Finding Original Regulatory Networks with Weight Matrices

Nobuhisa UEDA † Taisuke SATO †
ueda@mi.cs.titech.ac.jp sato@cs.titech.ac.jp
† Dept. of Computer Science, Tokyo Institute of Technology

Abstract: We propose a new method to find regulatory networks with weight matrices from expression patterns. It estimates parameters in the network with a real-coded genetic algorithm called UNDX, finds structures with the random-restart hill-climbing search, and evaluates their fitness with an MDL-based fitness function. We also show experimental results using this method. In experiments, we succeeded in identifying a structure which generated the expression patterns used as data sets, while existing methods failed to discover the original one.

1 Introduction

Analyzing gene interactions is receiving growing attention in computational biology since it becomes possible to obtain expression data from DNA microarrays. For modeling interactions, various network models are proposed such as boolean networks [10, 3, 1], weighted network models [6, 4], and regulatory networks with weight matrices [11, 5].

Regulatory networks with weight matrices are able to generate expression patterns. Thus, once we find a regulatory network with a weight matrix from expression patterns, other expression patterns under various conditions, e.g., one s.t. transcriptions of some genes are disabled, are predictable by using the network. However, it seems difficult to find structures of regulatory networks with weight matrices generating original expression patterns. For example, some existing method [5] could not even select the original structure uniquely with 3 interacting genes from 3 expression patterns.

In this paper, we propose a new method to search for network structures generating the given expression patterns, and show experimental results in which our method successfully recovered the original one.

2 Preliminaries

$u_i(t)$ and $s_i(t)$ stand for *an expression* and *a regulation* of gene i at step t respectively. Let *an expression state* at step t be the set of expressions of all genes at step t, and *an expression pattern* a set of expression states. The length of an expression pattern is the number of expression states in the expression pattern.

We will henceforth call regulatory networks with weight matrices [11] or gene regulatory networks [5] just regulatory networks for short. A regulatory network $G = (V, E, W)$ is a weighted, directed graph with a set of vertices (genes) $V = \{1, \ldots, n\}$, a set of directed edges (or a structure of the regulatory network) $E = \{(i, j) | i, j \in V\}$, and a weight matrix $W = \{w_{ij}\}_{n \times n}$ ($w_{ij} \in \mathbf{R}$) where w_{ij} represents a weight of $(i, j) \in E$ and $(i, j) \in E$ iff $w_{ij} \neq 0$. In what follows, an original structure of an expression pattern means a structure which generates the expression pattern.

In this paper, we deal with network models in which regulations take continuous values, not discrete ones such as [6, 4]. For instance, the TReMM model was proposed by Weaver et al. [11], and another model, which we will call hereafter the MK model, was presented by Morohashi and Kitano [5].

In these models, given a regulatory network and initial expression states, expressions $u_i(t)$ $(t = 2, \ldots, T)$ for a gene i are calculable from previous expression states $u_j(t-1)$ $(j = 1, \ldots, n)$. Procedurally, a regulation $s_i(t-1)$ is computed as a sum of $u_j(t-1)$ weighted by the given weight matrix, and $u_i(t)$ is then calculated as a value of functions of $s_i(t-1)$.

2.1 The TReMM Model

The TReMM model defines a regulation $s_i(t)$ and an expression $u_i(t)$ as

$$s_i(t) = \sum_{j=1}^{n} w_{ji} u_j(t), \; u_i(t+1) = \frac{m_i}{1 + e^{-s_i(t)}} \qquad (1)$$

where m_i is a constant denoting the maximum expression of a gene i.

We consider here estimation of weight matrices. Note that, from eq.(1), the following relation holds between $u_i(t+1)$ and $s_i(t)$:

$$s_i(t) = -\ln\left(\frac{m_i}{u_i(t+1)} - 1\right).$$

Hence, given an expression pattern, we can calculate

$s_i(t)$. For notational convenience, we define

$$U=\begin{pmatrix} u_1(1) & \cdots & u_n(1) \\ \vdots & \ddots & \vdots \\ u_1(T-1) & \cdots & u_n(T-1) \end{pmatrix},$$

$$w_i=\begin{pmatrix} w_{1i} \\ \vdots \\ w_{ni} \end{pmatrix}, s_i=\begin{pmatrix} s_i(1) \\ \vdots \\ s_i(T-1) \end{pmatrix}.$$

Then $Uw_i = s_i$ holds from eq.(1). By using a generalized inverse matrix U^+ [8], we estimate $\tilde{w}_i = U^+ s_i$.

In order to take structures as well as weight matrices of regulatory networks into account, Weaver et al. proposed a method called REM (Reverse Engineering of Matrices) [11] in figure 1. For each gene i, REM prepares n edges s.t. every gene (vertex) has an outgoing edge into i, and repeats the following three steps: (1) estimating weights of the prepared edges, (2) calculating an euclidean error between a given expression pattern and an estimated one with the weights, (3) removing an edge which has the smallest weight absolute value in the edges. This continues until i has no incoming edge. Then, weights with the minimum euclidean error are selected as output.

procedure REM(U,m,n);
$E = \{(i,j)|1 \le i,j \le n\}$; $E_{prev} := E$;
$s_i(t) := -\log(\frac{m_i}{u_i(t+1)} - 1)$ $(1 \le i \le n, 1 \le t \le T-1)$
$s_i := (s_i(1), \ldots, s_i(T-3))^T$;
$U := \{u_{t,i}\}_{(T-3)\times n}$; $U' := \{u_{T-1,i}\}_{1\times n}$;
for $j := 1$ **to** n;
  $Err' = \infty$; $X := U$;
  while $(E \cap \{(k,j)|1 \le k \le n\}) \neq \emptyset$
    find a generalized inverse matrix X^+
      for $Xw_j = s_j$; % $w_j = (w_{1j}, \ldots, w_{nj})^T$;
    $\tilde{w}_j := X^+ s_j$; $s'_j := U'\tilde{w}_j$; $Err = |s_j(T-1) - s'_j|$;
    if $Err' \ge Err$ **then**
      $E_{prev} := E$; $w_j := \tilde{w}_j$; $Err' := Err$;
    find k s.t. $w_{kj} = \min_{k'} w_{k'j}$;
    $E := E - \{(k,j)\}$; $x_k(t) := 0$ $(1 \le t \le T)$;
  end-while;
  $E := E_{prev}$;
end-for;
$W := (w_1, \ldots, w_n)$;
output E, W;

Figure 1: REM

2.2 The MK model

In the MK model, a regulation $s_i(t)$ and an expression $u_i(t)$ are defined as

$$s_i(t) = \sum_{j=1}^{n} w_{ji} u_i(t) - h_i, \; u_i(t+1) = F[s_i(t)] + D \cdot u_i(t)$$

where h_i is a threshold for a gene i, D $(0 \le D \le 1)$ is the decay rate of expression states at the previous step, and

$$F(s) = \begin{cases} 0 & s < 0, \\ s & 0 \le s < 1, \\ 1 & s \ge 1. \end{cases}$$

In estimation of weight matrices from expression patterns, general inverse matrices cannot be employed like REM. The reason is that $F(s)$ is not a one-to-one function, and then $s_i(t)$ is not calculable from $u_i(t+1)$. Hence, a genetic algorithm was adopted to estimate weight matrices in [5].

In the MK model, Morohashi and Kitano proposed a method called the *in silico* sampling and screening in figure 2. This method aims at extracting a group of plausible structures from possible ones for given expression patterns, and requires a threshold for each expression pattern.

3 Proposed method

This section describes our method which is applicable to the TReMM model and the MK model.

First, for estimating weight matrices, a real-coded genetic algorithm called UNDX [7] is applied, since several real-coded GAs are reported to show higher performance in function optimization than GAs based on binary or Gray representation.

Next, in order to avoid structures with redundant edges of regulatory networks, the minimum description length principle (MDL principle) [9] is used, and an MDL-based fitness function f is introduced as follows:

$$f(U, \tilde{U}, E, T) = \frac{nT}{2} \log \frac{\sum_{i=1}^{n}\sum_{t=1}^{T}(u_i(t) - \tilde{u}_i(t))^2}{nT} + \frac{k+1}{2} \log nT,$$

where n is the number of genes, T is the length of a given expression pattern, $u_i(t)$ and $\tilde{u}_i(t)$ are a expression of a gene i at step t in the expression pattern and one calculated with the estimated weight matrix, and k is the number of edges in the regulatory network. We suppose the best structure is the one with the minimum

procedure iss $(U_0, U_1, \ldots, U_l, Th_0, Th_1, \ldots, Th_l)$;
$\mathcal{G}$:=sampling(U_0,Th_0);
for $i := 1$ **to** l;
  $\mathcal{G}_i$:=screening($U_i, Th_i, \mathcal{G}$);
end-for
output $\mathcal{G}_1 \cap \cdots \cap \mathcal{G}_l$;

function sampling(U,Th);
$\mathcal{E} := \{E | E$ is a possible structure for $U\}$; $\mathcal{G} := \emptyset$;
for each structure $E \in \mathcal{E}$;
  estimate W for U and E using a genetic algorithm;
  calculate U' with W;
  TSS :=the value of total sum square error
    between U and U';
  if $TSS \leq Th$ **then** $\mathcal{G} := \mathcal{G} \cup \{(E, W)\}$;
end-for
return $\mathcal{G}$;

function screening(U,Th,$\mathcal{G}$);
$\mathcal{G}' := \emptyset$;
for each $(E, W) \in \mathcal{G}$
  calculate U' with W where W is modified
    to reflect mutant expression states U;
  TSS :=the value of total sum square error
    between U and U';
  if $TSS \leq Th$ **then** $\mathcal{G}' := \mathcal{G}' \cup \{(E, W)\}$;
end-for
return $\mathcal{G}'$;

Figure 2: The *in silico* sampling and screening

fitness value determined by this function. This function was originally derived for fitting polynomials [2], not for estimating structure of regulatory networks.

Lastly, the hill-climbing search is used for finding candidate structures of regulatory networks. The reason is that the number of all possible structure of networks is 2^{n^2}, and it is intractable to evaluate fitness of all structures when n is large. Search should restart several times from randomly initialized structures in order to escape trivial local minima.

The proposed method is summarized as in figure 3, where g_1 and g_2 are functions from expressions to regulations and from regulations to expressions respectively.

procedure random-restart_hill-climbing(U,n,k);
$Fit_{best} = \infty$;
for $h := 1$ **to** k
  initialize E_{prev} randomly; $Fit_{prev} := \infty$;
  while true
    $Fit' := \infty$;
    for all l, m s.t $1 \leq l, m \leq n$;
      E :=neighbor(E_{prev}, l, m);
      estimate a weight matrix W
        with UNDX s.t. $w_{ij} = 0$ if $(i, j) \notin E$;
      $\tilde{u}_i(1) := u_i(1)$ $(1 \leq i \leq n)$
      for $t := 1$ **to** $T - 1$;
        $s_i(t) := g_1(W, \tilde{u}_j(t), \ldots, \tilde{u}_n(t))$: $(1 \leq i \leq n)$
        $\tilde{u}_i(t+1) := g_2(s_i(t))$; $(1 \leq i \leq n)$
      end-for;
      $Fit := f(U, \tilde{U}, E, T)$; % f: fitness function
      if $Fit < Fit'$ **then**
        $Fit' := Fit$; $E' := E$; $W' := W$;
    end-for;
    if $Fit' < Fit_{prev}$ **then**
      $Fit_{prev} := Fit'$; $E_{prev} := E'$; $W_{prev} := W$;
    else break;
  end-while;
$Fit(h) := Fit_{prev}$; $E(h) := E_{prev}$; $W(h) := W_{prev}$;
end-for
$h' := \arg\min_h Fit(h)$;
output $E(h'), W(h')$;

function neighbor(E,i,j);
if $(i, j) \in E$ **then** $E := E - \{(i, j)\}$;
else $E := E \cup \{(i, j)\}$;
return E;

Figure 3: Proposed method

4 Experimental Results

In this section, we provide empirical evidence that the proposed method is able to find the original structure from the expression patterns. Experiments are carried out using the TReMM model and the MK model respectively.

First we show a result in the TReMM model. The result is based on an artificial data set, and it was calculated from a regulatory network in figure 4. The original structure was taken from [5] as an example of a regulative interaction in biological models. The initial expressions were $u_1(1) = 0.5, u_2(1) = 0.1$ and $u_3(1) = 0.1$, and the maximum expression m_i was 5.0 $(1 \leq i \leq 3)$. With this initial condition, the given regulatory network generated expressions $u_i(t)$. Noise $\epsilon_{i,t}$ was added to $u_i(t)$ after all expressions were generated, where $\epsilon_{i,t} \sim N(0, 0.04)$ was a normally distributed random number, and the length of the expression pattern was 10. The generated expression pattern is plotted in figure 4.

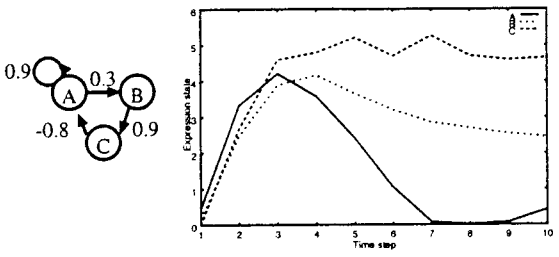
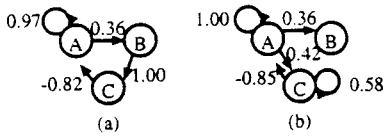


Figure 4: The original structure and the expression pattern

Next, we explain the setting of this experiment. In the proposed method, search of structures was carried out 10 times with randomly initialized structures. In UNDX, the number of applying crossover in one generation was 300, the population size was 300, the number of total generations was 200, and α and β were set to 0.5 and 0.35 respectively where α and β were parameters in UNDX specifying variances of random numbers used for generating children.

Two structures were found in this experiment as shown in figure 5 with 10 trials of search, where the structure (a) was found 6 times, and the structure (b) 4 times. The values of both fitness and the mean square errors (MSEs) were also depicted. The proposed method returned the structure (a) since its fitness value was best, and hence the original structure was identified.

From the result of this experiment, it seems sufficient to use the mean square error as a fitness function. This MDL-based fitness function, in fact, worked effectively in the hill-climbing search. For example, figure 6 is a



Structure	Fitness	MSE
(a)	-48.78	0.022
(b)	-46.50	0.023

Figure 5: Regulatory networks found with the hill-climbing search

sequence of structures at some trial of search with the minimum fitness value. The values of the MSEs and the fitness are also shown. In some transitions, e.g., between the first iteration and the second one, values of the MSEs increase but those of fitness still decrease. In these transitions, if the MSEs were employed, search would stop before finding the original structure.

For comparison, we show a structure found with REM [11] in figure 7, from the expression pattern. Compared with the original structure, the discovered one contains redundant edges. In this case, REM failed to find the original structure.

We also have conducted an experiment using the MK model with a regulatory network and an expression pattern in figure 8. This expression pattern was generated under the same condition as in the experiment in the TReMM model, where threshold h_i is 0.0 for all i, the decay rate D 0.8, and the length of the expression pattern 50.

The setting of this experiment was the same as the previous experiment except the number of total generations in UNDX, which was 500. As a result of the experiment, four structures in figure 9 were found from 10 randomly initialized structures, where the structure (a) was found 7 times, and the other structures once respectively. The fitness values and the MSEs of these structures are also set out in the figure. The proposed method found the structure (a) as its fitness value was best, and the original structure was successfully identified.

Lastly, we compare the proposed method with the *in silico* sampling and screening, proposed for finding original structures in the the MK model [5]. It has been reported that, for the same original structure in this experiment, the *in silico* sampling and screening selected 283 structures from an expression pattern without noise, and 8 structures including the original one from the expression pattern and two additional expression patterns [5]. In contrast to this, the proposed method was able to find the original structure from only one expression pattern with noise.

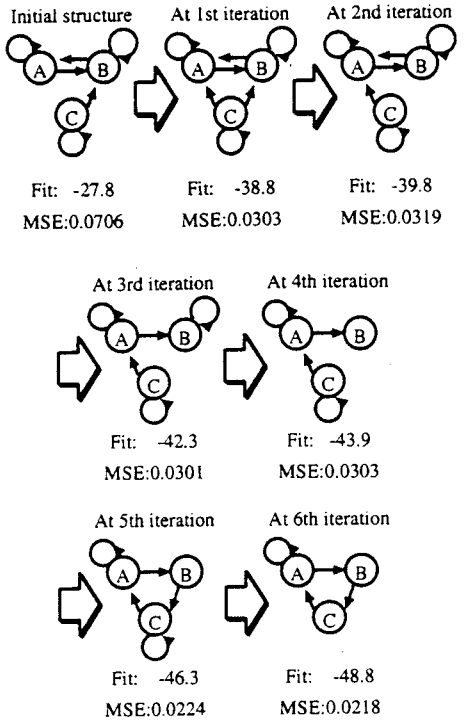


Figure 6: A sequence of structures with the minimum fitness values at iterations

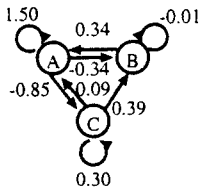


Figure 7: Regulatory networks found with REM

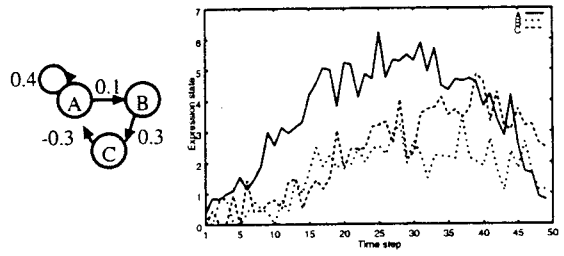
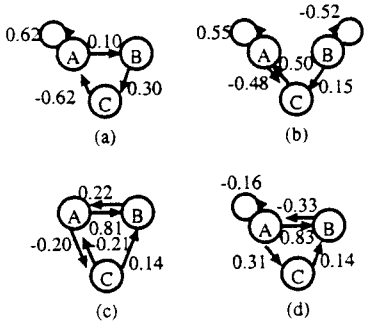


Figure 8: The original structure and the expression pattern



Structure	Fitness	MSE
(a)	-94.58	0.240
(b)	-73.94	0.305
(c)	-58.81	0.374
(d)	-55.95	0.388

Figure 9: Regulatory networks found with the hill-climbing search

5 Discussion

We have proposed a new method to find regulatory networks with weight matrices from expression patterns, and have shown experimental results in which the proposed method has successfully identified a structure with 3 genes that generated the expression patterns.

On the other hand, the proposed method may not find original regulatory networks with dozens of genes. One reason is that, we found experimentally numerous different structures which fit a noisy expression pattern better than the original one. We carried out an experiment with 20 randomly generated structures with 20 genes on the TReMM model, where expression patterns contain at most 5% noise. We discovered, for a noisy expression pattern, about 200^{20} structures on average whose errors are smaller than those of the original structures.

Acknowledgments

This work is supported in part by the Grand-in-aid for Scientific Research on Priority Areas (Discovery Science) 2000 of the Ministry of Education, Science, Sports, and Culture, Japan.

References

[1] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano. Identification of gene regulatory networks by strategic gene disruption and gene overexpressions. In *Proceedings of the Ninth*

Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 695–702 (1998).

[2] T. S. Han and K. Kobayashi. *Mathematics of Information and Coding* (in Japanese), Baifu-kan (1999).

[3] S. Liang, S. Fuhrman, and R. Somogyi. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures, In *Proceedings on Pacific Symposium on Biocomputing* 3, pp.18–29 (1998).

[4] T. Moriyama, A. Shinohara, M. Takeda, O. Maruyama, T. Goto, S. Miyano, and S. Kuhara. Finding genetic network from experiments by weighted network model. In *Proceedings of the Tenth Workshop on Genome Informatics* (1999).

[5] M. Morohashi and H. Kitano. Identifying gene regulatory networks from time series expression data by *in silico* sampling and screening. In *Proceedings of the fifth European Conference on Artificial Life* (ECAL'99), Lecture Notes in Artificial Intelligence 1674, pp.477–486 (1999).

[6] K. Noda, A. Shinohara, M. Takeda, S. Matsumoto, S. Miyano, and S. Kuhara. Finding genetic network from experiments by weighted network model. In *Proceedings of the Ninth Workshop on Genome Informatics*, pp.141–150 (1998).

[7] I. Ono and S. Kobayashi. A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover. In *Proceedings of the Seventh International Conference on Genetic Algorithms*, pp.246–253 (1997).

[8] C. R. Rao and S. K. Mitra. *Generalized inverse of matrices and its applications*. Wiley (1971).

[9] J. Rissanen. Modeling by shortest data description. *Automatica*, 14, pp.465–471 (1978).

[10] R. Somogyi and C. A. Sniegoski. Modeling the complexity of genetic networks: Understanding multigene and pleiotropic regulation. *Complexity*, 1, pp. 45-63 (1996).

[11] D. C. Weaver, C. T. Workman, and G. D. Stormo. Modeling regulatory networks with weight matrices. In *Proceedings on Pacific Symposium on Biocomputing* 4, pp.112–123 (1999).