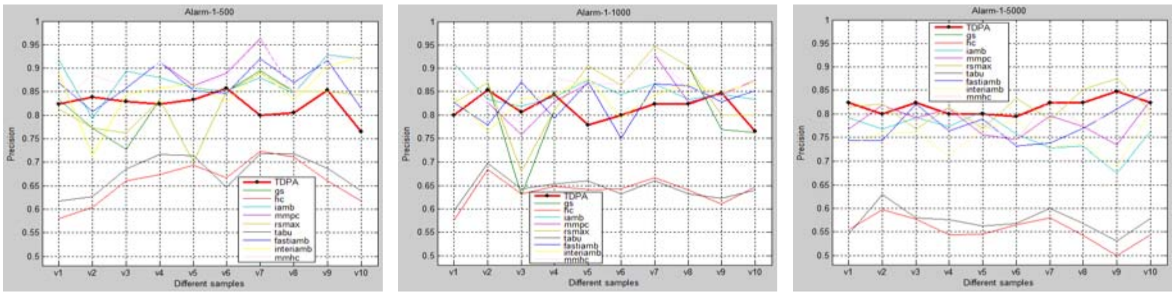

Fig. 1. Learning precision comparison of different methods (Alarm-1)

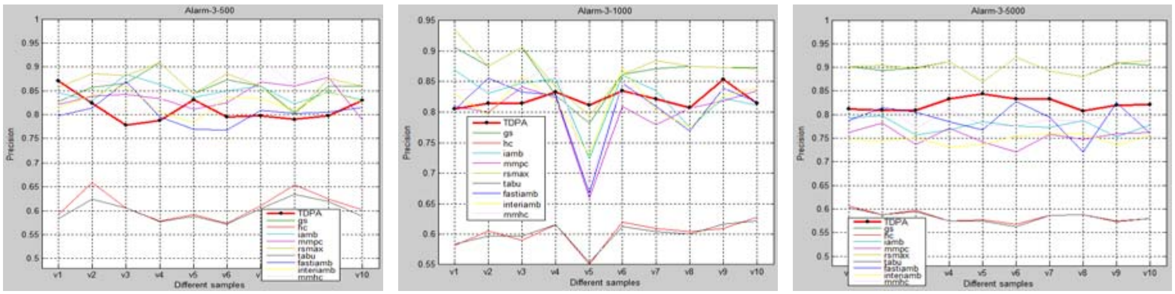

Fig. 2. Learning precision comparison of different methods (Alarm-3)

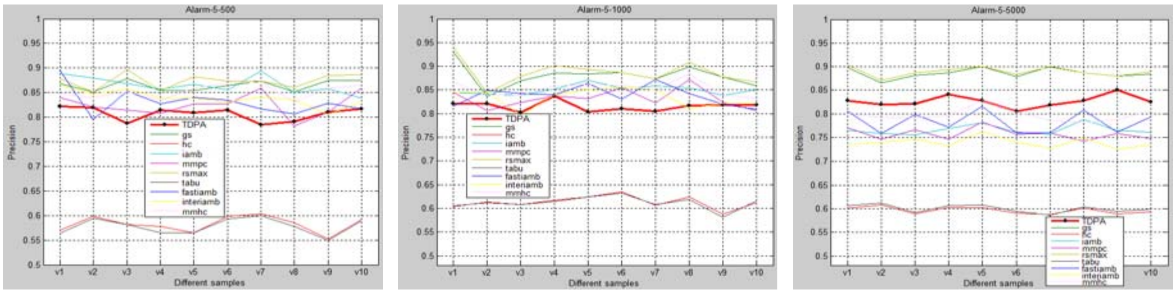

Fig. 3. Learning precision comparison of different methods (Alarm-5)

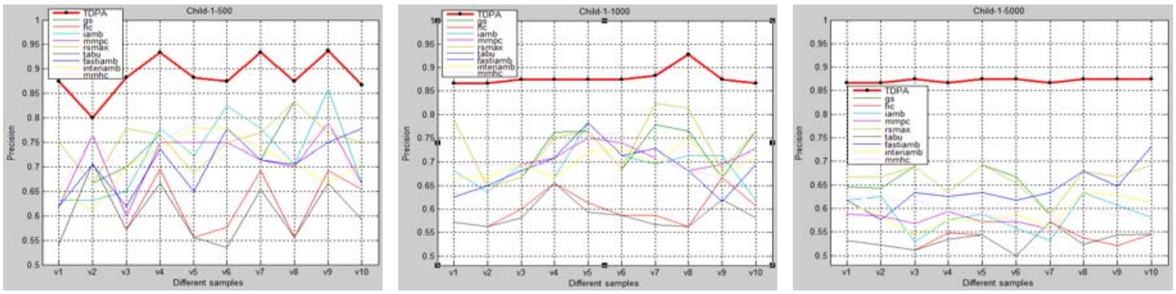

Fig. 4. Learning precision comparison of different methods (Child-1)

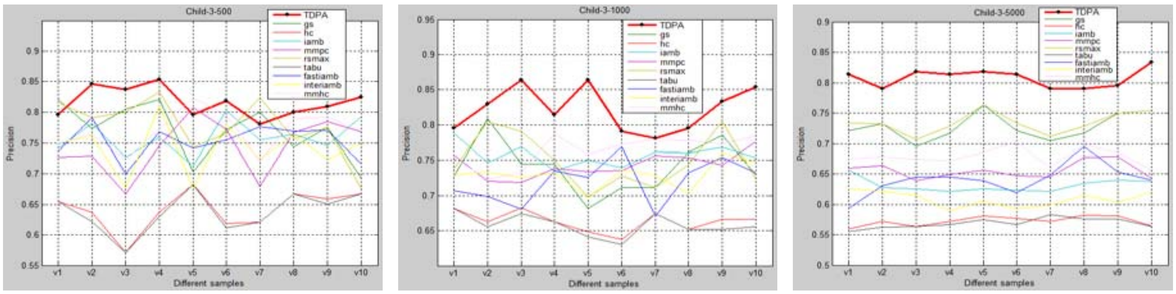

Fig. 5. Learning precision comparison of different methods (Child-3)

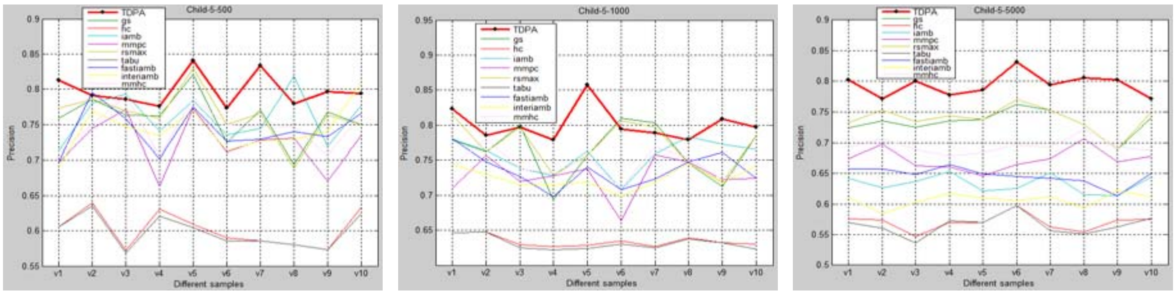

Fig. 6. Learning precision comparison of different methods (Child-5)

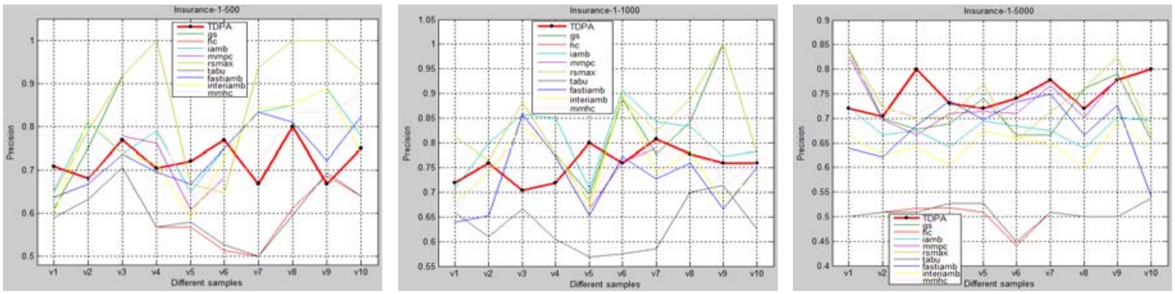

Fig. 7. Learning precision comparison of different methods (Insurance-1)

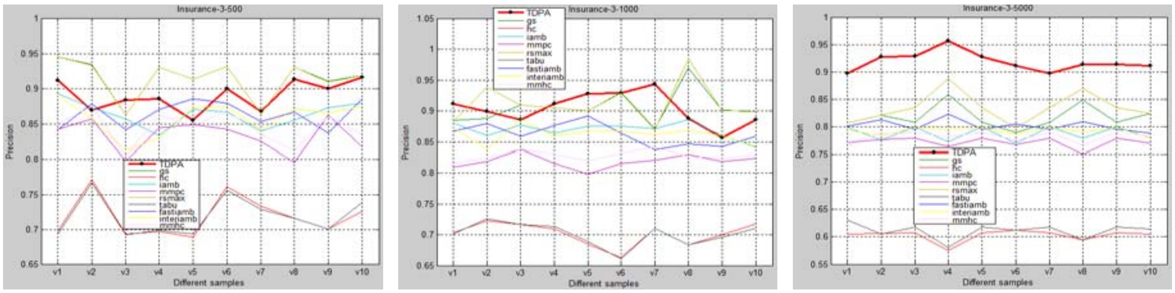

Fig. 8. Learning precision comparison of different methods (Insurance-3)

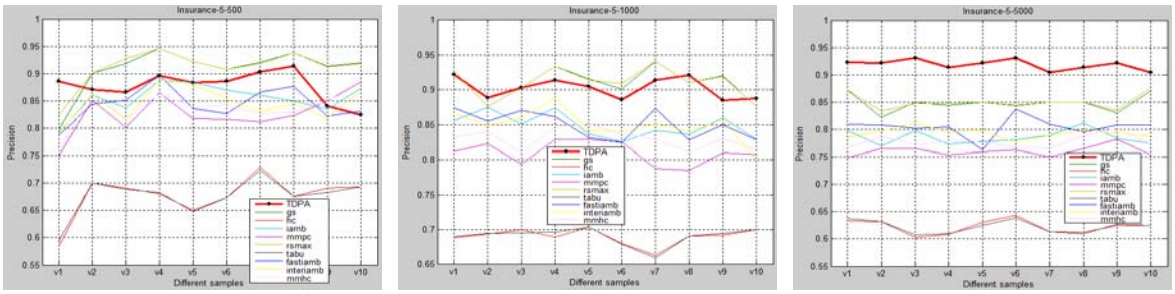

Fig. 9. Learning precision comparison of different methods (Insurance-5)

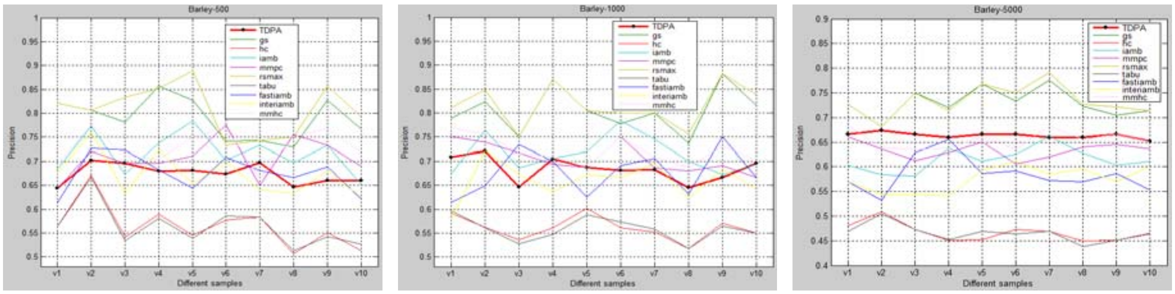

Fig. 10. Learning precision comparison of different methods (Barley)

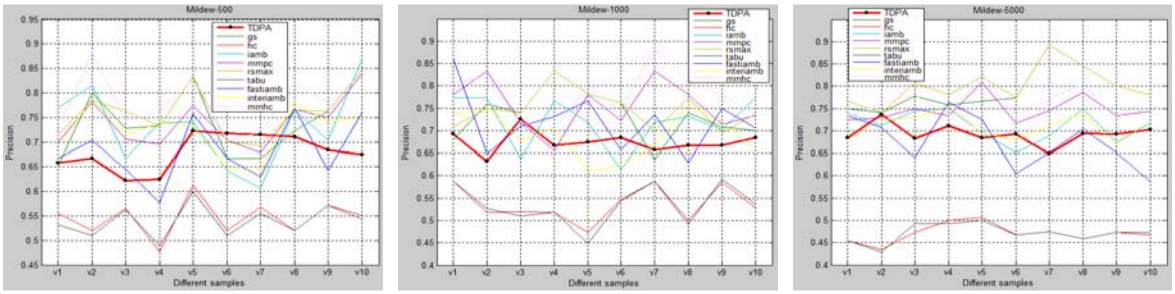

Fig. 11. Learning precision comparison of different methods (Mildew)

Through Fig. 1-11, we can see the learning precision of different methods is different for the particular data set. We can see the learning precision of *hc* and *tabu* is lower than the other 8 kinds of methods apparently. When the sample number of all the data sets is taken 5000, we can see the learning precision of our *TPDA* method is better than the sample is taken 500 and 1000. It means our *TPDA* method is suitable for processing the data with large sample size. For the data sets of *Alarm-1*, *Alarm-3* and *Alarm-5*, the learning effect of our improved *TPDA* method is all most the same as other methods when the data sample size is 500 and 1000. But when the sample is 5000, the learning precision of our improved *TPDA* method is enhanced obviously.

In addition, there are large differences of learning precision about the specific method for different data sets. For the data sets of *Child-1*, *Child-3*, *Child-5*, *Insurance-1*, *Insurance-3* and *Insurance-5*, when the sample is taken 5000 especially, the learning precision of the *TPDA* method is much higher than other methods basically. In Fig. 10 and Fig. 11, we can also see the learning precision of *TPDA* on the data sets of *Barley* and *Mildew* is far less than the data sets of *Alarm*, *Child* and *Insurance*. Compared with other 9 kinds of methods, the learning effect of *TPDA* on the data sets of *Barley* and *Mildew* has no obvious advantages.

Attempting to analyze the impact of different data sets for our improved *TPDA* method, we analyze the distribution of the data sets of *Alarm-3*, *Child-3*, *Insurance-3*, *Barley* and *Mildew*. We randomly select the v3 version of these data sets to do comparison and analysis, the result is shown in Fig. 12-14.

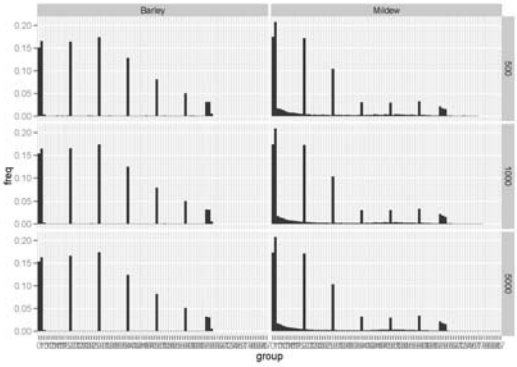

Fig. 12. Data distribution of Barley_Mildew_v3

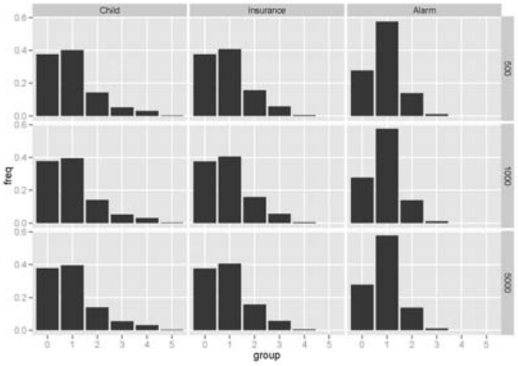

Fig. 13. Data distribution of Child-3_Insurance-3_Alarm-3_v3

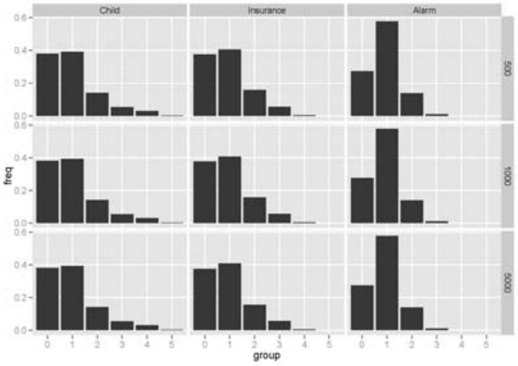

Fig. 14. Data distribution of Child-5_Insurance-5_Alarm-5_v3

Through Fig. 12-14 we can see the distribution of different data sets is largely different. In Fig. 12, the distribution of the data sets of *Barley* and *Mildew* is consistent. The number of discrete value in the two data sets is very large, as many as 80 kinds, and the frequency of different discrete values is largely different. In Fig. 13-14, compared with the data sets of *Barley* and *Mildew*, the distribution of the *Alarm*, *Child* and *Insurance* is relatively similar. The number of discrete values in these data sets is less. In addition, we can see the classification frequency of the two main discrete values is higher in the data sets of *Child* and *Insurance*, and their frequency is almost identical. But the classification frequency of the main discrete values in the data set of *Alarm* is largely different. In this data set, the frequency of one discrete value is significantly higher than the others.

In combination with the results in Fig. 1-11, the learning effect of our improved *TPDA* method is not good for the data sets of *Barley* and *Mildew*, and its learning effect is best for the data sets of *Child* and *Insurance*. For the data set of *Alarm*, the learning effect of *TPDA* method becomes better when the sample size becomes more and more (500-5000). It is concluded that our improved *TPDA* method is suitable for processing the data with large sample size, with small number of discrete values (such as two values), and with the frequency of different discrete values is about same. It is not suitable for processing the data with more discrete values and with large frequency difference of discrete values

### 3.2 Learning effect comparison of improved TPDA

In this experiment, we take the dataset of *Insurance* as example to do comparison and analysis. Fig. 15-17 show the learning precision comparison of the two methods about the data sets of *Insurance*-1, *Insurance*-3 and *Insurance*-5. In order to ensure the learning precision comparison, we take the average precision of 10 versions of each data set to do the comparison and analysis. In Fig. 15-17, T1~T10 in *X* axis refers to different thresholds set in the three stages respectively, as shown in the following: (0.04,0.055), (0.04,0.06), (0.045,0.055), (0.05,0.055), (0.05,0.06), (0.055,0.055), (0.06,0.055), (0.065,0.055), (0.065,0.06), (0.065,0.065). In the form of (*X*, *Y*), *X* refers to the threshold set in the stage of *Drafting*. *Y* refers to the threshold set in the stages of *Thickening* and *Thinning*.

Through Fig. 15-17, we can see the learning precision of the two methods is largely different for different data sets. And the learning precision of them is different for the same data set. In all, the learning effect of using the entropy estimation approach of Gaussian kernel probability density estimator is better than without using it except for the data set of *Insurance*-1-5000. In addition, we can see the learning precision gap between the two methods is relatively large for the data set of *Insurance*-3 and *Insurance*-5. For the data set of *Insurance*-1, there is little difference in the gap of the two methods.

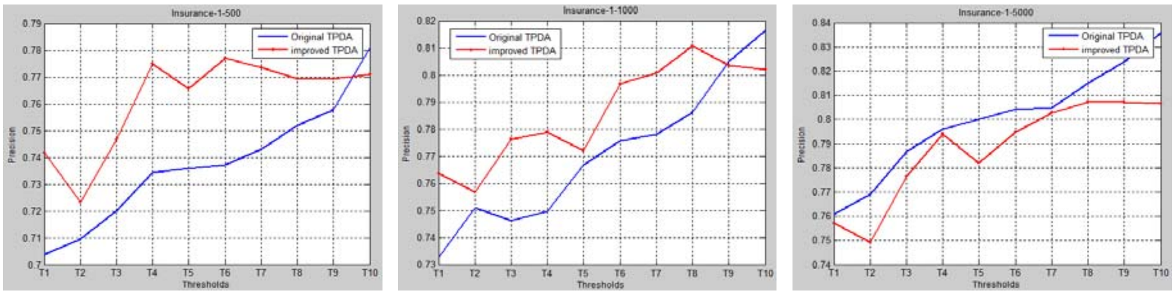

Fig. 15. Learning precision comparison of improved *TPDA* (Insurance-1)

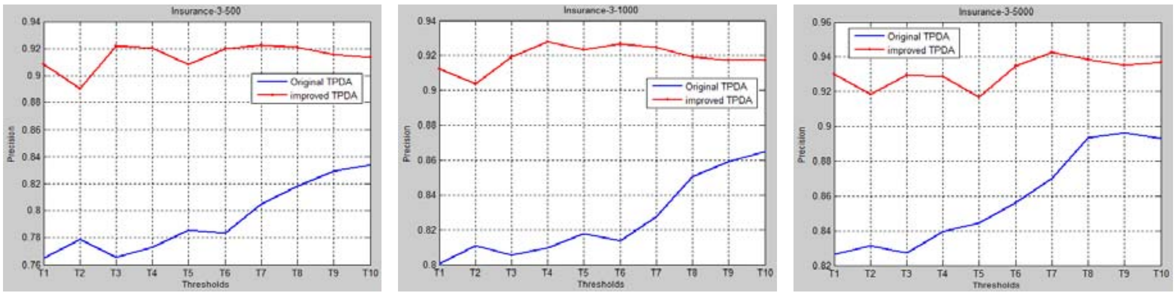

Fig. 16. Learning precision comparison of improved *TPDA* (Insurance-3)

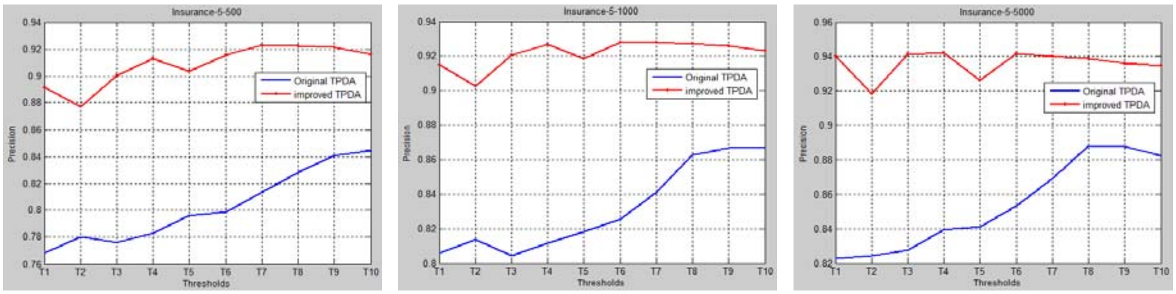

Fig. 17. Learning precision comparison of improved *TPDA* (Insurance-5)

### 3.3 Maize Expression Data Experiment

We have assembled a global maize germplasm collection with 368 elite inbred lines (Association Mapping Panel, AMP). We have got the expression data of 28,679 genes [19]. Please visit http://www.maizego.org/ or http://modem.hzau.edu.cn/ [20] to get the detailed information of the data set. We have used the biostatistics methods to calculate the *p*-value between every two genes according to the gene expression of maize, and select the top 189 gene regulatory relationships to do the comparison and analysis. These 189 gene regulatory relationships are stored in the file of *gene_links* (see supplemental material), which involves about 39 genes.

(1) Learning effect comparison of different thresholds

In the case of setting different thresholds in the three stages of *TPDA*, we compare the learning effect when to set different thresholds, as seen in Table 1. *X* denotes the threshold in the stage of *Drafting*, and *Y* denotes the threshold that is set in the stages of *Thickening* and *Thinning*.

Similarly, $Precision = Num_{match} / Num_{total}$ is used to do the precision calculation. $Num_{total}$ denotes the total number of edges in the learning Bayesian network. $Num_{match}$ denotes the common edge number in the learning Bayesian network and the gene regulatory relationships in the above mentioned *gene_links* (see supplemental material).

TABLE 1

Learning Results of Different Thresholds on Real Maize Data Set

| X<br>Y | 0.06 | 0.07 | 0.08 | 0.09 | 0.1 | 0.2 | 0.3 |
|---|---|---|---|---|---|---|---|
| 0.01 | 110/165=0.667 | 102/150=0.68 | 93/132=0.705 | 88/117=0.752 | 77/98=0.786 | 20/21=0.952 | 9/9=1 |
| 0.02 | 60/100=0.6 | 57/95=0.6 | 54/88=0.614 | 53/79=0.671 | 49/70=0.7 | 19/20=0.95 | 1 |
| 0.03 | 50/73=0.685 | 46/69=0.667 | 45/68=0.662 | 43/64=0.672 | 40/57=0.702 | 18/19=0.947 | 1 |

Obviously, the *TPDA* method has different learning effect when to set different thresholds. It has relative better learning effect when to set the threshold of *Drafting* stage to 0.1, and set the threshold of *Thickening* and *Thinning* to 0.01.

(2) Learning precision comparison of different methods

In the case of setting the threshold in the *Drafting* stage to 0.1, and setting the threshold of *Thickening* and *Thinning* to 0.01, we use 10 kinds of Bayesian network learning methods to construct the gene regulatory network of the above mentioned 39 genes. The learning result is shown in Table 2.

TABLE 2
Learning Precision Comparison of Different Methods

| Methods | *TPDA* | *gs* | *hc* | *iamb* | *mmpc* | *rsmax* | *tabu* | *fastiamb* | *interiamb* | *mmhc* |
|---|---|---|---|---|---|---|---|---|---|---|
| Total edge number | 98 | 25 | 70 | 51 | 34 | 24 | 70 | 54 | 52 | 33 |
| common edge with *gene_links* | 77 | 9 | 44 | 26 | 16 | 9 | 44 | 32 | 29 | 16 |
| Precision | 0.786 | 0.36 | 0.6286 | 0.5098 | 0.4706 | 0.375 | 0.6286 | 0.5926 | 0.5577 | 0.4848 |

Compared with other 9 kinds of Bayesian network learning methods, our improved *TPDA* method can learn more number of gene regulatory relationships in *gene_links*. The total edge number and the common edge number with *gene_links* of our improved *TPDA* is the maximum, and thus the learning precision of this method is the largest of all. The *TPDA* method can get 98 gene regulatory relationships in total, the common edge number with *gene_links* is 77, accounting for 78.6% of the total number of learning results, accounting for 41% of the total edge number in *gene_links*.

## 4 CONCLUSION

Mine the gene regulatory relationships and constructing gene regulatory network (*GRN*) is a challenging and significant research problem in biological studies. In this work, we use an improved three-phase dependency analysis (*TPDA*) Bayesian network learning method to construct the gene regulatory network. It mainly uses the entropy estimation approach of Gaussian kernel probability density estimator to calculate the (conditional) mutual information between genes. The data set of public benchmark and our maize global germplasm are used to do the validation of the improved method. For the discrete values in the benchmark data set, the experiment results show our *TPDA* method is suitable for processing the data with large sample size, with small number of discrete values, and the frequency of different discrete values is about same. The experiment results about the real maize data set show the improved *TPDA* method performs better than other 9 kinds of Bayesian network learning algorithms, and it shows better learning precision than the original *TPDA* method.

## ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China under Grant No.31601078, the Natural Science Foundation of Hubei Province under Grant No.2016CFB231, the Fundamental Research Funds for the Central Universities under grant No. 2015BC017, the Fundamental Research Funds for the Central Universities under grant No.2662018JC030.

## REFERENCES

[1] Y. X. R. Wang and H. Huang, "Review on statistical methods for gene network reconstruction using expression data," *Journal of Theoretical Biology*, vol.362, pp.53-61, 2014.

[2] L. E. Chai, S. K. Loh, S. T. Low, et al., "A review on the computational approaches for gene regulatory network construction," *Computers in biology and medicine*, vol. 48, pp.55-65, 2014.

[3] T. S. Gardner, D. Di Bernardo, D. Lorenz, et al., "Inferring genetic networks and identifying compound mode of action via expression profiling," *Science*, vol. 301, no. 5629, pp.102-105, 2003.

[4] Y. Wang, T. Joshi, X. S. Zhang, et al., "Inferring gene regulatory networks from multiple microarray datasets," *Bioinformatics*, vol.22, no. 19, pp.2413-2420, 2006.

[5] P. Vera-Licona, A. Jarrah, L. D. Garcia-Puente, et al., "An algebra-based method for inferring gene regulatory networks," *BMC Systems Biology*, vol. 8, no. 1, pp.37, 2014.

[6] P. J. Pemberton-Ross, M. Pachkov and E. van Nimwegen, "ARMADA: Using motif activity dynamics to infer gene regulatory networks from gene expression data," *Methods*, vol. 85, pp.62-74, 2015.

[7] A. A. Margolin, I. Nemenman, K. Basso, et al., "ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context," *BMC Bioinformatics*, vol. 7, no.Suppl 1, pp. S7, 2006.

[8] X. Zhang, X. M. Zhao, K. He, et al., "Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information," *Bioinformatics*, vol. 28, no. 1, pp.98-104, 2012.

[9] X. Zhang, J. Zhao, J. K. Hao, et al., "Conditional mutual inclusive information enables accurate quantification of associations in gene regulatory networks," *Nucleic Acids Research*, vol. 43, no. 5, pp.e31-e31, 2015.

[10] F. Petralia, P. Wang, J. Yang, et al., "Integrative random forest for gene regulatory network inference," *Bioinformatics*, vol. 31, no. 12, pp.197-205, 2015.

[11] S. Isci, H. Dogan, C. Ozturk, et al., "Bayesian network prior: network analysis of biological data using external knowledge," *Bioinformatics*, vol. 30, no. 6, pp.860-867, 2014.

[12] E. S. Adabor, G. K. Acquaah-Mensah and F. T. Oduro, "SAGA: A hybrid search algorithm for Bayesian Network structure learning of transcriptional regulatory networks," *Journal of Biomedical Informatics*, vol. 53, pp.27-35, 2015.

[13] S. Gao and X. Wang, "Quantitative utilization of prior biological knowledge in the Bayesian network modeling of gene expression data," *BMC Bioinformatics*, vol. 12, no. 1, pp.359, 2011.

[14] Y. Xiao, Y. Lv, H. Zhao, et al., "Predicting the Functions of Long Noncoding RNAs Using RNA-Seq Based on Bayesian Network," *Biomed Research International*, vol. 2015, pp.839590, 2015.

[15] X. W. Chen, G. Anantha and X. Wang, "An effective structure learning method for constructing gene networks," *Bioinformatics*, vol. 22, no. 11, pp.1367-1374, 2006.

[16] W. Gong, N. Koyano-Nakagawa, T. Li, et al., "Inferring dynamic gene regulatory networks in cardiac differentiation through the integration of multi-dimensional data," *BMC Bioinformatics*, vol. 16, no. 1, pp.74, 2015.

[17] J. Cheng, G. Grainer, J. Kelly, et al., "Learning Bayesian networks from data: An information-theory based approach," *Artificial Intelligence*, vol. 137, pp.43-90, 2002.

[18] I. Tsamardinos, L. E. Brown and C. F. Aliferis, "The max-min hill-climbing Bayesian network structure learning algorithm," *Machine learning*, vol. 65, no. 1, pp.31-78, 2006.

[19] J. Fu, Y. Cheng, J. Linghu, et al., "RNA sequencing reveals the complex regulatory network in the maize kernel," *Nature communications*, vol. 4, no. 4, pp.2832, 2013.

[20] H. J. Liu, F. Wang, Y. J. Xiao, et al., "MODEM: Multi-omics data envelopment and mining in maize," *Database: The Journal of Biological Databases and Curation*, vol. 2016, pp.baw117, 2016.