



GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY  
Volume 11 Issue 21 Version 1.0 December 2011  
Type: Double Blind Peer Reviewed International Research Journal  
Publisher: Global Journals Inc. (USA)  
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

## Generation of Genetic Networks from a Small Number of Gene Expression Patterns under the Boolean Network Model

By Momotaz Begum, Sumaya Kazary, Md. Jakir Hossain, Sohag Kumar Bhadra, Md. Rokon Uddin

*Dept. of CSE, DUETGazipur, Bangladesh*

**Abstract** - There are lots of work for inferring genetic network architectures from state transition tables which correspond to time series of gene expression patterns, using the Boolean network model. Results of those computational experiments suggested that a small number of state transition (**INPUT/OUTPUT**) pairs are sufficient in order to infer the original Boolean network correctly. Tatsuya **AKUTSU**, Satoru **MIYANO** and Satoru **KUHARA** gave a mathematical proof for this. So there is possibility to devise an algorithm to generate all consistent genetic networks from a small number of gene expression patterns under the Boolean network model.

**General Terms** : *Gene network, gene regulatory network, Genetic network, gene expression pattern, Boolean network, microarray data, Sum of product, Boolean algebra.*

**Keywords** : *Generation, consistent network.*

**GJCST Classification** : *C.2.1*



GENERATION OF GENETIC NETWORKS FROM A SMALL NUMBER OF GENE EXPRESSION PATTERNS UNDER THE BOOLEAN NETWORK MODEL

*Strictly as per the compliance and regulations of:*



© 2011 . Momotaz Begum, Sumaya Kazary , Md. Jakir Hossain , Sohag Kumar Bhadra , Md. Rokon Uddin. This is a research/review paper, distributed under the terms of the Creative Commons Attribution-Noncommercial 3.0 Unported License (<http://creativecommons.org/licenses/by-nc/3.0/>), permitting all non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

# Generation of Genetic Networks from a Small Number of Gene Expression Patterns under the Boolean Network Model

Momotaz Begum<sup>α</sup>, Sumaya Kazary<sup>Ω</sup>, Md. Jakir Hossain<sup>β</sup>, Sohag Kumar Bhadra<sup>ψ</sup>, Md. Rokon Uddin<sup>ξ</sup>

**Abstract** - There are lots of work for inferring genetic network architectures from state transition tables which correspond to time series of gene expression patterns, using the Boolean network model. Results of those computational experiments suggested that a small number of state transition (INPUT/OUTPUT) pairs are sufficient in order to infer the original Boolean network correctly. Tatsuya AKUTSU, Satoru MIYANO and Satoru KUHARA gave a mathematical proof for this. So there is possibility to devise an algorithm to generate all consistent genetic networks from a small number of gene expression patterns under the Boolean network model.

**General Terms** : Gene network, gene regulatory network, Genetic network, gene expression pattern, Boolean network, microarray data, Sum of product, Boolean algebra.

**Keywords** : Generation, consistent network.

## 1. INTRODUCTION

Inference of gene regulation mechanism from time series of gene expression patterns are getting more important especially due to the invention of DNA microarray technology. Expression profiles of several thousands of genes are now being produced for further analyses. Some methods have been proposed for the inference of gene regulation mechanism from time series of gene expression patterns.

A statistical method is proposed by Arkin, Shen and Ross. They used correlation matrices to infer chemical reaction networks from time series of measured concentration of species. Though they treated chemical reaction networks, they also suggested that their method might be applied to genetic networks. However, it seems difficult to apply their method to the inference of large scale networks.

A metabolic pathway is suggested to be inferred by DeRisi, Iyer and Brown from gene expression patterns of *Saccharomyces cerevisiae* obtained by using DNA microarrays. A network model similar to the Boolean network model is constructed by Yuh, Bolouri

and Davidson from time series of expression patterns relating to a sea urchin gene. But their inference methods are not systematic or automatic. Besides, some studies have been done on the inference of genetic networks from state transition data using the Boolean network.

On the other hand, Liang, Fuhrman and Somogyi proposed an algorithm named REVEAL for inference of Boolean networks (corresponding to genetic networks) from state transition tables (corresponding to time series of gene expression patterns). REVEAL used information theoretic principles in order to reduce the search space. They made some computational experiments on REVEAL. The results suggested that only a small number of state transition pairs (100 pairs from 1015) were sufficient for inferring Boolean networks with 50 nodes (genes) whose in degree (the number of input nodes to a node) was bounded by 3.

Tatsuya AKUTSU, Satoru MIYANO and Satoru KUHARA gave a mathematical proof for their observation. We will extend their algorithm to identify genetic networks from gene expression patterns derived by gene disruptions and gene over expressions using a Boolean network-like model. They proved mathematically a lower bound and an upper bound of the number of expression patterns required to identify the network correctly. In this paper we will try to extend try to provide an algorithm to generate all the consistent genetic networks from a small number of gene expression patterns under the Boolean network model.

## II. GENERATION APPROACH

### a) Genetic network

A **gene regulatory network** or **genetic regulatory network (GRN)** is a collection of DNA segments in a cell which interact with each other (indirectly through their RNA and protein expression products) and with other substances in the cell, thereby governing the rates at which genes in the network are transcribed into mRNA.

### b) Boolean Network

A Boolean network  $G(V;F)$  consists of a set  $V = \{v_1, v_2, \dots, v_n\}$  of nodes representing genes and a list  $F = (f_1, f_2, \dots, f_n)$  of Boolean functions, where a Boolean function  $f_i(v_{i_1}, v_{i_2}, \dots, v_{i_k})$  with inputs from specified nodes  $v_{i_1}, v_{i_2}, \dots, v_{i_k}$  is assigned to each node  $v_i$ . For a subset  $U$

Author <sup>α</sup> : Lecturer Dept. of CSE, DUETGazipur, Bangladesh.

E-mail : momotaz03\_duet@yahoo.com

Author <sup>Ω</sup> : Assitant professor Dept. of CSE, DUET Gazipur, Bangladesh.

E-mail : kazal\_duet@yahoo.com

Author <sup>β</sup> : CSE, DUET Gazipur, Bangladesh.

E-mail : newjakir@gmail.com

Author <sup>ψ</sup> : CSE, DUET Gazipur, Bangladesh.

E-mail : bhadra035@gmail.com

Author <sup>ξ</sup> : CSE, DUET, Gazipur, Bangladesh.

E-mail : rokon.duet@gmail.com

of  $V$ , an expression pattern of  $U$  is a function from  $U$  to  $\{0,1\}$ . An expression pattern of  $V$  is also called a state of a Boolean network. That is, represents the states of nodes (genes), where each node is assumed to take either 0 (not-express) or 1 (express) as its state value.

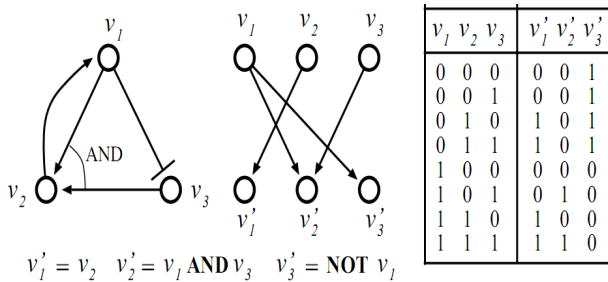


Figure 1(a) : A genetic network represented by Boolean network

c) Definition of Problem

Let  $(I_j;O_j)$  be a pair of expression patterns of  $\{v_1,v_2,\dots,v_n\}$ , where  $I_j$  corresponds to the INPUT and  $O_j$  corresponds to the OUTPUT. We call the pair  $(I_j;O_j)$  an example. We say that a node  $v_i$  in a Boolean network  $G(V;F)$  is consistent with an example  $(I_j;O_j)$ , if  $O_j(v_i)=f_i(I_j(v_{i1},\dots,I_j(v_{ik}))$  holds. We say that a Boolean network  $G(V;F)$  is consistent with  $(I_j;O_j)$  if all nodes are consistent with  $(I_j;O_j)$ . For a set of examples  $EX = \{(I_1;O_1),(I_2;O_2),\dots,(I_m;O_m)\}$ , we say that  $G(V;F)$  (resp. node  $v_i$ ) is consistent with  $EX$  if  $G(V;F)$  (resp. node  $v_i$ ) is consistent with all  $(I_j;O_j)$  for  $1 \leq j \leq m$ . Then, the problems are defined as:-

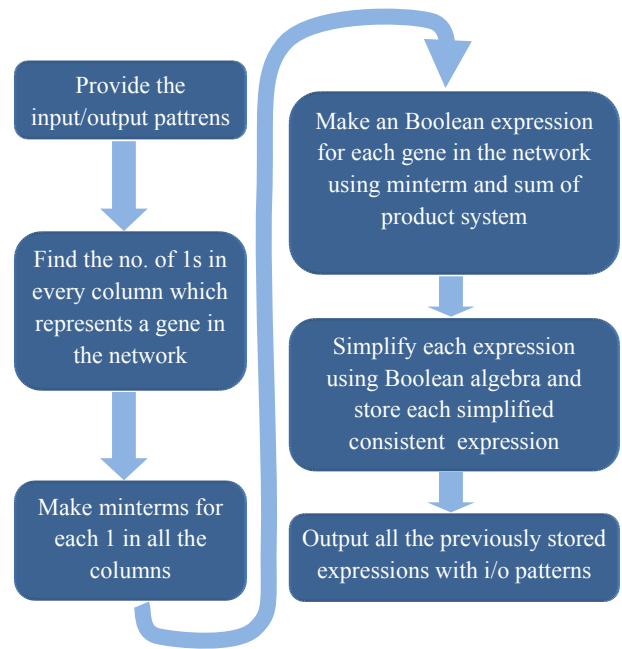
**Consistency** : Given  $n$  (the number of nodes) and  $EX$ , decide whether or not there exists a Boolean network consistent with  $EX$  and output one if it exists;

**Generation** : Given  $n$  (the number of nodes) and  $EX$ , generate all the number of Boolean networks consistent with  $EX$ .

	$v_1$	$v_2$	$v_3$	$v'_1$	$v'_2$	$v'_3$	
$I_1$	1	0	0	0	0	1	$O_1$
$I_2$	0	1	0	0	1	1	$O_2$
$I_3$	0	1	1	1	0	0	$O_3$
$G_1$	$v'_1 = v_2, v'_2 = v_2 \text{ AND } (\text{NOT } v_3)$						
	$v'_3 = \text{NOT } v_3$						

Table 1(a) : A genetic network with its input/output patterns and a consistent Boolean network

d) Process Development



e) Formulation of minterms

For a boolean function of  $n$  variables  $x_1, \dots, x_n$ , a product term in which each of the  $n$  variables appears once (in either its complemented or uncomplemented form) is called a minterm. Thus, a minterm is a logical expression of  $n$  variables that employs only the complement operator and the conjunction operator.

For example,  $abc, ab'c$  and  $abc'$  are 3 examples of the 8 minterms for a Boolean function of the three variables  $a, b$  and  $c$ . The customary reading of the last of these is  $a$  AND  $b$  AND NOT- $c$ .

There are  $2^n$  minterms of  $n$  variables, since a variable in the minterm expression can be in either its direct or its complemented form—two choices per  $n$  variables.

It is apparent that minterm  $n$  gives a true value (i.e., 1) for just one combination of the input variables. For example, minterm 5,  $a b' c$ , is true only when  $a$  and  $c$  both are true and  $b$  is false—the input arrangement where  $a = 1, b = 0, c = 1$  results in 1.

If one is given a truth table of a logical function, it is possible to write the function as a "sum of products". This is a special form of disjunctive normal form. For example, if given the truth table for the arithmetic sum bit  $u$  of one bit position's logic of an adder circuit, as a function of  $x$  and  $y$  from the addends and the carry in,  $ci$ :

Table 1(a) : Minterms formulation from a Boolean network

$ci$	$x$	$y$	$u(ci,x,y)$
0	0	0	0
0	0	1	1

0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Observing that the rows that have an output of 1 are the 2nd, 3rd, 5th, and 8th, we can write  $u$  as a sum of minterms  $m_1, m_2, m_4,$  and  $m_7$ . If we wish to verify this:  $u(ci, x, y) = m_1 + m_2 + m_4 + m_7 = (ci' x' y) + (ci' x y) + (ci x' y) + (ci x y)$  evaluated for all 8 combinations of the three variables will match the table.

On the other hand if we assign serial numbers to each row of the input expression patterns from 0 to  $2^n - 1$  then some of products can be expressed with summation notation ( $\Sigma$ ).

For example, for the above table as in output column the row 1,2,4 and 7 contain 1, so the output expression can be formulated as  $u(ci, x, y) = \Sigma(1,2,4,7)$ .

It is apparent that this notation can be simplified using the help of computer more easily and efficiently.

f) Algorithm

Suppose we have  $n$  number of nodes ( $v_1, v_2, \dots, v_n$ ) and a pair of expression pattern (I,O), where I correspond to input pattern and O corresponds to output pattern.  $V_1$  to  $v_n$  are assumed to take either 0 (not-express) or 1 (express) as its state value. First we generate a Boolean network consistent with (I,O) and then eliminate nodes which aren't essential to generate more networks. The idea is as follows:-

1. Repeat step 2 for each node  $v_i \in V$ .
2. Generate Boolean equation and assign in  $\hat{v}_i$  using Sum of Product (SOP) notation based on input and output expression patterns (I and O).
3. Repeat step 4 and 5 for each  $\hat{v}_i$ .
4. Eliminate variables  $v_i$  from  $\hat{v}_i$  using Boolean algebra until elimination is impossible.
5. This is a new Boolean network consistent with input and output expression patterns (I and O), store it.

g) Complexity

Consider the following expression pattern:-

Input			Output		
$v_1$	$v_2$	$v_3$	$\hat{v}_1$	$\hat{v}_2$	$\hat{v}_3$
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	1	0	0
0	1	1	0	0	1
1	0	0	0	0	0
1	0	1	1	1	0

1	1	0	0	0	1
1	1	1	0	0	1

Table 2(a) : Generation of Boolean network from input /output expression patterns

As here are 3 genes we have got 3 Boolean expressions for them. They can be formulated according to previous discussion .

They are listed below:-

$$\hat{v}_1 = \Sigma(3,5), \hat{v}_2 = \Sigma(0,5), \hat{v}_3 = \Sigma(0,3,6,7).$$

Using the above example first we'll calculate the running time of the propose 4d algorithm.

To generate an Boolean equation for any state first we have to check how many 1s it has in output pattern. It can have maximum  $2^n$  1s. So  $O(n * 2^n)$  is the running time for generation of each equation. As there are  $n$  nodes and one variable is associated with each node so running time of step two of the algorithm is  $O(n^2 * 2^n)$ .

Again for each node, there can be maximum  $2^n$  minterms and step 4 eliminates one minterm in each iteration. Now two minterms can be chosen in  $\binom{m}{2}$  ways (let  $m = 2^n$ ). If we reduce one minterm at a time then complexity of step 4 can be calculated as follows:-

In first iteration there are  $m$  minterms. So we can choose two of them to reduce into one in  $\binom{m}{2}$  ways. In second iteration there are  $m-1$  minterms. So we can choose two of them to reduce into one in  $\binom{m-1}{2}$  ways and so on. Similarly at the end we should have two minterms and we can choose them in one way. So the complexity will be

$$\begin{aligned} f(m) &= \binom{m}{2} + \binom{m-1}{2} + \binom{m-2}{2} + \dots + \binom{m-(m-3)}{2} + \binom{m-(m-2)}{2} \\ &= \binom{m+1}{3} \\ &= \frac{(m+1)!}{3!(m-2)!} \\ &= \frac{m^3 - m}{6} \\ &= \frac{(2^n)^3 - 2^n}{6} \\ &= \frac{8^n - 2^n}{6} \end{aligned}$$

This function expresses the complexity of one simplifying one equation. There are at most  $n$  equations.

So the overall complexity is given by  $\frac{8^n - 2^n}{6} * n$ .

So the running time of the algorithm is  $O(\frac{8^n - 2^n}{6} * n + n^2 * 2^n) = O(n * 8^n)$ .

### III. EXPERIMENTAL OUTCOMES

#### a) Performance Analysis

This paper improves the previous works done by the authors worldwide.

The improvements are listed below:-

- 1) It generates only the consistent network. So overhead of checking inconsistent networks is eliminated.
- 2) Huge improvement is achieved in running time.
- 3) As all consistent networks are generated, so counting problem is solved simultaneously.
- 4) The in degree/out degree have no limit. A node (gene) can be expressed by any number of nodes(genes).

### IV. CONCLUSION

We have proved mathematically that to generate all consistent Boolean networks it requires  $O(n \cdot 8^n)$  times. For that purpose, we proposed a simple algorithm. Of course, real biological systems are different from Boolean networks nodes in a Boolean network take binary values which are updated synchronously, whereas quantities of gene expressions in real cells are not binary and are changing continuously in time. However, owing to its simplicity, the proposed algorithm can be extended in various way. This algorithm can be extended to identify, count and enumerate all the consistent networks also.

Finally, we believe that our theoretical results, along with the mathematical computations encourage the attempts to discover the gene regulation mechanism from time series of gene expression patterns.

### REFERENCES REFERENCES REFERENCIAS

1. Tatsuya AKUTSU, Satoru MIYANO and Satoru KUHARA, Identification Of Genetic networks From A Small Number Of Gene Expression Patterns Under The Boolean Network Model.
2. S. Liang, S. Fuhrman and R. Somogyi, REVEAL, a general reverse engineering algorithm for inference of genetic network architectures, Pacif\_c Symposium on Biocomputing 3, 18 (1998).
3. Chia-Chin Wu, Hsuan-Cheng Huang, Hsueh-Fen Juan and Shui-Tein Chen, GeneNetwork: an interactive tool for reconstruction of genetic networks using microarray data
4. D.A. Kightley, N. Chandra, and K. Elliston, Inferring Gene Regulatory Networks from Raw Data: A Molecular Epistemics Approach, Pacific Symposium on Biocomputing 9:510-520(2004).
5. Rui Xu, Donald C. Wunsch II, and Ronald L. Frank, Inference of Genetic Regulatory Networks with Recurrent Neural Network Models Using Particle Swarm Optimization
6. T. Akutsu, S. Kuhara, O. Maruyama and S. Miyano, Identification of gene regulatory networks by

- strategic gene disruptions and gene over expressions, Proc. 9th ACM-IAM Symp. Discrete Algorithms, 695(1998).
7. A.Arkin,P. Shen and J. Ross, A test case of correlation metric construction of a reaction pathway from measurements, Science 277, 1275(1997).
8. J.L. DeRisi, V.R. Lyer and P.O. Brown, Exploring the metabolic and genetic control of gene expression on a genomic scale, Science 278, 680(1997).
9. M.J. Kearns and U.V. Vazirani, An Introduction to Computational Learning Theory, The MIT Press (1994).
10. H.H. McAdams and L. Shapiro, Circuit simulation of genetic networks, Science 269, 650 (1995).
11. R. Somogyi and C.A. Sniegoski, Modeling the complexity of genetic networks: Understanding multigene and pleiotropic regulation, Complexity 1, 45 (1996).
12. R. Thomas, D. Thie\_ry and M. Kaufman, Dynamical behaviour of biological regulatory networks -. Biological role of feedback loops and practical use of the concept of the loop-characteristic state, Bulletin of Mathematical Biology 57, 247 (1995).
13. Wikipedia ([http://en.wikipedia.org/wiki/Genetic\\_network](http://en.wikipedia.org/wiki/Genetic_network)), *the free encyclopedia*.
14. Donald E. Knuth, Concrete Mathematics
15. X. Wen et al, Large-scale temporal gene expression mapping of central nervous system development, Proc. Natl. Acad. Sci. USA 95, 334 (1998)
16. A.Wuensche, Genomic regulation modeled as a network with basins of attraction, Pacif\_c Symposium on Biocomputing 3, 89 (1998). 11. C-H. Yuh, H.
17. Bolouri and E.H. Davidson, Genomic Cis-regulatory logic: experimental and computational analysis of a sea urchin gene, Science 279, 1896 (1998).