Inference of S-system models of genetic networks using product unit neural networks

doi: 10.1109/ICSMC.2008.4811480

Inference of S-system Models of Genetic Networks using Product Unit Neural Networks

Hiroaki Murata*, Makoto Koshino†, Masatomo Mitamura‡ and Haruhiko Kimura‡
* Graduate School of Natural Science and Technology, Kanazawa University
Kakuma town, Kanazawa city, Ishikawa prefecture, Japan
Email: hiro@blitz.ec.t.kanazawa-u.ac.jp
†Electronics and Information Engineering, Ishikawa national College of Technology, INCT
Tsubata town, Kahoku-gun, Ishikawa prefecture, Japan
‡Graduate School of Natural Science and Technology, Kanazawa University
Kakuma town, Kanazawa city, Ishikawa prefecture, Japan

***Abstract*—In this study, we proposed the method of inference of genetic networks which expresses the regulation of genes. The proposed method does not solve the differential equations, learns the genetic networks using Product-Unit-Neural-Network (PUNN) and infer the S-system model of genetic networks which describes a set of differential equations. The experimental results show the proposal method is 160 times faster than the previous method which estimated S-system model of genetic networks while maintaining equivalent performance to the previous method.**

***Keywords*—genetic networks, s-system model, product unit neural networks, PUNN, particle swarm optimization, PSO.**

I. INTRODUCTION

Within an organism, many genes interface with each other. This interaction of gene called genetic networks, and it discovers the gene function. Because of quantitative time-series observation of gene expression is becoming possible, for example by cell array technology, many researchers infer the mathematical models which describe the genetic networks from gene expression data. The purpose of the genetic network inference problem is to identify the mathematical model from the observed gene expression data. Among mathematical models which proposed to describe the genetic networks [1], [2] by many researchers, we focus on the mathematical model based on a set of differential equations because these models have an ability to capture the dynamics. In particular, we focus on the S-system model [3] and the Neural Network model (NN model) [4].

The S-system model is a set of non-linear differential equations in which the component processes are characterized by power-law functions. This model is expected to obtain hints about genetic networks because the power-law formalism is possible to express a variety of natural phenomena. The inference method using the S-system model (call the S-system model method) infers the genetic networks by inference the coefficient in a set of non-linear differential equations. Therefore, this model is time-consuming because it requires solving a set of non-linear differential equations. In the Neural Network model (NN model), we regard the differential equation, which describes the genetic network, as the relational the gene expression level and the differential coefficient of the gene expression level. The NN model is the layered-neural-network learned the relational expression. The inference method using the NN model (call the NN model method) infers the genetic networks by learning the relational expression. The NN model method is not required to solve a set of differential equations. This method is able to decrease in computational time than the S-system model method. However, since the relational expression becomes the black box, the NN model method is unable to infer detailed genetic network than the S-system model method.

In this study, we tried to develop the inference method which is able to infer genetic network as detailed as S-system model without a time-consuming. We then found the product-unit-neural-network (PUNN) is able to represent the S-system model without a becoming a black box. The PUNN includes a product-unit which computes the weighted product of input instead of weighted summation. This characteristic conforms to the S-system model which is power-law formalism. Therefore, in this study, we proposed the inference method using the product-unit-neural-networks (PUNN) model. The PUNN model is based on the NN model, replaced layered-neural-network by PUNN. Note that the PUNN model is product-unit-neural-network learned the relational expression. The inference method using the PUNN model (call the PUNN model method) infers the genetic networks by learning the relational expression. Because of the relational expression does not become the black box in the PUNN model method, the inference method using this model is expected detailed genetic network as S-system model in a short time than the S-system model method. The experimental results show the proposal method is 160 times faster than the previous method which inferred S-system model of genetic networks while maintaining equivalent performance to the previous method.

II. PREVIOUS WORKS: INFERENCE METHOD USING A DIFFERENTIAL EQUATIONS MODEL

A. *Differential Equations Model*

The differential equations model of genetic network is described as follows:

1-4244-2384-2/08/$20.00 © 2008 IEEE

$$\frac{dX_i}{dt} = G_i(X_1, \cdots, X_N) \quad (i = 1, \cdots, N) \tag{1}$$

where X_i represents expression level of ith gene, N is the number of genes in the system, and G_i is a function of an arbitrary form. Inference of the differential equations model of genetic network is to identify the function G_i from the observed gene expression data. However, because it is difficult to identity the arbitrary function G_i, some model including hypothesis have been proposed (see [2]). Among them, we focus on the S-system model [3] and the Neural Network model (NN model) [4].

B. S-system Model

1) Definition of the S-system Model: The S-system model is a simplified model of the general mass-action law which is capable of describing chemical reaction systems. In simplify the general mass-action, the S-system model excludes an environment and unknown gene. This model however expected to obtain hints about genetic networks. The S-system model of genetic network is described as follows:

$$\begin{cases} \dfrac{dX_1}{dt} = \alpha_1 \displaystyle\prod_{j=1}^{N} X_j^{g_{1,j}} - \beta_1 \prod_{j=1}^{N} X_j^{h_{1,j}} \\ \qquad\vdots \\ \dfrac{dX_N}{dt} = \alpha_N \displaystyle\prod_{j=1}^{N} X_j^{g_{N,j}} - \beta_N \prod_{j=1}^{N} X_j^{h_{N,j}} \end{cases} \tag{2}$$

where X_i represents expression level of ith gene, N is the number of genes in the system, i is the suffix of gene.The first term represents the total influence of X_j that increases X_i and the second term means total decreasing influence to X_i. α_i and β_i are rate constants. $g_{i,j}$ and $h_{i,j}$ are interaction coefficients and called kinetic order [1]. In the S-system model, the gene expression level is depend on blance between an excitatory and an inhibitory influence.

2) The inference method using S-system model: The S-system model represents a set of $\alpha_i, \beta_i, g_{i,j}, h_{i,j}$ called S-system parameter. The inference method using the S-system model (call the S-system model method) infers genetic network by finding the S-system parameter which minimizes f defined as follows:

$$f = \sum_{i=1}^{N} \sum_{t=1}^{T} \left(\frac{X_{cal,i,t} - X_{exp,i,t}}{X_{exp,i,t}} \right)^2 \tag{3}$$

where N is the number of genes in the system, T is the number of sampling points of the observed gene expression data. $X_{cal,i,t}$ is numerically calculated gene expression level at time t of ith gene using the set of differential equations (2). $X_{exp,i,t}$ is the observed gene expression level at time t of ith gene. The f means error between the observed gene expression data and the calculated gene expression. In other word, the S-system model method is solving a non-linear function optimization problems to find the S-system parameter, calculated gene expression data using which fits the observed gene expression data.

Since $2N(N+1)$ S-system parameters must be determined in order to solve the set of differential equations (2), the S-system model method finds the S-system parameter in $2N(N+1)$-dimensional space. This space is too high-dimensional in cases where we try to infer S-system model of large-scale genetic networks consisted many genes. To overcome this problem, Maki et al.[11] proposed the strategy of dividing the genetic network inference problem into several sub-problems. In this strategy, each sub-problem corresponds to each gene. The purpose of the sub-problem corresponding to ith gene is to find the S-system parameter (only corresponding to ith gene) which minimizes f_i defined as follows:

$$\begin{aligned} f_i &= \sum_{t=1}^{T} \left(\frac{X\prime_{cal,i,t} - X_{exp,i,t}}{X_{exp,i,t}} \right)^2 && (4) \\ \frac{dX_i}{dt} &= \alpha_i \prod_{j=1}^{N} Y_j^{g_{i,j}} - \beta_i \prod_{j=1}^{N} Y_j^{h_{i,j}} && (5) \\ Y_j &= \begin{cases} X_j & \text{if } j = i \\ \hat{X}_j & \text{if } j \neq i \end{cases} && (6) \end{aligned}$$

where $X\prime_{cal,i,t}$ is numerically calculated gene expression level at time t of ith gene using the differential equation (5). $\hat{X}_j$ is an estimated gene expression level of jth gene acquired not by solving a differential equations, but by making a direct estimation from the observed gene expression data. We can make $\hat{X}_j$s using an interpolation technique such as a spline interpolation or a local linear regression. In order to solve the differential equation (5), the S-system parameter needed to determine are only $2(N+1)$ parameters (i.e. $\alpha_i, \beta_i, g_{i,1}, \cdots, g_{i,N}, h_{i,1}, \cdots, h_{i,N}$). Thus, this strategy divides a $2N(N+1)$-dimensional problem space into N sub-problem which has $2(N+1)$-dimensional problem space.

The genetic network inference problem based on the S-system model may have multiple optima because the degree-of-freedom of the model is high and the observed gene expression data are usually polluted by the measurement error. To increase the probability of inferring a correct S-system model, the objective function including a priori knowledge of the genetic network is proposed. The priori knowledge is ”genetic networks is sparse connection”. The other introduced idea is Akaike’s information criteria (AIC) which is a measure of the goodness of fit an estimated statistical model.

C. Neural Network Model

1) Definition of Neural Network Model: The time complexity of the genetic network inference problem on the S-system model becomes huge size because it is frequently required to solve the differential equation (5). The neural network model (the NN model)[4] eliminates the need to solve a differential equations, and reduce the time complexity from

[1] In the strict sense of the word, because ”kinetic order” means ”reaction order” in chemical kinetics, we have to refer ”α_i and β_i” as ”amount corresponding to kinetic order”. In this study, however, we refer ”α_i and β_i” as ”kinetic order” in accordance with [6], [7].

S-system model. The differential equation (1), which described the genetic networks, is regarded as the relational expression of the differential coefficient of gene expression level ($\frac{dX_i}{dt}$) and the gene expression level (X_i) in the neural network model. The NN model is the layered-neural-network learned the relational expression.

2) The inference method using NN model: The inference method using the NN model (call the NN model method) infers the genetic networks by learning the relational expression. The NN model method[4] is not required to solve a set of differential equations. Here, the $\frac{dX_i}{dt}$ is acquired by making a direct calculation from the observed gene expression data [2]. This method is able to decrease in computational time than the S-system model method. However, since the relational expression becomes the black box, the NN model method is unable to infer detailed genetic network than the S-system model method.

III. PROPOSAL METHOD: INFERENCE METHOD USING A PUNN MODEL

In this study, we propose the product-unit-neural-network model (the PUNN model) and the inference method using this model (call the PUNN model method). The PUNN model is based on the NN model, replaced layered-neural-network by PUNN. Because the PUNN is able to describe the S-system model of genetic networks without a becoming a black box, the PUNN model method is expected detailed genetic network as S-system model in a short time than the S-system model method.

A. Product Unit Neural Network

The PUNN consists of two types of unit: the summation unit and the product unit. The summation unit is outputs the weighted sum of input signal, and is used in terminating the network. In the other, the product unit is outputs the weighted product of input signal, and is use in hidden layer. In the simple PUNN, which is consist of three layers, the output of kth unit in hidden layer and lth unit in output layer are calculate using equation (7) and equation (8), respectively.

$$y_k^H = \prod_{j=1}^{n_I} (X_j)^{w_{j,k}^{IH}} \tag{7}$$

$$y_l^O = \sum_{k=1}^{n_H} w_{k,l}^{HO} y_k^H - \theta_l^O \tag{8}$$

where n_I and n_H are the number of unit in input layer and output layer, respectively. X_j is the input of the jth input unit. $w_{j,k}^{IH}$ is the weight between the jth input unit and the kth hidden unit. $w_{k,l}^{HO}$ is the weight between the kth hidden unit and the lth output unit. θ_l^O is the threshold parameter of the lth output unit.

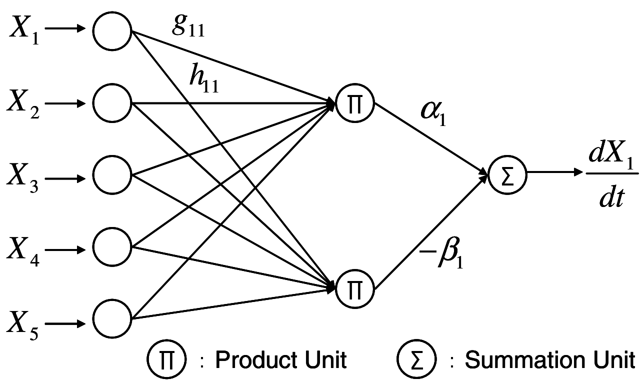

Fig. 1. The PUNN model which describes a part of genetic network compounded five genes.

B. Definition of Product Unit Neural Network Model

We proposed the PUNN model defined as the PUNN setting up as follows. $n_I = N$ (N is the number of gene in system), $n_H = 2$ and the number of output unit is one. The $w_{j,1}^{IH}$ and $w_{j,2}^{IH}$ are represent $g_{i,j}$ and $h_{i,j}$, respectively. The $w_{1,1}^{HO}$ and $w_{2,1}^{HO}$ are represent α_i and $-\beta_i$, respectively. The θ_1^O is 0. The output of the PUNN model is calculated using equations (9) which is described the S-system model of genetic network.

$$y_1^o = \alpha_i \prod_{j=1}^{N} X_j^{g_{i,j}} - \beta_i \prod_{j=1}^{N} X_j^{h_{i,j}} \tag{9}$$

Thus, the PUNN model is able to describe the S-system model of genetic networks without a becoming a black box. For instance, Fig. (1) shows the PUNN model described the genetic networks.

C. The inference method using PUNN model

1) The concept of the PUNN model.: The inference method using the PUNN (call the PUNN model method) infers by learning to the PUNN. In the PUNN model method, the PUNN is learned to minimize equation (11) which represents the error (in the strict sense of word, "the amount corresponding to error") between the calculated differential coefficient of gene expression level and the observed gene expression level. Here, $\mathbf{X_t}$ is the gene expression level of all of the genes at time t. The $y_1^o(\mathbf{X_t})$ is the output of the PUNN inputted $\mathbf{X_t}$, and equals to the calculated differential coefficient of gene expression level using S-system parameters. The y_t is estimated differential coefficient of observed gene expression level at time t. The reason why we use equation (11) instead of (10) is to evaluate accurately when small amount of equation (10).

Here, the gradient descent method, which is widely used for learning neural networks, fails to leaning PUNN in general[9], [10]. In this study, because of using the optimization algorithm to lean the PUNN , the minimized function (11) is not called

[2] In [4], the observed gene expression data is interpolated using spline interpolation

”error function” but called ”the objective function”.

$$e_{P,i} = \sum_{t=1}^{T} \left(y_1^o(\mathbf{X_t}) - y_t\right)^2 \qquad (10)$$

$$E'_{P,i} = \begin{cases} -\infty & \textbf{if } e_{P,i} = 0 \\ \log(e_{P,i}) & \textbf{if } 0 < e_{P,i} < 1 \\ e_{P,i} & \textbf{if } 1 \leq e_{P,i} \end{cases} \qquad (11)$$

2) The penalty term using clustering: One method of introducing a priori knowledge, which is “genetic network is sparse connection”, is introducing the penalty term in the objective function. In the previous works, as penalty term, the number of non-zero kinetic order is limited to fixed value. In contrast, we introduced the penalty term using dynamic limit by clustering technique. In the penalty term we introduced, the kinetic order is divided two cluster order by absolute value using clustering, and limited to the number of kinetic order in the cluster, the centroid of which is larger than the centroid of another cluster. The penalty term we proposed is defined as follows.

$$P_{P,i} = \begin{cases} \infty & \textbf{if } K_{nz} < 2 \\ 0 & \textbf{if } K_{nz} = 2 \\ \sum_{x \in K_{c1}} x + (2N - K_{nz}) & \textbf{else} \end{cases} \qquad (12)$$

Here, the kinetic order is divided three cluster; K_{c0} consisted a kinetic order value of 0, K_{c1} consisted a kinetic order of which absolute value is small, and K_{c2} consisted a kinetic order of which absolute value is large. The K_{nz} is the number of kinetic order in K_{c0}. The N is the number of genes, and $2N$ is the number of kinetic order. If $K_{nz} < 2$, the penalty term $P_{P,i}$ is set to ∞. The reason of this setting is that the S-system model, which balances between an excitatory and an inhibitory influence, is impossible to represent using only one kinetic order.

3) Dynamic Objective Function: When solving the optimization problem which has penalty term in the objective function, the dynamic coefficient is often included in the objective function to optimize efficiently. In this study, we also included the dynamic coefficient defined as equation (13) in the PUNN mode method. Here, t is time elapsing from starting the PUNN model method. Finally, the PUNN model method uses equation (13) to learn the PUNN.

$$E_{P,i} = c_P(t) \times E'_{P,i} + (1 - c_P(t)) \times P_{P,i} \qquad (13)$$

$$c_P(t) = \left(1 - (1 - \frac{t}{t_{end}})^2\right) \times 0.4 + 0.5 \qquad (14)$$

IV. OPTIMIZATION METHOD

The gradient descent method, which is widely used for learning neural networks, fails to leaning PUNN in general[9], [10]. The PUNN is learned using optimization algorithm such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and LeapFrog Optimization (LFOP) [10]. In this study, we use PSO to learn the PUNN, because PSO obtains a good result in both CPU time and leaning accuracy in [10].

```
Initialize($Indiv$)
$N_{Init}$ is setting at random integer value in $[2, \frac{N}{2}]$
Select a number of
Select $N_{Init}$ kinetic order in a random drawing, and those are named $X_a$.
for all kinetic order $i$ do
    if $i \in X_a$ do
        Initialize the kinetic order $i$.
    end if
end for
```

Fig. 2. The initialize method.

```
LocalSearch($Indiv$)
<The 1th phase>
$Indiv_0 = Indiv$
Loop $i = 1, \cdots, N_{L1}$ do
    for all parameter $j$ do
        $R_1$ initializes at random in $[0.95, 1.05]$.
        $Indiv_{i,j} = Indiv_{0,j} * R_1$
    end for
end Loop
$Indiv$ updates to best $Indiv_i$ for $(i = 0, \cdots, N_{L1})$
<The 2th phase>
Sort the kinetic order of $Indiv$ all together in ascending order of their absolute values. i.e. $|K(i)| \leq |K(i+1)|$ for $(i = 1, \cdots, 2N - 1)$
Loop $i = 1, \cdots, 2N$ do
    Generate a individual $indiv'$ from $Indiv$ by setting $K(i) = 0$.
    if $f(Indiv') < f(indiv)$ then
        $Indiv$ = $Indiv'$
    endif
end Loop
```

Fig. 3. The local search algorithm which updates $Indiv$. The 1th phase represents the new process we proposed, and the 2th phase represents the HCLS.

In this study, we also introduced the initializing method and local search. In the previous works, all parameter are initialize at random when generating initial individual. This means that generated genetic networks are not sparsely connected. Because genetic networks are known to be sparsely connected, we introduced this priori knowledge in initializing method. The Fig. 2 shows the initializing method we introduced. In this method, the initialize kinetic order limit to N_{init} which is randomize at $[2, \frac{N}{2}]$. As the local search, Hill-Climbing-Local-Search (HCLS), which targets at kinetic order, are proposed in the previous work[7]. In this study, we introduced a new local search by adding a new process to HCLS. The new process, which targets all parameter, is updating individual to best individual in N_L of a individual generating by adding a ±5% error to target individual. The Fig. 3 shows the local search we proposed.

V. EXPERIMENT

A. Experiment Setting

To show the effectiveness of the proposed method, we applied it to artificial genetic network inference problem which is widely used in previous works[6], [7]. The artificial genetic network we used is consists of five genes (N=5).

The gene expression data used in this experiment, which is called “the observed gene expression data” , is obtained by calculating using the equation (2). The target S-system parameter and initial value of gene expression level are listed in [6], and we used these value in this experiment. If an insufficient amount of observed gene expression data, the genetic network inference problem may multiple optima. Therefore, to narrow down a optima and infer the correct genetic network, we used 15 sets of gene expression data, each covering all five genes. In a practical application, these sets of gene expression data could be obtained by actual biological experiments under different experimental conditions. A total of 11 sampling points for the gene expression data were assigned on each gene in each set according to previous work[6].

Because the objective function differ between the S-system model method and the PUNN model method, we used the parameter error defined as equation (15) instead of the objective value for the discussion of experimental results.

$$PE(\mathbf{P}) = \sum_{i=1}^{N} \left(|\alpha_i - \alpha_i^t| + |\beta_i - \beta_i^t|\right) + \sum_{i=1}^{N}\sum_{j=1}^{N} \left(|g_{i,j} - g_{i,j}^t| + |h_{i,j} - h_{i,j}^t|\right) \tag{15}$$

Here, $\mathbf{P} = \{\alpha_i, \beta_i, g_{i,j}, h_{i,j}\}(i,j = 1, \cdots, n)$, and α_i^t, β_i^t, $g_{i,j}^t$, $h_{i,j}^t$ are S-system parameter of target genetic network. Note that $\sum \left(|\alpha_i^t| + |\beta_i^t|\right) + \sum\sum \left(|g_{i.j}^t| + |h_{i,j}^t|\right) = 114$. In the S-system model method and the PUNN model method, the number of individuals was 60, and N_{L1} was 20. The experiment is executed in AMD Althorn(tm) 64 X2 Dual Core Processor 3800+ 2.01 GHz, 1.00GB.

B. How to estimate the differential coefficient of gene expression level

The PUNN model method infers the genetic network using estimated differential coefficient of observed gene expression level as teacher signal. It is considered that the effectiveness of the PUNN model method has a connection with accuracy of estimated differential coefficient of observed gene expression level. To confirm this relationship, we used three kind of estimated differential coefficient of observed gene expression level; (1) the calculated differential coefficient from interpolated gene expression data using spline interpolation, (2) the calculated differential coefficient from correct gene expression data and (3) the correct differential coefficient. (1)'s differential coefficient is able to obtain from only observed gene expression data. In contrast, (2)'s and (3)'s differential coefficient assumes that it is possible to obtain a detailed gene expression level and differential coefficient, respectively. Among those differential coefficients, (1)'s differential coefficient is the worst accuracy and (3)'s differential coefficient is the best accuracy. In this study, the PUNN model using each differential coefficient is called "PUNN model (interpolation)", "PUNN model (difference)" and "PUNN model (accuracy)", respectively. In this experiment, the (2)'s differential coefficient is calculated from the gene expression data before sampling, and the (3)'s differential coefficient is calculated from the correct gene expression data and target genetic networks.

C. Experimental Result

Table (I) shows the experimental results of 30 independent experiments. The CPU time means execution time to infer all S-system parameters, it does NOT mean execution time to infer only sub-problem. The ♯ of evaluation means the number of parameter evaluations, and this means the number of objective function evaluations. The error means "parameter error". In this experiment, we limited CPU times to 10 hours (each sub-problem was inferred in less than 2 hours). When the number of iteration exceeded 500, the S-system model method was unable to infer; therefore, it was omitted.

VI. Discussion

In the number of iteration is 500, the S-system model method infers in about 20000 seconds, and each PUNN model method infers in about 130 seconds. This result indicates that the PUNN model method is about 160 times faster than the S-system model method, and it has a significant effect that to eliminate the need to solving a differential equations.

In the parameter error, both average value and standard deviation value are obtained a better result by the PUNN model method, and it seems that the PUNN model method expects to obtain a better result than S-system model method. In the minimization parameter error, the S-system model method is obtained a better result than the PUNN model (interpolation) method. However, compared with the result in the previous works shown in table (II), the result obtained by the PUNN model (interpolation) method is much better.

In the relationship between the parameter error and the accuracy of estimated differential coefficient on the PUNN model method, the PUNN model (interpolation) is obtained the worst parameter error, and the PUNN model (accuracy) is obtained the best parameter error. This sequence is same order of the accuracy of estimated differential coefficient. It seems that this result is generated by "over-fitting". The PUNN model tries to obtain the S-system parameter which represents the observed differential coefficient. If the observed differential coefficient is incorrect, the S-system parameter, which is tried to obtain, differs from S-system parameter in target genetic networks. This difference becomes large corresponding to inaccuracy of observed differential coefficient.

VII. Conclusion

In this study, we proposed the PUNN model to solve the genetic network inference problem. The PUNN model method infers the S-system model of genetic networks by leaning to the PUNN described the S-system model of genetic network. Therefore, the PUNN model method is possible to eliminate the need to solve the differential equations, and infers the S-system model of genetic networks at short times. The experimental results are as follows that 1) the PUNN model method is expected to obtain the better S-system parameter than the S-system model method, 2) the PUNN model method is about 160 times faster than S-system model method, and 3) the performance of the PUNN model method advances by

TABLE I
THE RESULTS OF INFERENCE OF GENETIC NETWORKS IN EACH MODEL.

Number of iteration	model	CPU time[s]	♯ of evaluation	Error(ave.)	Error(SD)	Error(min.)
500	S-system model	19891.73	1518420	6.2197	14.7249	0.0571
	PUNN model (interpolation)	123.43	1251817	3.2900	1.6494	1.1332
	PUNN model (difference)	132.08	1249353	2.3724	2.0382	0.2476
	PUNN model (accurate)	123.38	1253127	2.1277	1.6807	0.2361
1000	PUNN model (interpolation)	181.15	2092374	2.6717	1.5272	1.1284
	PUNN model (difference)	181.66	1947153	1.5281	1.8697	0.0701
	PUNN model (accurate)	173.98	2035241	1.5609	1.8026	0.0624
5000	PUNN model (interpolation)	667.95	9034026	1.8170	0.8323	1.1867
	PUNN model (difference)	571.29	7398901	0.3022	0.8187	0.0118
	PUNN model (accurate)	562.43	6890762	0.6055	1.3121	0.0000
10000	PUNN model (interpolation)	1317.66	18064577	1.6636	0.4853	1.1972
	PUNN model (difference)	1102.35	14045943	0.2909	1.2666	0.0118
	PUNN model (accurate)	1017.19	12804644	0.2889	0.8469	0.0000
50000	PUNN model (interpolation)	6301.80	90134959	1.5278	0.1054	1.2124
	PUNN model (difference)	4727.22	60929488	0.0289	0.0317	0.0118
	PUNN model (accurate)	4728.00	59369760	0.0190	0.0401	0.0000

TABLE II
THE ESTIMATED GENETIC NETWORKS IN THE PREVIOUS WORKS.

Author	computer	CPU time[s]	Error
Kimura[6]	Pentium 3, 1GHz	17640	2.463
Noman[7]	Pentium, 1.7GHz	18000	0.724

$*$The error with the best parameters in each study is shown.

improvement accuracy of estimated differential coefficient of gene expression data.

In this study, we only considered the genetic network which is consist five genes, and the gene expression data without measurement error. Therefore, we are going to apply to the large genetic networks and the gene expression data with measurement error. In addition, we are going to improve accuracy of estimated differential coefficient of gene expression level, to advance the performance of the PUNN model method.

REFERENCES

[1] H. D. Jong, "Modeling and simulation of genetic regulatory system: a literature review, " Journal of Computational Biology, Vol. 9, No. 1, pp.67 – 103. (2002)

[2] J. Mason, P. S. Linsay, J. J. Collins and L. Glass, "Evolving complex dynamics in electronic models of genetic networks," Chaos, Vol. 14, No. 3, pp. 707 – 715. (2004)

[3] M. A. Savageau, "Biochemical systems analysis. II The steady state solution for an n-pool system using a power-law approximation," Journal of Theoretical Biology, Vol. 25, pp. 370 – 379. (1969)

[4] S. Kimura, K. Sonota, S. Yamane, K. Matsumura, and M. Hakakeyama, "Proposing an Identification Method for a Neural Network Model of Genetic Networks," Information Processing Society of Japan Digital Courier, Vol.3(2007), pp.153–163. (2007)

[5] R. Durbin and D. Rumelhart, "Product units: a computationally powerful and biologically plausible extension to backpropagation networks," Neural Computation, Vol.1, No.1, pp.133 – 142. (1989)

[6] S. Kimura, M. Hatakeyama and A. Konagaya, "Inference of S-system Models of Genetic Networks from Noisy Time-series Data," Chem-Bio Informatics Journal, Vol. 4, No. 1, pp.1-14. (2004)

[7] N. Noman and H. Iba, "Inference of Genetic Networks using S-system: Information Criteria for Model Selection," Proc. of GECCO-2006 , pp.263-270. (2006)

[8] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi and M. Tomita, "Dynamic modeling of genetic networks using genetic algorithm and S-system," Bioinformatics, Vol. 19, No. 5, pp.643 – 650. (2006)

[9] A. P. Engelbrecht and A. Ismail, "Training Product Unit Neural Networks," Stability and Control: Theory and Applications, Vol. 2, No. 1–2, pp. 59 – 74. (1999)

[10] A. Ismail and A. P. Engelbrecht, "Global Optimization Algorithms for Training product Unit neural networks, " IEEE Computer society, Vol. 1, pp. 132 – 137. (2000)

[11] Y. Maki, T. Ueda, M. Okamoto, N. Uematsu, Y. Inamura and Y. Eguchi, "Inference of genetic network using the expression profile time course data of mouse P19 cells," Genome Informatics, Vol. 13, pp. 382 – 383. (2002)