Comparative Study of Score Functions for Edge Orientation Problem in Network Estimation

Hitoshi AFUSO[†], **Takeo OKAZAKI**[††] and **Morikazu NAKAMURA**[††]

[†]Graduate School of Engineering and Science in University of the Ryukyus, Okinawa, Nishihara Senbaru 1, JAPAN
[††]Faculty of Information Engineering in University of the Ryukyus, Okinawa, Nishihara Senbaru 1, JAPAN

Summary
In this paper, we compared the score functions for edge orientation problem in estimation of genetic network from DNA microarray data. We focused four score functions, Bayesian Information Criterion(BIC), Bayesian-Dirichlet Metric(BDM), K2 Metric and PageRank Orientation Metric(PROM). To compare and evaluate the performance of each score function in various situations, we generated genetic networks changes of gene expression level, artificially. To generate such artificial networks, we used subsampling technique from large-scale real transcriptional regulatory network. To determine the set of genes that observed their expression level changed, we considered not only the structures of generated networks, but also the certain type of gene regulation that it makes difficult to detect the change of expression levels in knock-out or knock-down DNA microarray experiments.

Key words:
TRNs estimation, Edge Orientation Problem, Bayesian score functions, PageRank Orientation Metric

1. Introduction

Inside life-form cells, many genes or proteins interact each other, and these interactions utilize certain biological functions. Recently years, the understanding of these complex interactions had received attentions. The technique that can observe the expressions of large amount of genes at a time gives us massive information about the internal connections among genes in life-form cells. As a instance of such technique, we can see DNA microarray[1]. Up to now, various studies using DNA microarray data has been done. In those studies, the estimation of transcriptional regulatory networks(TRNs) is the one of most challenging topic. TRNs represent the relationships about regulatory among genes as a directed and edge-labeled graph in which each node represent a gene and edge denotes that existence regulatory relationship among genes. The labels on each edge are corresponding to whether the corresponding regulation is positive or negative regulation.

Recently years, various methods to estimate TRNs from DNA microarray data have been proposed. As first example of them, we can cite Boolean network. Boolean network represents gene expression as binomial value, on or off, and considers the relationships among genes as logical functions constructed from logical operators such NOT, OR and AND among them. Akutsu et al[2] and Liang[3] used gene expression data based on various experimental conditions and proposed the method to estimate Boolean network. While Boolean network model can represent the complex relationships among genes simply, it has pointed out its weakness to the noise on DNA microarray data. To solve that, Akutsu[4] and Shumulevich et al[5] proposed that new Boolean network model that can handle noise and method to estimate it from noisy DNA microarray data.

As the another example of the estimation method, Friedman et al[6] proposed that method based on Bayesian network model that is one kind of graphical model. The method that based on Bayesian network model had succeeded to reconstruct the relatively large TRNs comparing to another method. However, almost Bayesian network model has to discretize continuous DNA microarray data as the processing of estimation. This denotes that the estimation results may depend on the result of discretization. To solve that problem, Friedman et al[6] extended Bayesian network by linear model and Imoto et al[7] produced the method that based on non-parametric regression model. While the approaches based on Bayesian network were very successful, they have a difficulty for application to estimation of actual TRNs. That is, the class of TRNs that Bayesian network model can handle limited to that ones have no cyclic structure. It is said that such cyclic structures exist in real life-form system such as metabolic system for glucose. Therefore, there are some difficult cases to apply Bayesian network approach to. To exceed that limitation, Kim et al[8] proposed new kind of Bayesian network, that called Dynamic Bayesian network. By using time series expression data, the model can estimate the cyclic regulatory relationships. However, it is said that computational complexity of such Bayesian methods that use time-seriese data is so high and then, it becomes difficult to estimate TRNs in practical size.

The methods that showed above estimated TRNs as directed networks. On the other hand, Toh et al[9] and Basso et al[10] had proposed the method to estimate genetic network as undirected network. However, it is difficult to say that we can understand the underlying

Manuscript received September 5, 2012
Manuscript revised September 20, 2012

phenomena, such as disease developments, without the knowledge about the directions of regulatory relationships.

To overcome the problem that undirected network estimation methods contain, Afuso et al[11] proposed the method was constructed from two steps. At first step of that method, causal relationships among genes are detected as paths without direction. And next, searching the orientation to obtained undirected paths such that maximize certain score function would be utilized. To determine the orientation of undirected edge, i.e., to determine the direction of causal relationships, we can use several score functions. As traditional score function, one can see Bayesian Information Criterion(BIC), Bayesian Dirichlet Metric(BDM) and K2 Metric. And also, in Afuso[], new kind of score function, called PageRank Orientation Metric(PROM), had been proposed for orientation to undirected edges. Although, in actual case, it is difficult to determine which score function is more suitable for orientation to given undirected graphs.

In this paper, we compared four score functions above, BIC, BDE, K2 and PROM, to determine which score function can lead more accurate edge orientation. To compare those score functions in various situations, we need varied patterns of DNA microarray data and TRNs. But it is difficult to collect such actual data. Then, in this paper, the artificial TRNs and DNA microarray data were generated. Using artificial TRNs and generated DNA microarray data from those, comparisons of four score functions were utilized according to local landscape around the true orientation.

The rests of this paper were organized as follows. First we gave the explanation about the problem that originally defined in Afuso[11] and modified it slightly to make the comparison of score function easier. Second, we showed brief introduction about four score functions compared in this paper. Next, the generation method of artificial TRNs was shown. After that, we proposed the method to generate artificial data denotes DNA expression level changes from artificial TRNs. After that, we utilized the comparative experiments using TRNs in *E.coli*. And finally, we concluded our results.

2. Edge Orientation Problem

In this paper, we focused the problem that corresponds to Edge Orientation Phase in Afuso[11]. We call this problem Edge Orientation Problem. it is to find the orientation to each edge in given undirected graph such maximize certain score function. The problem is formulated as follows.

[INPUT]
1. Undirected graph G that represents direct interaction among genes.
2. (n, d) matrix **M** contains the resulted value of DNA microarray experiments. Where n and d denote the number of samples and the one of vertices, respectively. Note that this matrix **M** contains continuous value corresponding expression of each gene in its elements.
3. Score function *score* for evaluation of the directed graph G' that obtained by assigning orientations to each edge in undirected graph G.
[OUTPUT]
Directed graph G' such that maximize the value of given score function, score.

Originally, the one of input of edge orientation problem was a matrix that elements contain the results of DNA microarray experiments. On the other hand, we need to discretize given data to calculate the BIC or other score function. Then, in actual analysis, we have to distinguish the genes which their expression level had been changed from these continuous values. There are various criteria to make such decision among biologists and it has huge effect for DNA microarray analysis. We wanted to compare just only ability of edge orientation without considering such critical problem. To do so, we modified above problem as follows.

[INPUT]
1. Undirected graph G that represents direct interaction among genes.
2' (n, d) binary matrix **M**' contains the {0, 1} value according to corresponding gene' expression level changed or not.
3. Score function *score* for evaluation of the directed graph G' that obtained by assigning orientations to each edge in undirected graph G.
[OUTPUT]
Directed graph G' such that maximize the value of given score function, score.

Modified point is only that continuous input matrix **M** to binomial matrix **M**'. Matrix **M**' contains the information about the change of expression level for each gene. In this sense, we refer this matrix **M**' as Gene Expression Level Change(GELC) data matrix. Considering this modified Edge Orientation Problem, we compared the performance of each score function for edge orientation.

3. Score Functions for Edge Orientation

In the Edge Orientation Problem, there are four score functions. Bayesian Information Criterion(BIC) is most popular one of Bayesian approach. In this score function, all variables in the network are assumed that they are samples from multinomial distribution. It is based on maximization of posterior probability. BIC score of

directed graph *S* is calculated with following expression Eq.1

$$BIC(S)=\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}(N_{ijk}+\alpha_{ijk}-1) \times \log\frac{N_{ijk}+\alpha_{ijk}-1}{N_{ij}+\alpha_{ij}-r_i}-Dim(S)\log(N) \quad (1)$$

where n, q_i and r_i denote the number of variables, the one of value configuration of parents of variable *i* and one of instances of variable *i*, respectively. N_{ijk} corresponds to the frequency of *k*-th instance of the value of variable *i*, in the *j*-th situation of parents' condition. The term *Dim*(*S*)log(*N*) is the penalty for complexity of the structure of directed graph *S*. α_{ijk} is the prior information about the frequencies of corresponding instances in each parents' state. That constant can be considered as the pseudo-frequency of the instances that not observed in actual data. If one has no information about the variables, they set to zero.

Bayesian Dirichlet Metric(BDM) is a another score function based on Bayesian approach. Given a directed graph *S*, corresponding BDM score *BDM(S)* is calculated by following formula, Eq.2.

$$BDM(S)=\sum_{i=1}^{n}\left\{\sum_{j=1}^{q_i}\left\{\log\frac{\Gamma(\alpha_{ij})}{\alpha_{ij}+N_{ij}}+\sum_{k=1}^{r_i}\log\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})}\right\}\right\} \quad (2)$$

where Γ denotes the gamma function and α_{ijk} are the parameters that Dirichlet distribution. BDM can be obtained based on the assumption, so-called *Dirichlet prior*. In that assumption, each variable in given Bayesian network is considered should be observed in the data according to corresponding parameters in Dirichlet distribution. In other words, each variable is Dirichlet distributed under the certain prior conditions. If there is no prior information about the probability distributions of variables, then its parameters α_{ijk} are set to one certain value. In such special and typical case, BDM are also called Bayesian Dirichlet equivalent Metric(BDe).

As one more another kind of score function based on Bayesian approach, we can see K2 Metric. K2 Metric is special case of BDM. In BDM, if we have no information about the probability distribution that each variable is distributed, the parameters in Dirichlet distribution are set to certain same value. In K2 Metric, in addition, the value set to 1. In such case, variables are not according to Dirichlet distribution. Instead of that, they are distributed by unique distribution. K2 Metric can be calculated by using following formula, Eq.3.

$$K2(S)=\sum_{i=1}^{n}\left\{\sum_{j=1}^{q_i}\left\{\log\frac{r_i}{r_i+N_{ij}}+\sum_{k=1}^{r_i}\log N_{ijk}\right\}\right\} \quad (3)$$

Previous four score functions are based on probabilistic approach. On the other hand, in Afuso[11], the score function is based on network structure had been pro- posed. This score function is called as PageRank[12] Orientation Metric(PROM). In the calculation of it, at first, we estimate PageRank of the target TRNs from given DNA microarray data. Next, PageRank value of candidate TRNs is also calculated. And finally, these two PageRank values are compared and if these values are similar, then candidate and target TRNs are considered as also be similar. To estimate the PageRank value from DNA microarray data, the absolute values in the data are summed up over the experiments and normalized it to ensure that its 1-norm is 1, same like to original PageRank vector is. To measure the similarity between estimated PageRank from data and the one from candidate TRNs, we used correlation function because PageRank is relative value. To summarize, the PROM value *PROM(S)* corresponding to given directed graph *S* is calculated with the formula, Eq.4.

$$PROM(S)=Corr(PR_{est}, PR_{cand}) \quad (4)$$

In Eq.4, PR_{est} and PR_{calc} denote the estimated and calculated PageRank, respectively.

Using these four score functions, we can estimate structure of Bayesian network. In other words, we can obtain the orientation to each path in given undirected graph.

4. Generation of Artificial Data for Comparison Experiment

To compare the score functions above in varied situations, we need various type of TRNs that are known whole structure in advance and DNA microarray data corresponding those networks. However, it is difficult to collect such actual data. Instead of that, we used the artificial data. The generation method of artificial data is constructed from two parts, generation of artificial TRNs and one of artificial GELC data matrix.

4. 1 Generation of Artificial TRNs

For valid comparison, it is necessary to produce realistic TRNs. In other words, we need to generate the network that has structural property similar to actual TRNs. In some researches, such structural property had been shown. As one interesting example of that, we can see the existence of network motifs[13] in actual TRNs. Network motifs denote the sub-structures that occur in given network, significantly. As other example, there are two major structural property, Scale-free[14] and Small-world property[15]. Scale-free property denotes that the network

has hierarchical structure. Small-world property can be considered that denotes rough density of edges in any sub-structure. To generate realistic network that has properties above artificially, the generation method had been proposed[16]. Its main idea is the extraction of sub-structure in large actual network given in advance. The subgraphs extracted with such way are called *Modules.* The method to extract modules used the certain score, so-called *Modularity*[17]. Modularity reflects the significance of edge density in extracted module. It represents the difference of expectation value of edge density in given module comparing with the one that assumed its vertices were connected at random. Module extraction starts from a seed vertex that is selected randomly among the nodes of the source network. From this seed, a subgraph is grown by iteratively adding vertices to it until a desired size is reached. At each step, from all neighbors of the subgraph, we select the vertex that leads to the highest modularity *Q*.

$$Q = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s} \tag{5}$$

where m is the total number of edges in the graph, **s** is the index vector defining the module such that $s_i = 1$ if vertex i is part of the module and $s_i = -1$, if vertex i is not. Matrix **B** is called *Modularity matrix* with elements $B_{ij} = A_{ij} - P_{ij}$. A_{ij} is the actual number of edges between vertex i and j and $P_{ij} = k_i k_j / 2m$ is expected number of edges in a randomized graph. k_i is a degree of vertex i. Using this modularity extraction method and real large TRNs of *E.coli*[18] we generated artificial TRNs in our experiments.

4. 2 Generation of Artificial Data

To generate GELC data matrix from given TRN, we were used the effect propagation model. It was constructed from following assumptions. First, in the each DNA microarray experiment, only one gene would be changed its expression level by experimental stimulation. This means that we can control each gene's expression level ideally. Second, the generated effect of expression level changes in a particular gene *g* would be propagated to a gene *g'* that is regulated by gene *g* and *g'* would be chosen at random from genes regulated by *g*. This assumption models that selective regulation of genes controlled by unobservable variables. And Finally, in each DNA microarray experiment, the changes of expression level never vanish until they are observed. This assumption is corresponding to that speed of effect propagation is high enough comparing to the interval of observation in DNA microarray experiments. Summing up above assumption, in our propagation model, each effect of expression level change is propagated at random according to the structure of corresponding TRN. In other words, propagation can be considered as a random walk over the TRN. By iterating this random walk, we generated artificial gene expression level change data corresponding to given TRN. From adjacency matrix **A** corresponding to given TRN, we generated artificial gene expression data matrix **M**' with following pseudo-code, GENERATEARTICIFIALDATA**.**

procedure GENERATEARTIFICIALDATA(Adjacency matrix **A** of corresponding to TRN, Size of data m)
1 : matrix **M**' = **0**.
2 : i = 0
3 : **WHILE**(i < m)
4 : vector **v** = **0**
5 : vector **u** = **v**
6 : Select a index i of **v** at random
7 : $v_i = 1$
9 : **REPEAT**
10: $\mathbf{v}_{next} = \mathbf{A}\mathbf{v}$
11: Select a index j from which elements of $\mathbf{v}_{next}$ is 1.
12: $v_j = 1$ and $v_{others} = 0$
13: **u** = **u** + **v**
14: **UNTIL** any elements of **u** were unchanged.
15: Set **u** to i-th row of **M**'
16: $i = i + 1$
15: **END WHILE**
18: **RETURN** **M**'

Procedure. 1 Procedure of Data Generation from given TRN.

Using above procedure, we generated the artificial gene expression level data that elements have 0 or 1.

5. Design of Comparison Experiments

To clarify which score function lead to accurate orientation, we compared the four score functions, BIC, BDM, K2 and PROM. For utilization of valid comparison of them, we designed comparison experiments as follows.

5. 1 Data Preparation

In the generation of artificial TRNs using module extraction method shown in Section 3.1, we have only one real large TRN of *E.coli*[18]. Because of such restriction, we have to confirm extracted TRNs contain varied sub-structures of real one. On the other hand, to capture the feature of structure of given network, some network characteristics had been proposed, such as average clustering coefficient[19]. It is possible to represent given

network as a vector of such network characteristics value. Then, we prepared TRNs that have 100 vertices for each using following steps.

1. Generate 160 TRN that each one has 100 vertices using module extraction method and large real *E.coli* TRN[18]. The real TRN we used contains 1,565(about 1,600) vertices and initial vertex of module extraction was determined at random. From these facts, we extracted 160 TRNs so that each vertex in real TRN could be extracted ten times averagely.
2. Calculate 13 network characteristics for each generate TRNs to represent each TRN as a 13 dimensional vector. After such vector representation, normalization of each dimension was utilized. Now, we have 160 vectors in 13-dimensional space. See Appendix for details of 13 network characteristics we used.
3. Execute clustering using Ward method and Euclid distance. By considering obtained dendrogram, we manually divide them into some clusters.
4. Choose centroids for each cluster as test subject TRNs.

By using above steps, we can generate artificial TRNs that contains varied structure similar to real one. In this paper, we generated 11 representative TRNs as test subject. From obtained TRNs, we constructed GELC data matrix using Procedure.1 shown in Section 3.2. Considering that the accuracy of PageRank estimation from artificial data might be influenced by the number of sampled random walks, we generated GELC data matrix in three cases, the matrix consists of 1000, 10,000, and 100,000 samples. And also, considering that the source point of random walk is chosen at random, we produced 30 GELC data matrix for each case. Eventually, we constructed the dataset consists of 11 TRNs and for each TRN, corresponding GELC data that contains 30 GELC data matrices in three cases.

5. 2 Evaluation Index

In the comparison of score functions, we focused the point whether true orientation of each TRN would be optimum solution when some candidates of orientations are given. That is to say, we focused the coherence of score functions to true orientation. If given score function of orientation doesn't have such coherence, it becomes difficult to find true orientation by solving optimization problem with such score function. In this sense, the coherence is critical index. In the evaluation of score functions using coherence, we need to determine the set of candidates as solutions of edge orientation problem. To this end, we produced candidates from each TRN by alternating the direction for each edge in true TRN. To measure the coherence of score functions, we used the ratio of the solutions that has better score value than true TRNs, as score. We referred this score as R score in this paper. As shown in Section 4.1, GELC dataset contained 30 data for each case of the number of samples. Then, we calculated minimum and maximum of R score in 30 data for each case. They correspond to best and worst coherence in 30 trials, respectively. Using minimum and maximum of R score, the coherence of score function had been measured in the comparison experiments.

5. 3 Other Parameter Settings

In the experiments, we assumed that no information about the probability of each gene expression was given. Then, all parameters to BDM were set to 10. For calculation of PageRank of candidate directed graph, the damping factor was set to 0.85.

6. Experimental Result

In this section, we showed the comparison results in obtained from above experiments.

The minimum values of R score in each case of the number of samples were shown in Table.1 to .3. In those tables, the best value was typed in bold and underlined. As shown in Table.1 to .3, PROM resulted best in almost TRNs. And K2 could not lead best result in any case in these experiments. In these experiments, it was shown that PROM is preferred for almost TRNs that extracted from *E.coli* TRN in view of best coherence. However, from Table.4, we can see that PROM resulted worst value and BIC lead best value for all TRNs. These results show that it is risky to use PROM for edge orientation in the case that the number of samples is small. But Table.5 and .6, such risk decreases as increasing the number of samples. Eventually, as shown in Table.6, PROM would be best choice in the case such we have enough number of samples comparing to the number of vertices. And PROM only had such monotonic response against the increasing the number of samples.

Summing up above results, PROM would be better choice in the case that we can assume that give data was produced under the random propagation model.

7. Conclusion

In this paper, we compared four score functions for edge orientation problem to clarify which one is suitable for various TRNs' structures. To this end, we generated many artificial TRNs using the module extraction method. And next, we selected representatives to ensure that obtained TRNs contain varied structure of real TRN of *E.coli*. Using generated TRNs, GELC data was constructed by using certain model that means ideal experimental conditions in DNA microarray experiments. The

comparison was utilized focusing to the coherence of score functions because that property of score function is critical for finding the accurate edge orientation by solving optimization problem. The comparison results showed that, PROM would be better choice under the situation we can assume that given data was produced by random walk over TRNs with enough large sample size.

In this paper, we focused only ideal GELC data generation model. Then as future task, we can see other comparison experiments under the more plausible GELC generation model.

References

[1] DeRisi.J.L, Lyer.V.R, and Brown.P.O, "Exploring the metabolic and genetic control of gene expression on a genomic scale", Science, 278, pp.680-686, 1997

[2] T.AKutsu, S.Kuhara, O.Maruyama, S.Miyano, "Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions", Proc 9th ACM-SIAM SODA, pp.695-702, 1998

[3] S.Liang, S.Fuhrman and R.Somogyi, "REVEAL, a general reverse engineering algorithm for inference of genetic network architectures", Pacific Symposium on Biocomputing, 3, pp.18-29. 1998

[4] T.Akutsu, S.Miyano and S.Kuhara, "Inferring qualitative relations in genetic networks and metabolic pathways", Bioinformatics, 16, pp.727-734, 2000

[5] I.Shmulevich, E.Dougherty, S.Kim, and W.Zhang, "Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks", Bioinformatics, 18, pp.261-274, 2002

[6] N.Friedman, K.Murphy, S.Russell, "Learning the structure of dynamic probabilistic networks", Proc of the 14th Conference on the Uncertainty in Artificial Intelligence, pp.139-147, 1998

[7] N.Friedman, M.Linial, I.Nachmann, and D.Pe'er, "Using Bayesian network to analyze expression data", Journal of Computational Biology, 7, pp.601-620, 2000

[8] S.Imoto, T.Goto and S.Miyano, "Estimation of genetic networks and functional structures between genes by using Bayesian network and nonparametric regression", 7th Pacific Symposium on Biocomputing, pp.175-186, 2002

[9] S.Kim, S.Imoto and S.Miyano, "Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data", Biosystems, 75, pp.57-65, 2004

[10] H.Toh and K.Horimoto, "Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling", Pharmacogenomics, 3, pp.507-525, 2002

[11] K.Basso, A.Margolin and G.Stolovitzky, "Reverse engineering of regulatory networks in human B cells", Nature Genetics, 37, pp.382-390, 2005

[12] H.Afuso, M.Nakamura, T.Okazaki, "Genetic network estimation with covariance selection and score function based on PageRank", IPSJ SIG, 58, pp.5-8, 2008

[13] S.Brin, L.Page, "The Anatomy of a large-scale hypertextual web search engine", Computer Networks and ISDN Systems, vol30, pp.107-117, 1998

[14] R.Milo, S.Shen-Orr, S.Itzkovits, N.Kashtan, D.Chklovskii and U.Alon, "Network motifs: simple building blocks of complex networks", Science 25:298(5594), pp.824-827, 2002

[15] Raya.K, Ernst.W, "How scale-free are biological networks", Journal of Computational Biology, 13(3), pp.810-818, 2006

[16] Watts.D.J, Strogatz.S.H, "Collective dynamics of small world networks", Nature, vol.393, pp.440-442, 1998

[17] Daniel Marbach, Thomas Shaffter, Claudio Mattiussi and Dario Floreano, "Generating realistic in silico gene network for performance assessment of reverse engineering methods", J Comput Biol, 16(2), pp.229-239, 2009

[18] Newman, M.E.J. "Finding community structure in networks using the eigenvectors of matrices". Phys. Rev. E, 74, 2006

[19] Gama-Castro.S, *et al,* "RegulonDB version 7.0: Transcriptional regulation of Escherichia coli K-12 integrated within genetic sensory response units", Nucleic Acids Res, 39, D98, 2011

[20] Zhiyu.L, Chen.W, Qiong.Z and Huayong.W, "Clustering coefficient queries on massive dynamic social networks", Proc. 11th International Conference on Web-age Information Management, pp.115-126, 2010

Table 1 Minimum R scores with 1,000 samples

Function	TRN #1	TRN #2	TRN #3	TRN #4
BIC	0.039	**0.000**	0.483	0.052
BDM	0.039	**0.000**	0.425	0.052
K2	0.137	0.033	0.266	0.260
PROM	**0.000**	**0.000**	**0.053**	**0.000**
Function	TRN #5	TRN #6	TRN #7	TRN #8
BIC	0.311	0.057	0.037	0.270
BDM	0.246	0.057	0.029	0.390
K2	0.275	0.209	0.119	0.306
PROM	**0.004**	**0.021**	**0.000**	**0.000**
Function	TRN #9	TRN #10	TRN #11	
BIC	0.012	0.016	0.217	
BDM	**0.006**	0.016	0.200	
K2	0.234	0.213	0.356	
PROM	0.028	**0.012**	**0.018**	

Table 2 Minimum R scores with 10,000 samples

Function	TRN #1	TRN #2	TRN #3	TRN #4
BIC	0.039	**0.000**	0.491	0.052
BDM	0.078	0.279	0.366	0.052
K2	0.186	0.101	0.375	0.513
PROM	**0.000**	**0.000**	**0.000**	**0.000**
Function	TRN #5	TRN #6	TRN #7	TRN #8
BIC	0.362	**0.057**	0.037	0.290
BDM	0.434	0.076	0.246	0.418
K2	0.420	0.257	0.276	0.597
PROM	**0.000**	0.136	**0.000**	**0.000**
Function	TRN #9	TRN #10	TRN #11	
BIC	0.018	0.016	0.200	
BDM	0.066	0.180	0.278	
K2	0.349	0.270	0.408	
PROM	**0.004**	**0.000**	**0.000**	

Table 3 Minimum R scores with 100,000 samples

Function	TRN #1	TRN #2	TRN #3	TRN #4
BIC	0.029	**0.000**	0.425	0.000
BDM	0.049	0.338	0.241	0.304
K2	0.166	0.084	0.350	0.747
PROM	**0.000**	**0.000**	**0.000**	**0.000**
Function	TRN #5	TRN #6	TRN #7	TRN #8

BIC	0.311	**0.047**	0.022	0.294
BDM	0.362	0.104	0.328	0.597
K2	0.391	0.219	0.425	0.824
PROM	**0.000**	0.122	**0.000**	**0.000**
Function	TRN #9	TRN #10	TRN #11	
BIC	0.006	0.008	0.217	
BDM	0.271	0.319	0.234	
K2	0.475	0.532	0.400	
PROM	**0.004**	**0.000**	**0.000**	

Table 4 Maximum R scores with 1,000 samples

Function	TRN #1	TRN #2	TRN #3	TRN #4
BIC	**0.059**	**0.305**	**0.558**	**0.243**
BDM	0.490	0.331	0.658	0.417
K2	0.853	0.881	0.725	0.826
PROM	0.875	1.000	0.750	0.538
Function	TRN #5	TRN #6	TRN #7	TRN #8
BIC	**0.449**	**0.200**	**0.194**	**0.311**
BDM	0.522	0.400	0.284	0.458
K2	0.812	0.714	0.858	0.582
PROM	0.714	0.835	0.641	0.616
Function	TRN #9	TRN #10	TRN #11	
BIC	**0.181**	**0.107**	**0.383**	
BDM	0.247	0.393	0.600	
K2	0.693	0.820	0.765	
PROM	0.882	0.795	0.380	

Table 5 Maximum R scores with 10,000 samples

Function	TRN #1	TRN #2	TRN #3	TRN #4
BIC	**0.245**	**0.178**	0.633	0.252
BDM	0.598	0.805	0.575	0.461
K2	0.892	0.873	0.675	0.843
PROM	0.600	1.000	**0.447**	**0.092**
Function	TRN #5	TRN #6	TRN #7	TRN #8
BIC	0.449	**0.419**	0.239	0.315
BDM	0.558	0.638	0.567	0.478
K2	0.819	0.667	0.851	0.765
PROM	**0.022**	0.705	**0.019**	**0.007**
Function	TRN #9	TRN #10	TRN #11	
BIC	**0.133**	**0.262**	0.278	
BDM	0.373	0.648	0.565	
K2	0.729	0.910	0.800	
PROM	0.373	0.487	**0.038**	

Table 6 Maximum R scores with 100,000 samples

Function	TRN #1	TRN #2	TRN #3	TRN #4
BIC	0.176	0.364	0.500	0.374
BDM	0.863	0.831	0.358	0.643
K2	0.882	0.983	0.442	0.939
PROM	**0.008**	**0.254**	**0.008**	**0.000**
Function	TRN #5	TRN #6	TRN #7	TRN #8
BIC	0.370	0.476	0.201	0.315
BDM	0.536	0.533	0.619	0.649
K2	0.623	0.400	0.910	0.912
PROM	**0.013**	**0.345**	**0.006**	**0.000**
Function	TRN #9	TRN #10	TRN #11	
BIC	**0.048**	0.320	0.548	
BDM	0.506	0.762	0.548	
K2	0.741	0.861	0.713	
PROM	0.085	**0.006**	**0.025**	

Hitoshi AFUSO received the B.S. and M.S. degrees in Information Engineering from University of the Ryukyus in 2005 and 2008, respectively. He belongs to doctoral course in same university. His research area is Bioinformatics. Espe-cially, he is interested in stochastic model of genetic expression and alignment of biological network such as TRNs.

Takeo OKAZAKI took B.Sc., M.Sc. from Kyushu University in 1987 and 1989, respectively. He had been a research assistant at Kyushu University from 1989 to 1995. He has been a lecturer at University of the Ryukyus since 1995. His research interests are statistical data normalization for analysis, statistical causal relationship analysis.

Morikazu NAKAMURA took B.E., and M.E. from University of the Ryukyus in 1989 and 1991, respectively. He took ph.D from Osaka University in 1995. He has been a professor at University of the Ryukyus. His research interest includes design and analysis of parallel and distributed algorithms.

Appendix: List of Network Characteristics

We used following 13 network characteristics to represent give TRN as a vector.

List of network characteristics used in this paper

Density of edges, Mean of in-degrees, Variance of in-degrees, Mean of out-degrees, Variance of out-degrees, Mean of path length, Variance of path length, Mean of clustering coefficient, Variance of clustering coefficient, Mean of closeness, Variance of closeness, Mean of betweenness and Variance of betweenness