# Reverse Engineering of a Gene Regulatory Network from Time-Series Data Using Mutual Information

Shohag Barman, Yung-Keun Kwon*
School of Electrical Engineering, University of Ulsan, 93, Daehak-ro, Nam-gu, Ulsan 680-749

**Abstract**

Reverse engineering of gene regulatory network is a challenging task in computational biology. To detect a regulatory relationship among genes from time series data is called reverse engineering. Reverse engineering helps to discover the architecture of the underlying gene regulatory network. Besides, it insights into the disease process, biological process and drug discovery. There are many statistical approaches available for reverse engineering of gene regulatory network. In our paper, we propose pairwise mutual information for the reverse engineering of a gene regulatory network from time series data. Firstly, we create random boolean networks by the well-known Erdős-Rényi model. Secondly, we generate artificial time series data from that network. Then, we calculate pairwise mutual information for predicting the network. We implement of our system on java platform. To visualize the random boolean network graphically we use cytoscape plugins 2.8.0.

## 1. Introduction

Identifying and better understanding of complex gene network is an importance research area in computational biology. Gene regulatory networks (GRNs) play a crucial role in the understanding of complex biological processes. Reverse engineering is a key issue for understanding of complex gene networks from time series data. Reverse engineering provides us deep understanding of gene interactions in a network. A GRN is consists of nodes such as genes, proteins, mRNA and their interactions (regulatory relationships). The regulatory relationship between genes can be activation (positive relation) or inhibition (negative relation). To discover the interaction structure between genes in GRN is our motivation. For example, there are two nodes without any connection that means no regulatory relationship between nodes. If we use reverse engineering algorithm to discover this interaction then we may find the idea of a cancer drug or AIDS drug etc. Reverse engineering of GRN is a challenging task. There are many reverse engineering algorithms proposed to describe the interactions among genes. Most of them have an inference problem because of the dimensionality (the number of genes are huge but the time points are few). There are many reverse engineering algorithms based on information theoretic approaches. REVEAL is the first information theoretic approach to reconstruct the gene regulatory relationship between genes in a network [1]. Another is TimeDelay-ARACNE which can recover the mechanism of time dependency between gene expression data [2]. In our paper, we propose a novel information theoretic approach called pairwise mutual information for reconstructing gene regulatory network. The main goal of our work is to reconstruct the network from real network and observe the structure of the network. It is important to choose a model for analyzing a GRN. A review paper [3] explains a number of models for inferring network such as Bayesian Networks [4], Dynamic Bayesian Networks [5], Boolean Networks[6], Probabilistic Boolean Networks [7], and Differential Equation Models [8]. In this paper we select the Boolean network model as it is very simple and easy to understand. Many researchers use biological gene expression dataset but in our work we use artificial time series dataset from RBN. We show how to infer the structure of gene network and find an interesting issue for reverse engineering of a network structure. Sometimes our mutual information algorithm can reconstruct the real network structure correctly. Moreover, we investigate the probabilistic nature based on incoming link in the network.

We organize the rest of the paper as follows: Section 2 describes the basic idea of Boolean network and Mutual Information. Section 3 focuses our proposed approach which includes generation of random Boolean network, creation of time series data form RBN, and calculation of mutual information using mutual information based feature selection algorithm. We describe our experiments in Section 4. Section 5 includes the conclusion and future Work.

## 2. Boolean Network and Mutual Information

The basic idea of Boolean network and mutual information is the key to understand the reverse engineering of GRN. This section contains the basic idea of Boolean network and mutual information.

### Random Boolean network

Kauffman [3] introduced Boolean networks in 1969 as a model for gene regulatory networks. A Boolean network G (V, F) consists of a set of nodes $V=\{V_1,V_2,.......,V_n\}$ representing genes and a set of boolean functions $F=\{f_1, f_2,......, f_n\}$ . A boolean function (used to update value) $f_i(v_{i_1},....,v_{i_k})$ with specific nodes $v_{i_1},....,v_{i_k}$ is assigned to each node $v_i$ .When a boolean network is used to model gene regulatory network, genes are represented by binary variable 0 and 1. The value 1 represents gene is “turn-on” or gene is expressed where the value 0 means the gene is “turn off” or not expressed. The state of a Boolean network is expressed as state vector of all nodes. In random Boolean network, nodes $v_i$ and $v_j$ has two regulatory relationship known as positive (“activation”) and negative (“inhibition”) if the nodes are connected by directed link. The binary value of variable $v_i \in V$ at time t + 1 is determined by the values of other variables $v_{i_1}, v_{i_2}, ..., v_{i_k}$ using a link to $v_i$ at time t by the following boolean function:

$$V_i(t+1) = f_i(V_{i_1}(t), ...., V_{i_k}(t))$$

Then all variables of the network are contemporaneously updated.

### Mutual Information

This paper proposes a method based on information theoretic approach which is mutual information. Shannon [9] introduced mutual information, a fundamental concept of information theory. To understand mutual information, we must first introduce the concept of entropy and joint entropy. The entropy H can be defined as the measure of uncertainty of a random variable X. Let X be a discrete random vector and consider a probability mass function p(x). The entropy H(x) of a discrete random variable X is defined as follows:

$$H(\mathrm{X}) = -\sum_{x \in \chi} P(\mathrm{x}) \log \mathrm{P(x)}.$$

Now let us consider two discrete random variables X and Y with a joint probability distribution p(x, y). The joint entropy H(X, Y) of these random variables can be defines as follows:

$$H(\mathrm{X, Y}) = -\sum_{x \in X} \sum_{y \in Y} p(\mathrm{x, y}) \log \mathrm{p(x, y)}.$$

Basically, the mutual information of two discrete variables measures the amount of information that one variable contains the information of the other one. In another words, we can define mutual information as the reduction in the uncertainty of one random variable due to the knowledge of the other. Let us consider two random variables X and Y with a joint probability distribution p(x, y) and probability mass distribution p(x) and p(y). The mutual information I(X; Y) of random variables can be defines as follows:

$$\mathrm{I(X, Y)} = \sum_{x \in X} \sum_{y \in Y} p(\mathrm{x, y}) \log \frac{p(\mathrm{x,y})}{p(\mathrm{x})\mathrm{p(y)}}.$$

## 3. Methods

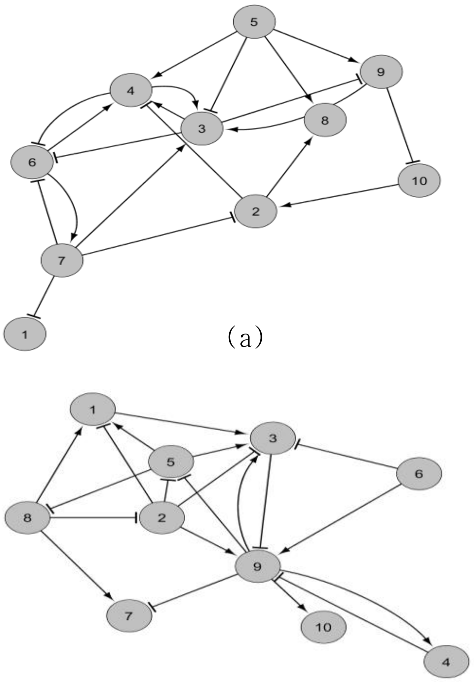
The overall process of our system is illustrated in Figure 2.

### Random Network Generation

The first step of our system is the generation of random boolean network. Erdős-Rényi is a well-known model for creating random network generation. Recently, this model has been widely used in many studies for analyzing biological network. Given the number of nodes (*N*) and a probability (*p*) that two arbitrary nodes are connected, we set some constraints for generating random network: (1) at least two nodes are needed for connectivity (2) every node must have at least one incoming edge and at least one outgoing edge as follows:

```
V←{1, …, N}; // V is the set of nodes
   E←[];  //  E is the set of edges
     For a:=1 to N
       For b:=1 to N
           If (a = b) continue;
           If (RandomNumber (0, 1) ≤ p)
           E←E∪{(a,b)};
       EndIf
      EndIf
     EndFor
    EndFor
// Random Number returns uniformly distributed a number over[0,1]
```

Figure 1 illustrates the different random boolean networks created by Erdős-Rényi model consisting of 10 nodes and 20 edges.



(a)

(b)

Figure 1. Random Boolean Network

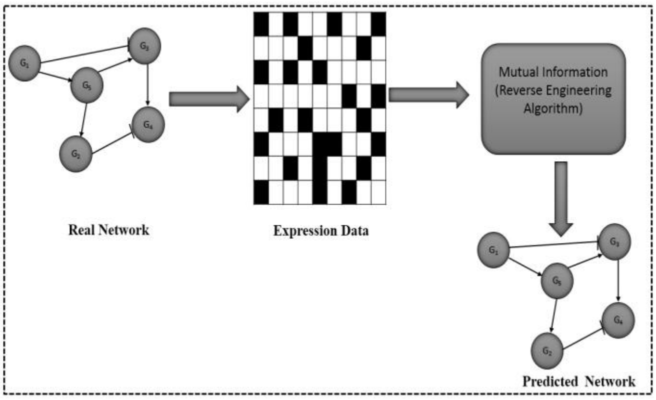
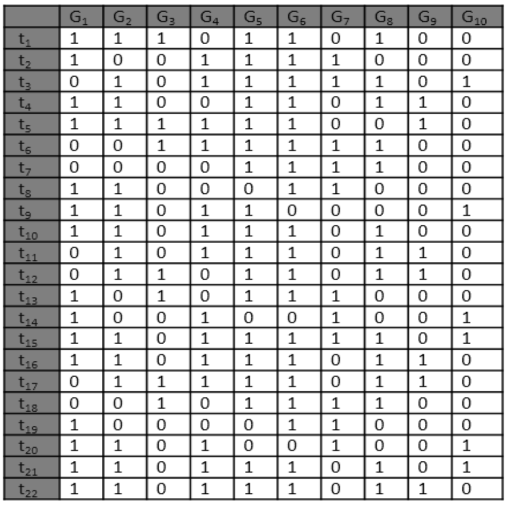


Figure 2. A conceptual view of reverse engineering of GRN

### Artificial Time Series Data Generation from RBN

The input of reverse engineering algorithm is the time series data. Mutual information algorithm takes the data as an input and outputs the predicted network as output. In our implementation we have generated our dataset according to the following rules:

1. Randomly choose an initial state (suppose $t_1$ in Figure 3).
2. Set the random update function and we consider only AND or OR boolean functions to every gene for updating.
3. Calculate the next state ($t_2$ in Figure 3).
4. Repeat step 3 until a steady state is found.



| | $G_1$ | $G_2$ | $G_3$ | $G_4$ | $G_5$ | $G_6$ | $G_7$ | $G_8$ | $G_9$ | $G_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $t_1$ | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| $t_2$ | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| $t_3$ | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| $t_4$ | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| $t_5$ | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| $t_6$ | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| $t_7$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| $t_8$ | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| $t_9$ | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| $t_{10}$ | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| $t_{11}$ | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| $t_{12}$ | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| $t_{13}$ | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| $t_{14}$ | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| $t_{15}$ | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| $t_{16}$ | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| $t_{17}$ | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| $t_{18}$ | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| $t_{19}$ | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| $t_{20}$ | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| $t_{21}$ | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| $t_{22}$ | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |

Figure 3. Time series dataset is generated using a Boolean network and a set of Boolean functions.

**Mutual Information-based Feature Selection Algorithm**

Mutual Information-based Feature Selection algorithm (MIFS) is used to select a feature or node based on time series data. In this step, we calculate pairwise mutual information for every gene pairs from microarray data using Battiti' s[10] MIFS algorithm. Here we use the time gap between target nodes and others nodes for calculating mutual information between them. For example, the target node is $G_1(t+1)$ and the other nodes are $G_2(t)$……$G_{10}(t)$(Figure 3). During calculation of pairwise mutual information, we didn't consider any self-loop.

The MIFS algorithm is as follows:

```
1. Set $F \leftarrow \{f_1, f_2, f_3, f_4, ..............., f_n\}$
       $S \leftarrow \phi$
2. Compute $I(\mathrm{C}; f)$ with the target node $\forall f \in F$.
3. Select the first feature that maximizes $I(C; f)$ , and set $F \leftarrow F \setminus \{f\}, \mathrm{S} \leftarrow \{f\}$.
4. Repeat until |S| = k
   (a) (computation of the MI between variables) for all couples of variables (f, s) with $f \in F$ , $s \in S$ compute $I(f; \mathrm{s})$ , if it not already selected.
   (b) (Selection of the next feature or next source node) chose feature $f \in F$ as the one that maximizes $I(C; f) - \sum_{s \in S} I(f; s)$ ; set $F \leftarrow F \setminus \{f\}$; set $S \leftarrow S \cup \{f\}$
5. Output S containing the selected features or solutions.
```

## 4. Results

After applying the MIFS algorithm we get the predicted network or reverse engineering network. We have conducted 100 random Boolean network. In figure 4, we showed the result of 50 random boolean network and the other 50 network result is shown in figure 5. Our proposed algorithm can reconstruct or infer many real network correctly. We find the interesting result based on the incoming link of the network. We observed the probabilistic nature in the random network based on the incoming link. Our algorithm can reconstruct the structure accurately for incoming link k=1 and k=2 which is shown in the both simulation. If the incoming link of a network is increased then the accuracy starts to decrease. Table 1 and table 2 describes the test cases of incoming link. In the simulation, results showing the strength of our system.

To measure performance of our algorithm we have used precision, recall, and accuracy.

Precision is the percentage of inferred connections which are correct compare to true network.

$$precision = \frac{TP}{TP + FP} \ .$$

Recall is the percentage of true connection which are correctly inferred by the MIFS algorithm.

$$recall = \frac{TP}{TP + FN} \ .$$

Accuracy is the percentage of the number of correctly inferred connections to the number of connections in the real network.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \ .$$

Table 1. Incoming links and test cases of 50 RBN

| Incoming links, k | Test cases |
|---|---|
| **k=1** | 60 |
| **k=2** | 105 |
| **k=3** | 25 |
| **k=4** | 10 |

Table 2. Incoming links and test cases of another 50 RBN

| Incoming links, k | Test cases |
|---|---|
| k=1 | 65 |
| k=2 | 108 |
| k=3 | 20 |
| k=4 | 7 |

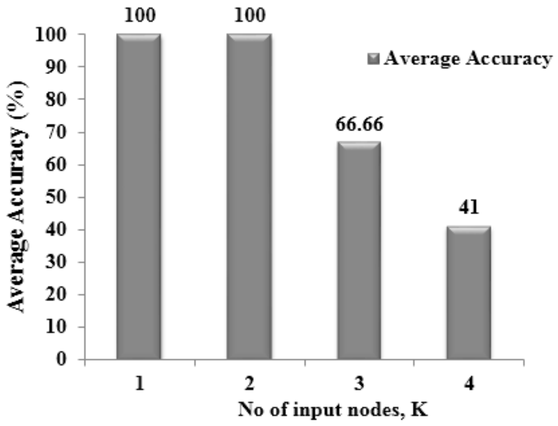


Figure 4. Average accuracy of incoming links: k=1, k=2, k=3 and k=4 for 50 RBN(Number of Nodes:10, Number of Edges: 20 )

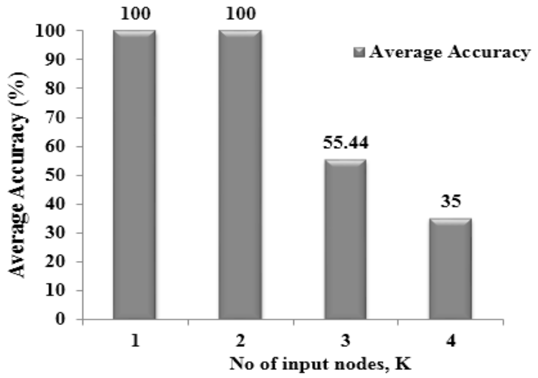


Figure 5. Average accuracy of incoming links: k=1, k=2, k=3 and k=4 for another 50 RBN (Number of Nodes: 10, Number of Edges: 20)

## 5. Conclusions:

We have used pairwise mutual information algorithm for reverse engineering of gene network. We have shown the structural behavior of GRN for reverse engineering. Our method shows the good accuracy result. In this paper, we have shown the gene interaction structure of a GRN on a small scale. We want to extend this work on a large scale network and also observe many incoming links. In future we will use a hybrid algorithm like combined mutual information and a genetic algorithm for improving the structure accuracy on a large scale gene network.

## References

[1] L. Shoudan, F. Stefanie, S. Roland. "REVEAL, a general reverse engineering algorithm for inference of genetic network architectures," in Pacific Symposiumon Biocomputing (World Scientific, Hawaii, 1998), p. 2

[2] Zoppoli, Pietro, S. Manella, and M. Ceccarelli. "TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach," Bmc Bioinformatics 11, no. 1 (2010): 154.

[3] H. Michael, L. Sandro, T. Susanne, Someren E. Van, R. Guthke. "Gene regulatory network inference: Data integration in dynamic models - A review,"

[4] T. Akutsu, S. Miyano, S. Kuhara. "Algorithms for inferring qualitative models of biological networks," Pacific Symposium on Biocomputing 2000, 4:17-28.

[5] K. Murphy, S. Mian. "Modelling gene expression data using dynamic Bayesian networks," In Technical report, Computer Science Division University of California, Berkeley, CA 1999.

[6] Stuart A. Kauffman. "Metabolic stability and epigenesis in randomly constructed genetic nets," J TheorBiol 1969, 22:437-467.

[7] I. Shmulevich, Edward R. Dougherty, Zhang W. "From boolean to probabilistic boolean networks as models of genetic regulatory networks," Proceedings of the IEEE 2002, 90(11):1778-1792.

[8] T. Chen, L. Hongyu, George M. Church. "Modeling gene expression with differential equations," Pacific Symposium Biocomputing 1999, 4:29-40.

[9] Thomas M. Cover, Joy A. Thomas. "Elements of Information Theory," John Wiley & Sons Inc., 2006.

[10] R. Battiti. "Using mutual information for selecting features in supervised neural net learning," IEEE Trans. Neural Network, vol. 5, no. 4, pp. 537–550, Jul. 1994.