ISLAMIC UNIVERSITY OF TECHNOLOGY

A Subsidiary Organ of the Organization of Islamic Corporation
Dhaka, Bangladesh

Determination Of Genetic Network From Time Series Gene Expression Data- A Modified Approach

Authors:

Hasan Md. Tusfiqur Alam (104409)
Nayreet Islam Rupak (104436)

Supervisor:

Tareque Mohmud Chowdhury
Assistant Professor
Department of Computer Science and Engineering

A thesis submitted to the Department of Computer Science & Engineering

In partial fulfilment of the requirements for the degree

B. Sc. Eng. in Computer Science &Engineering

Academic Year: 2013-2014

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the Analysis and investigation carried out by Hasan Md. Tusfiqur Alam and Nayreet Islam Rupak under the supervision of Tareque Mohmud Chowdhury in the Department of Computer Science and Engineering (CSE), IUT, Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

_____

Hasan MD. Tusfiqur Alam
Student ID – 104409

_____

Nayreet Islam Rupak
Student ID – 104436

_____

Supervisor:
Tareque Mohmud Chowdhury
Assistant Professor

Department of Computer Science and Engineering
Islamic University of Technology (IUT)

Abstract

Genetic Network is one of the most revolutionary discoveries in the field of Genetic Engineering. Gene regulatory networks control biological functions by regulating the level of gene expression. Discovering and understanding the complex causal relationships within gene networks has become a major issue in systems biology, computational biology and bioinformatics. The benefits of characterizing gene interaction are many, for example, Genetic networks provide knowledge about functional pathway in a given cell, representing processes such as metabolism, gene regulation, transport, and signal transduction , the effects of drugs on a regulatory pathway can be found, the development of cancer in a cell can be tracked, etc. Genes are the building blocks of a body. Genetic code directs functional property of every living organism. Genes directly encode proteins that make up the cell to function properly. At first DNA is converted into a mature messenger RNA (mRNA). Then mRNA is read and converted into amino acid sequence. The information contained in the nucleotide sequence is read as three –letter word called codon. Now amino acids coded by codon together form a polypeptide chain that is later folded into protein. Few proteins are parked into promoter region of another protein and performs various jobs like turn it on or off, regulate the protein production rate etc. Thus we can say that each gene here is responsible for influencing other gene or it might influence itself. For this reason expression level of the working genes always changes with time. DNA microarray experiments today allow to monitor the output of gene regulatory networks by measuring the gene expression levels of thousands of genes. Our primary focus on this paper is to find out methods for finding out those sets of genes that have some contribution for the growth of a bacteria called ***'Burkholderia Pseudomalli'*.** At various phases of the growth of ***'Burkholderia Pseudomalli'*** we performed computation using Microarray gene expression time series dataset. The dataset was obtained from GEO data base of NCBI website. Initially dataset contained information about 5289 genes in 47 consecutive time.

The entire work was divided into two phases. The first phase was data reduction as performing computation with this huge sizes of the microarray data is a pressure hardware of the computers as well as it is very much time consuming. So a data reduction methodology was applied which finds out the responsible genes actively taking part in the overall bacterial growth process or we can say that the dominant genes responsible for the growth was found out. The second phase was formation of genetic network from genetic dependencies in various time series. Finally genetic network was of those genes that are responsible for the growth.Once genetic network was determined this network can be used to study various unknown biological process, metabolic pathway engineering, drug discovery etc.

Contents

List of Figures:

List of Tables:

Chapter 1

Introduction

1.1 Overview

The hereditary characteristics of every living organism is defined by its genome. Genome is actually long sequence of DNA which is responsible to construct an organism. Now each genome can be divided into series of DNA sequence called genes. Each gene is responsible for producing a single protein.

The gene regulatory network structure can be shown by graph. Where each nodes represents genes, proteins, metabolites, their complexes or even modules and edges between nodes represents the interactions between genes .Proteins and metabolites appear as hidden variables and gene regulatory network can only be inferred from gene expression data as observed variables. So these hidden variable can produce unobserved effects which cannot be measured. Now several properties of gene regulatory network should be considered.

The GRNs should be sparse. That is, only a limited number of genes regulates other genes. Some genes in the network called “hubs” can regulate many genes, i.e. the out-degree of the nodes is not limited.

Another important feature is the scale-free GRNs topology. Scale-free networks have the power distribution function of the connectivity degree. This property provides the robustness of the networks regarding the random topology changes.

In general, each mRNA molecule are responsible to make a specific protein (or set of proteins). It is understood that inside the cell the translation state and transcription state is continuously updated from one state to another just like a feed-forward network in which one state is determined by its previous state. The whole process can be conceptualized as a network where all genes and their products actively participate in a regulatory event. Such network is called a genetic network.

Genetic network will vary for the same species in different situations. In single-celled organisms, regulatory networks respond to the external environment, thus

vital for survival of the organism. Thus a yeast cell, finding itself in a sugar solution, will turn on genes to make enzymes that process the sugar to alcohol. In multicellular animals the same principle has been put in the service of gene cascades that control body-shape. Each time a cell divides, two cells result which, although they contain the same genome in full, can differ in which genes are turned on and making proteins. Sometimes a 'self-sustaining feedback loop' ensures that a cell maintains its identity and passes it on.

1.2 Problem Statement

This paper mainly focuses on formation of Gene Regulatory Network from Microarray gene expression time series dataset of an organism at its various phases of growth. Once Genetic network is formed various processes such as metabolism, gene regulation, transport, and signal transduction, the effects of drugs on a regulatory pathway can be found as well as the development of cancer in a cell can be tracked.

Genetic Network formation can be done by obtaining time series dataset from various website then applying various algorithms on those data or choosing various models on methods of genetic network formation. A few approaches include Bayesian algorithm, artificial neural network formation, minimum description length principle, Conditional dependencies among genes in various time series.

1.3 Thesis Objective

Our thesis objective include identification of the genetic network using the Microarray time series data of a bacteria named “Burkholderia Pseudomalli”. The dataset obtained from the GEO database of www.ncbi.nlm.gov.inand consists of activity values of 5289 genes at 47 instants of time. As computation with the huge Microarray data is very difficult, efforts have been made to work out the

responsible genes actively taking part in the overall bacterial growth process. So whole work can be broadly classified into two phases: data reduction and creation of genetic network. Finally formed genetic network should be evaluated using ROC curve and results will be analyzed

Chapter 2

Literature Review

A gene regulatory network or **genetic network** can be defined as collection of genes in a cell which interact with other genes or itself indirectly and with other substances in the cell by producing a protein and then parking that protein into the promoter of other genes which then regulates the expression level of other genes, thus they govern the expression levels of mRNA and proteins.

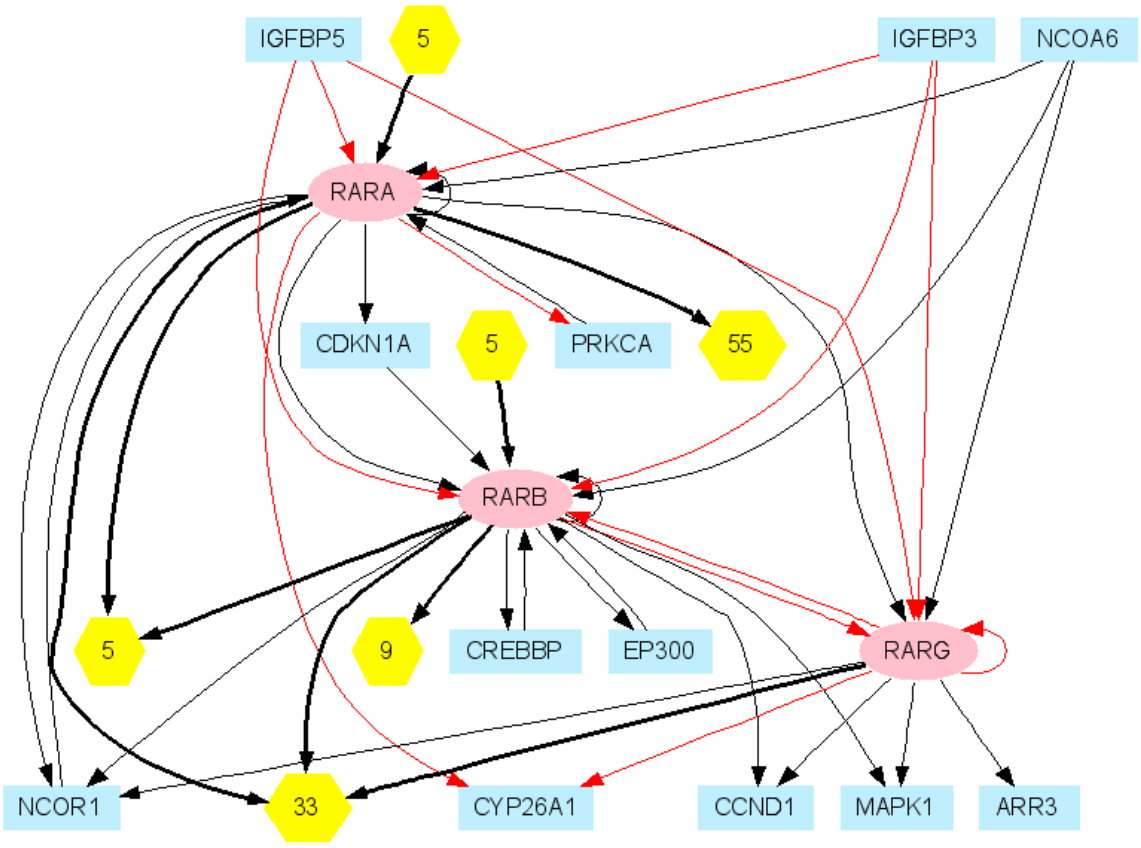

Figure 2.0: Example of gene regulatory network

2.1 Different Models of Gene Regulatory Network

Several methods have been proposed to develop maps of gene interaction, including Bayesian networks, dynamic Bayesian networks with hidden Markov model, and Boolean networks . More recently, neural networks have also been applied to the problem of gene expression data analysis.A brief overview of them is as follows.

2.1.1 Boolean networks

Boolean network is one of the simplest models of gene regulatory network. The genes are represented by nodes and the edges between nodes representing the interactions between genes. In Boolean networks, gene expression levels are discretized and presented by two-states levels. The state of the genes that have expression levels above a certain threshold is 1, otherwise 0. Boolean network is simple and enable analysis for a large networks.But it has disadvantages like it can only provide interaction for two states but in reality we have to consider interaction for multiple state models. As it considers only two states so it undergoes a huge number of information loss. This model is not much suitable for generating regulatory network.

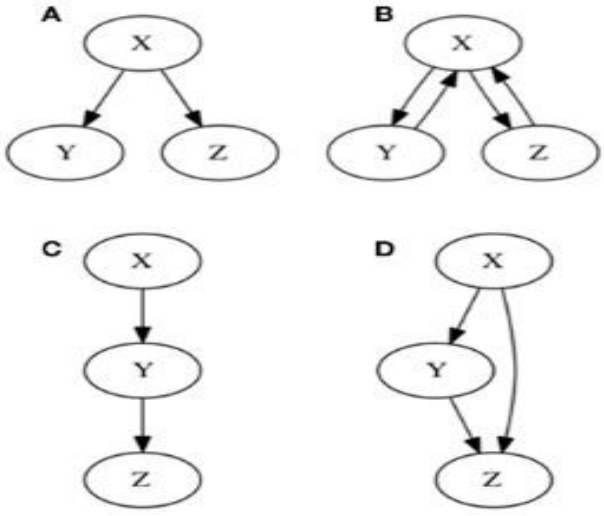

Figure 2.1.1: Example of Boolean Networks

2.1.2 Bayesian networks

Bayesian networks (BNs) are among the most effective models for GRNs inference. A Bayesian network is a special graph model defined as a triple (G; F; q), where G denotes the graph structure, F is the set of probability distributions and q is the set of parameters . The graph structure G is consisted of a set of n nodes X1,X2,…..,Xn and a set of directed edges between nodes. The nodes correspond to the random variables and directed edges show the conditional dependences between the random variables. If there is a directed edge from the nodeX to the node Y , which is denoted as X -> Y ,then X is a parent of Y , denoted as pa(Y), and Y is a child of X. If the node Z can be reached by following a directed path starting from node X, then the node Z is a descendant of X, and X is ancestor of Z. Nodes and edges together have to make a directed acyclic graph (DAG). One directed graph is acyclic if there is no directed path X1 ->X2 …. -> Xn such as X1 = Xn, i.e. there is no pathway that begins and ends at the same node. The joint probability distribution of all nodes is calculated by the conditional dependency of child node on parents. BNs inference is an NP-hard problem, BNs are the most suitable when they are applied to small networks consisted of tens to hundred genes.

It is possible to infer GRNs by BNs based on static, dynamic, discrete or continuous gene expression data. If the node states are continuous, then network inference is more difficult to carry out because of the additional complex calculations.

It is a multi-state model ,so it overcomes the limitation created by Boolean network. It provides better description of interaction between genes and Bayesian network can deal with noisy data.

Its main disadvantage is that it can not deal with feedback regulations and time series data. It has NP-heard learning problem. It greatly increases number of possible state transition.

2.1.3 Dynamic Bayesian networks

BNs can represent probabilistic relations between variables without time lags and their drawback is that they cannot deal with time series data. However, interactions in the real GRNs do not occur simultaneously, so there is a particular time lagging.
Another disadvantage of BNs is that they cannot represent real biological systems, where exists mutual interactions among entities of biological systems, i.e. feedback loops that exist among genes in the GRNs.
These shortcomings make BNs inappropriate for GRNs inference from time series gene expression data, where it is necessary to include dynamic (temporal) features of gene regulation. Thus, BNs are extended to model time features by introduction of dynamic is assumed that the changes in time series gene expression data occur in a finite number of discrete intervals T. Let X = { X1,X2,…..,Xn } is a set of time dependent variables and Xi[t] is a random variable representing the value of Xi at the time point t and 0 <= t <=T. The DBNs are effective for GRNs inference when they are combined with other types of biological data.

It can deal with time-series data, hidden variables. Dynamic Bayesian network can use prior knowledge. It can also deal with missing data and continuous and discrete states.

2.1.4 Artificial Neural Network

Genetic network models are worked out in order to extract the ‘gene regulation matrix’ that describes which gene(s) regulate(s) which gene(s) and what are the

effects of environmental inputs to such network. The regulatory effect on a particular gene expression data can be expressed by neural network. Each node of the network represents a particular gene and regulatory interactions among the genes are given by the wiring among the nodes. Each layer of the network represents the level of expression of genes at an instant of time, say t. in principle, in a fully connected network, all genes can control all other. In reality a gene is regulated by only a few genes. The result of the work justifies the statement. The state of the entire network is updated in every instant of time. The state of a gene expression at current instant is determined by the gene expression(s) of the previous instant. Thus the output of a node at instant $t+\Delta t$ is calculated based on the expression levels of the genes at time t, [xj] and the connecting weights [wij] among the genes.
Besides there are other models for developing a regulatory network like Differential and difference equations models, Association networks etc.

2.2 Related works

Work 1

Soumya Kanti Datta, Srirupa Dasgupta, Sounak Mitra, Dr. Goutam Saha has developed gene regulatory network of a bacteria named as **'Burkholderia Pseudomalli'** using dataset of 5289 genes at 47 instants of times. Their whole work can be broadly classified into two phases: data reduction and search for genetic network using neural network.

2.2.1 Data Reduction

In this work the concept of fidelity matrix has been adopted. The steps of the data reduction can be summarized as below:

- **Representation of Microarray data**: The Microarray data represents the expression value of 5289 genes at 47 time instants. A matrix **mat[5289, 47]** is formed where each row and column represent a gene and a time instant respectively.

- **Elementary column operation**: All the elements of the column k are subtracted from the corresponding elements of column k+1 and the result is stored in column k. This operation finds the change in expression of all the genes at successive time instants. If the resulting value is very near to zero then it is understood that the gene contribute very little over that time interval in the growth process and vice versa. The last column of the original matrix mat[5289, 47] is discarded thereby reducing the size to 5289 by 46 as the last column retains the originals values.

- **Deviation from mean:** The average expression value of each gene is calculated. For each gene, the average is subtracted from all the corresponding gene expressions.

- **Thresholding of the fidelity matrix:** The absolute value of each of the elements of 5289 by 46 matrix is calculated. This is the required 'fidelity matrix'. Then each expression value is compared with a threshold such that any expression less than threshold is reduced to zero or else that is kept as it is.

- **Obtaining the contributing genes:** The genes with only zeros as their expressions are discarded and rest are collected in another array. This manipulation results in 25 contributing genes. Thus the operation segregates 25 most contributing genes from the 5289 genes.

2.2.2 Search for Genetic Network using ARN

The resulting contributing genes were implemented using artificial neural network to form **'**gene regulation matrix' that describes which gene(s) regulate(s) which gene(s) and what are the effects of environmental inputs to such network. The work is based on the assumption that the regulatory effect on a particular gene expression data can be expressed by neural network. Each node of the network represents a particular gene and regulatory interactions among the genes are given by the wiring among the nodes. Each layer of the network represents the level of expression of genes at an instant of time, say t. in

principle, in a fully connected network, all genes can control all other. In reality a gene is regulated by only a few genes. The result of the work justifies the statement. The state of the entire network is updated in every instant of time. The state of a gene expression at current instant is determined by the gene expression(s) of the previous instant. Thus the output of a node at instant t+Δt is calculated based on the expression levels of the genes at time t, [xj] and the connecting weights [wij] among the genes. Finally a genetic network was found having 50 nodes and 67 edges.

Work 2

Sayan day, Dr. Goutam Saha has developed gene regulatory network of a bacteria named as **'Burkholderia Pseudomalli'** (same) and they have used same dataset for computation. Their work can be classified into two phases:

2.2.3 Data reduction

In this work this method is identical to previous approach.

2.2.4 Search for genetic network using Bayesian technique

The Bayesian belief network is a kind of probabilistic models for the construction of genetic network. It uses Direct Acyclic Graph (DAG) to represent dependency relationships between variables. Since every independent statement in belief networks satisfies a group of axioms, we can construct belief networks from data by analyzing conditional independence relationships. The Conditional Independence (CI) test based method is used by all the algorithms of the second category which analyze relations of different quantities based on their dependency relationships. To introduce their approach, let us first review the concept of d-separation, which plays an important role in our algorithm. For any three disjoint node sets X, Y, and Z in a belief network, X is to be d-separated from Y by Z if there is no active undirected path between X and Y. A path between X and Y is active if:

i) Every node in the path having head-to-head arrows is in Z or has a descendant in Z.

ii) Every other node in the path is outside Z.

2.3 Model Validation

ROC curves are applied in the GRNs reconstruction for validation of inferred networks. In a graph between two nodes, it might be an edge or it might be no edge, or expressed by the formalism of machine learning, each edge (instance) of the network belongs to either positive (p) or negative (n) class, and classifier outcomes belong to either class p or class n.

For a given two-class classifier and test samples, four cases are possible:

- TP(true positive), if the instance is positive and it is classified as positive;
- FN(false negative), if the instance is positive and it is classified as negative;
- TN(true negative), if the instance is negative and it is classified as negative
- FP(false positive), if the instance is negative and it is classified as positive.

The following rates are defined based on the defined TP, FN, TN and FP rates:

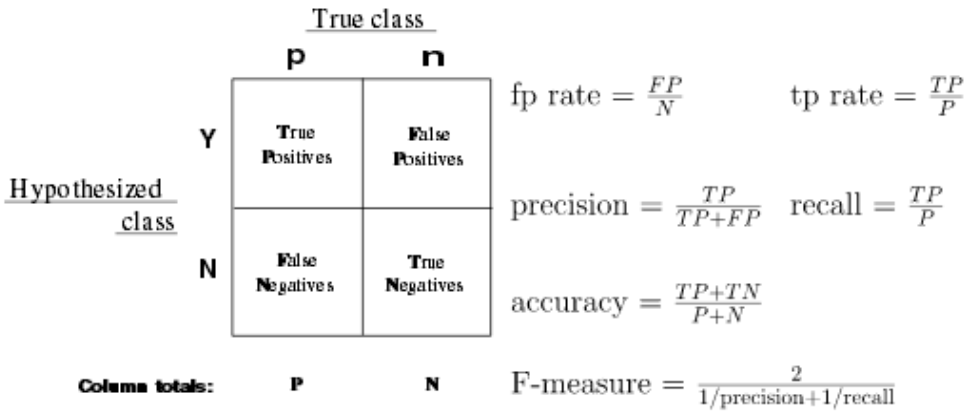

Figure 2.3: ROC measurement partition

Chapter 3

Proposed Method

In the first approach the method of data reduction suffers from the disadvantage that it discards 5264 genes in the process of finding the more contributing genes and the effect of those genes on the entire network is not studied. This may result in considerable deviation from the original and actual genetic network of the mentioned bacteria.

So our proposed method for determining gene regulatory network of a bacterium named as “Burkholderia Pseudomalli” from time series gene expression data obtained at Geo database of NCBI websites by following steps:

- Reducing data in multiple steps in an optimal way
- Finding the relation among the reduced genes
- Determining the most contributing genes
- Formulating a network among them.
- Evaluating results using Roc curve.

3.1 Algorithm

- At first dataset was obtained from geo database of NCBI website. Then the dataset was taken into a variable for further processing.
- The proposed algorithm is for dataset containing the expression values for genes at different timeseires.

- The redundant NAN (not a number) & negative expression values for any gene at any time state are discarded. We replaced them with zero’s. thus, a the dataset now contains the expression values for all genes at each time state containing either 0 or a positive expression value.

Chapter 3. *Proposed Method*

- We calculated the average expression value for the genes considering the genes only have non negative values.

$$\text{Average Expression Value} = \frac{1}{n}\sum_{k=1}^{n} expression\ value$$

n = no. of genes having positive expression value.

- Values less than the average expression values are made zero. Now we get matrix of size 5289 x 47 containing the values either above average expression value or zero.

- Now, elementary column operation is performed in the obtained matrix. We subtracted K+1 th column from the kth column. If the value is positive then in the K+1th column, the expression value is kept as it is. Otherwise, it is made zeros. It also signifies those genes have much contribution for the growth of the bacteria at that time state. Vice versa, in another matrix we kept the expression value in k+1 th column if the resultant column operation is negative. Now, we get two matrices.

- Now, for both the matrices we discarded those rows which have the entire row in each column zero value. Then for each gene, we checked at how many time states it remains active. If it is more than one then we've taken the gene into list.

- We made intersection operation for each genes, among the time state at which it has a value. We perform this operation for both the matrices.

- Now, for each genes in the list we get two intersection sets:-

    - Set1 = common genes those expression values increase with gene i.
    - Set2 = common genes those expression values decrease with gene i.

- Another intersection operation is made between Set1 and Set2. Thus we get the common set of genes for each gene whose expression value increased and decreased with the increase and decrease of ith gene.

- Thus we obtain the sets for each genes and we are coming to a conclusion that, those genes are the most influencing genes for the growth of that particular gene.

Chapter 4

Experimental Results & Analysis

Results were found in each time state how many genes are active , some sample data are:

Time state	No. of expressive genes
1	961
2	1018
4	1257 (most expressive)
32	585
33	509 (least expressive)
44	578
47	898

Table 4.1 : Sample example of number of genes active in each timestate

Chapter 4. *Experimental Results & Analysis*

For each gene its frequency on different time series is calculate as well as no of genes that are responsible for its growth or having interaction with it.

Interacting gene no.	Gene no.	Frequency
0	1	9
1	2	14
2	3	5
182	4	14
96	7	9
7	8	8
29	12	6
5	14	13
842	21	8
599	19	2

Table 4.2: Interacting genes number & frequency of appearance for each gene

Chapter 4. *Experimental Results & Analysis*

Finally genetic network of the responsible genes (found after data reduction)was formed.In this figure connection for gene no 2070 is shown.

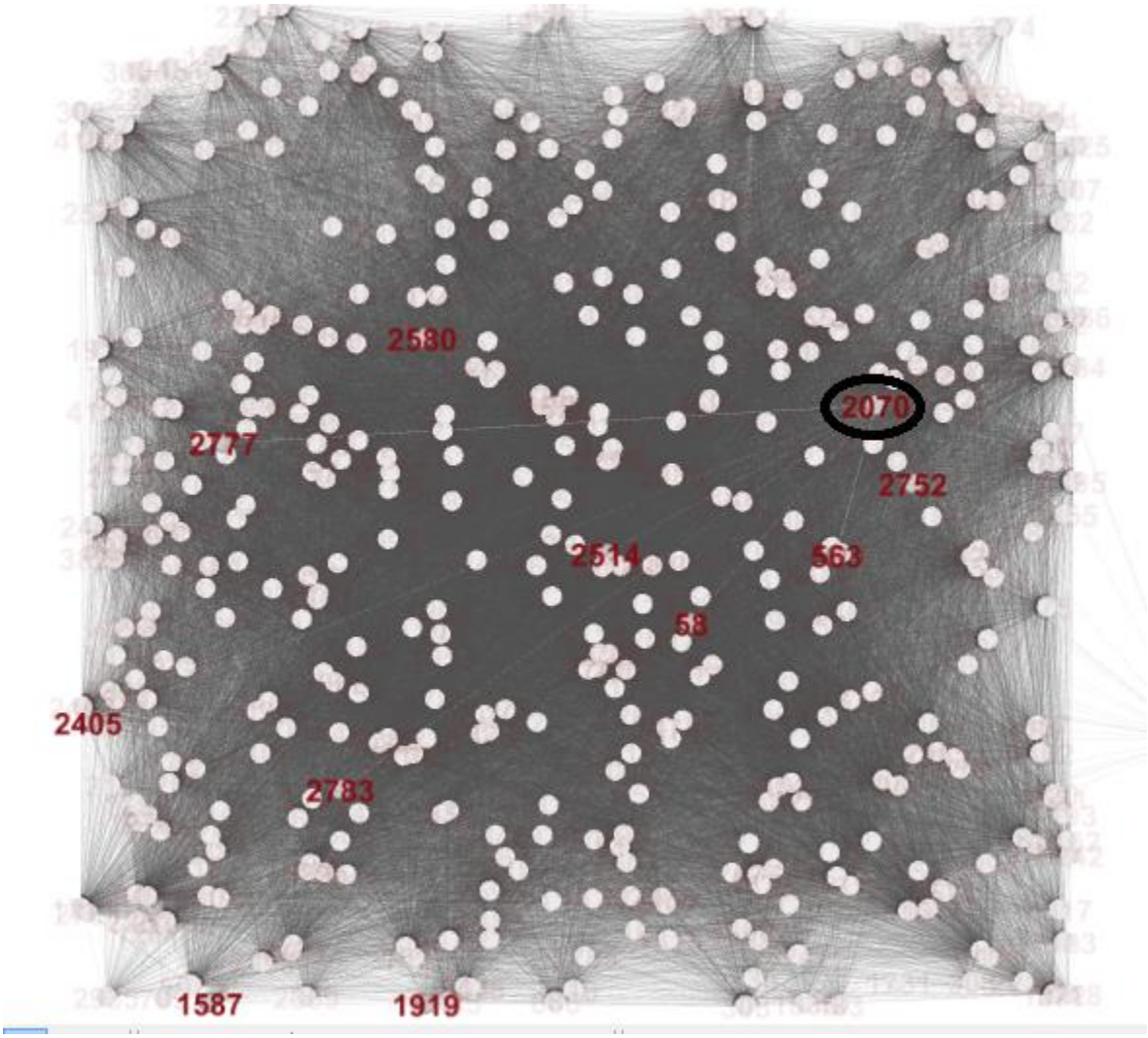

Fig 4.1 : Genetic network of “Burkholderia Pseudomalli”.

Chapter 5

Conclusion

Our proposed method approximates a better network as it-

- Provides regulatory network between more number of genes.
- More gene to gene interactions are considered
- Gene interactions for all time series is considered(unlike work1 and work2).

5.1 Future Works:

- The gene interactions and regulations will be considered for multiple time series and regulatory network will be created accordingly.
- From the obtained genes regulatory network will also be formed using artificial neural network approach
- Each network will be compared with each other.
- The network that provides best relationship will be considered.

Appendix A

Matlab Simulation Code of Proposed Method

positiveExp.m

```
val = geosoftread('GDS2365.soft');
[row col] = size(val.Data);

i=1;
j=1;
for r=1:row
   if isnan(val.Data(r,1))==0
      for c=1:col
         data(i,j) = val.Data(r,c);
         j=j+1;
      end
      j=1;
      i=i+1;
   end
end


[row1 col1] = size(data);
newData = zeros(row1,col1);
sum = 0;
rcount = 0;
ccount = 0;
```

Appendix A. *Matlab Simulation Code*

```
for r = 1:row1
   for c = 1:col1
      if data(r,c) > 0
         newData(r,c) = data(r,c);
         sum = sum + newData(r,c);
         rcount = rcount + 1;
      end
   end
end

threshold = sum/(rcount);

finalData = zeros(row1,col1);
count = 0;
for c = 1:col1
   for r = 1:row1
      if newData(r,c) > threshold
         finalData(r,c) = newData(r,c);
         count = count +1;
      end
   end
   totalcount(1,c) = count;
   count = 0;
end
finalData1 = finalData;

cx = 0;
for r = 1:row1
   for c = 1:col1-1
      secondFilter(r,c) = finalData(r,c+1) - finalData(r,c);
      if secondFilter(r,c) <= 0
         finalData1(r,c+1) = 0;
         cx = cx+1;
      end
   end
end
```

Appendix A. *Matlab Simulation Code*

```
count = 0;
k = 1;
p =0;
for r = 1:row1
   mark = 0;
   for c = 1:col1
      if finalData1(r,c) > 0
         mark = mark + 1;
      end
   end
   if mark >1
      positiveList(k,1) = r;
      k = k + 1;
   end

end

i = 1;
j = 1;
for c = 1:col1
   for r = 1:row1
      if finalData(r,c) > 0
         rmatrix(i,j) = r;
         i = i + 1;
      end
   end
   i = 1;
   j = j + 1;
end

[row2 col2] = size(rmatrix);
[row3 col3] = size(positiveList);
positiveSet([1:row3],[1:row3+1]) = nan;
% positiveSet([1:length(positiveList)],1) = positiveList;
rmax = 0;
```

Appendix A. *Matlab Simulation Code*

```
mp = 2;
for val = 1:row3
   temp = positiveList';
   for c = 1:col2
      for r = 1:row2
         if rmatrix(r,c) == positiveList(val,1)
            temp = intersect(temp,rmatrix(:,c));

         end
      end
   end
   max = length(temp);
   if max>rmax
      rmax = max;
      geneNo = val;
   end
   positiveSet(val+1,[1:length(temp)]) = temp;
   mp = mp + 1;
end
```

negativeExp.m

```
val = geosoftread('GDS2365.soft');
[row col] = size(val.Data);

i=1;
j=1;
for r=1:row
   if isnan(val.Data(r,1))==0
      for c=1:col
         data(i,j) = val.Data(r,c);
         j=j+1;
      end
      j=1;
      i=i+1;
```

```
   end
end


[row1 col1] = size(data);
newData = zeros(row1,col1);
sum = 0;
rcount = 0;
ccount = 0;
for r = 1:row1
   for c = 1:col1
      if data(r,c) > 0
         newData(r,c) = data(r,c);
         sum = sum + newData(r,c);
         rcount = rcount + 1;
      end
   end
end

threshold = sum/(rcount);

finalData = zeros(row1,col1);
count = 0;
for c = 1:col1
   for r = 1:row1
      if newData(r,c) > threshold
         finalData(r,c) = newData(r,c);
         count = count +1;
      end
   end
   totalcount(1,c) = count;
   count = 0;
end
finalData1 = finalData;

cx = 0;
```

Appendix A. *Matlab Simulation Code*

```
for r = 1:row1
   for c = 1:col1-1
      secondFilter(r,c) = finalData(r,c) - finalData(r,c+1);
      if secondFilter(r,c) <= 0
         finalData1(r,c+1) = 0;
         cx = cx+1;
      end
   end
end

count = 0;
k = 1;
p =0;
for r = 1:row1
   mark = 0;
   for c = 1:col1
      if finalData1(r,c) > 0
         mark = mark + 1;
      end
   end
   if mark >1
      negativeList(k,1) = r;
      k = k + 1;
   end

end

i = 1;
j = 1;
for c = 1:col1
   for r = 1:row1
      if finalData(r,c) > 0
         rmatrix(i,j) = r;
         i = i + 1;
      end
   end
```

```
   i = 1;
   j = j + 1;
end

[row2 col2] = size(rmatrix);
[row3 col3] = size(negativeList);
negativeSet([1:row3],[1:row3+1]) = nan;
% negativeSet([1:length(negativeList)],1) = negativeList;
rmax = 0;
mp = 2;
for val = 1:row3
   temp = negativeList';
   for c = 1:col2
      for r = 1:row2
         if rmatrix(r,c) == negativeList(val,1)
            temp = intersect(temp,rmatrix(:,c));

         end
      end
   end
   max = length(temp);
   if max>rmax
      rmax = max;
      geneNo = val;
   end
   negativeSet(val+1,[1:length(temp)]) = temp;
   mp = mp + 1;
end
```

mutualIntersect.m

```
[nrow ncol] = size(negativeList);
[prow pcol] = size(positiveList);
```

Appendix A. *Matlab Simulation Code*

```
x = 1;
y = 1;
p = intersect(positiveList,negativeList);

for r = 1:size(p)
   for pr = 1:prow
      if (p(r,1) == positiveList(pr,1))

         fpositiveSet(x,:) = positiveSet(pr+1,:);
         x = x + 1;
         break;
      end
   end

   for pr = 1:nrow
      if (p(r,1) == negativeList(pr,1))

         fnegativeSet(y,:) = negativeSet(pr+1,:);
         y = y + 1;
         break;
      end
   end
end



[row col] = size(fpositiveSet);
temp = 0;
for r = 1:row
   count = 0;
   for c = 1:col
      if fpositiveSet(r,c) > 0
         count = count + 1;
      end
   end
   if temp < count
```

```
      temp = count;
   end
end


%  fnegativeSet = negativeSet(2:2804,:);
fpositiveSet = fpositiveSet(:,1:2804);

mIntersectSet = zeros(2804,2804);

 for i = 1:length(p)
    temp = intersect(fpositiveSet(i,:),fnegativeSet(i,:));
    mIntersectSet(i,1:length(temp)) = temp;
 end
```

cardinality.m

```
data = csvread('Actual_output.csv');

[dr dc] = size(data);

for i = 1:dr
   cardinal(i,1) = data(i,1);
   count = 0;
   for r = 1:dr
      for c = 1:dc
         if data(i,1) == data(r,c)
            count = count + 1;
         end
      end
   end
   cardinal(i,2) = count;
end
```

Appendix A. *Matlab Simulation Code*

```
% for the outdegree

[row col] = size(ac_output);

for r = 1:row
   count = 0;
   out_degree(r,1) = ac_output(r,1);
   for c = 2:col
      if ac_output(r,c) == 0
         break;
      else
        count = count + 1;
      end
   end
   out_degree(r,2) = count-1;
end
```

exp.m

```
val = geosoftread('GDS2365.soft');
[row col] = size(val.Data);

i=1;
j=1;
for r=1:row
   if isnan(val.Data(r,1))==0
      for c=1:col
         data(i,j) = val.Data(r,c);
         j=j+1;
      end
      j=1;
      i=i+1;
   end
end
```

Appendix A. *Matlab Simulation Code*

```
[row1 col1] = size(data);
newData = zeros(row1,col1);
sum = 0;
rcount = 0;
ccount = 0;
for r = 1:row1
   for c = 1:col1
      if data(r,c) > 0
         newData(r,c) = data(r,c);
      end
   end
end
k = 1;
for r = 1:row1
   mark = 0;
   for c = 1:col1
      if newData(r,c) > 0
         mark = mark + 1;
      end
   end
   if mark >0
      positiveList(k,1) = r;
      k = k + 1;
   end

end


i = 1;
j = 1;
for c = 1:col1
   for r = 1:row1
      if newData(r,c) > 0
         rmatrix(i,j) = r;
         i = i + 1;
```

```
      end
   end
   i = 1;
   j = j + 1;
end

[row2 col2] = size(rmatrix);
[row3 col3] = size(positiveList);
positiveSet([1:row3],[1:row3+1]) = nan;
% positiveSet([1:length(positiveList)],1) = positiveList;
rmax = 0;
mp = 2;
for val = 1:row3
   temp = positiveList';
   for c = 1:col2
      for r = 1:row2
         if rmatrix(r,c) == positiveList(val,1)
            temp = intersect(temp,rmatrix(:,c));

         end
      end
   end
   max = length(temp);
   if max>rmax
      rmax = max;
      geneNo = val;
   end
   positiveSet(val+1,[1:length(temp)]) = temp;
   mp = mp + 1;
end


[row col]  = size(newData);
Set([1:row],[1:row+1]) = nan;
for r = 1:row
   temp = ones(5289,1);
```

```
   for c = 1:col
      if newData(r,c) ~= 0
         temp = intersect(temp,newData(:,c));
      end
   end
    Set(r,[1:length(temp)]) = temp';
end
```

roc.m

```
ac_output = csvread('Actual_output.csv');
[row col] = size(ac_output);
[row1 col1] = size(positiveSet);

for r = 1:row
   g_no = ac_output(r,1);
   desiredSet(r,:) = positiveSet(g_no,:);
end
 desiredSet = [ac_output(:,1) desiredSet(:,1:2804)];

 ival = intersect(ac_output,desiredSet);


tp = 0;
for r = 1:row
   ival = intersect(ac_output(r,2:row),desiredSet(r,2:row));
   tp = tp + length(ival);
end


fp = 1;
for r = 1:row
   for c = 2:col
```

```
      if ac_output(r,c) > 0
         mark = 0;
         for c = 2:row
            if desiredSet(r,c) == ac_output(r,c)
               mark = 1;
               break;
            end
         end
         if mark == 0
            fp = fp + 1;
         end
      end
   end
end


fn = 1;
for r = 1:row
   for c = 2:col
      if isnan(desiredSet(r,c)) == 0
         mark = 0;
         for c = 2:row
            if desiredSet(r,c) == ac_output(r,c)
               mark = 1;
               break;
            end
         end
         if mark == 0
            fn = fn + 1;
         end
      end
   end
end

total = 2748 * 2748;
tn = total - (tp + fn + fp);
```

Appendix A. *Matlab Simulation Code*

```
desiredSet1 = desiredSet;
for r = 1:row
   for c = 1:col
      if isnan(desiredSet1(r,c))
         desiredSet1(r,c) = 0;
      end
   end
end

g_list = ac_output(:,1)';
for r = 1:row
   g_list = [g_list; ac_output(:,1)'];
end

g_list = g_list(1:2748,:);
ac_output1 = ac_output(:,2:2749);
desiredSet1 = desiredSet1(:,2:2749);
value = 0;


for r = 1:row
   dif1 = setdiff(g_list(r,:),ac_output1(r,:));
   dif2 =  setdiff(g_list(r,:),desiredSet1(r,:));
   final_diff = intersect(dif1,dif2);
   value = value + length(final_diff);
end
```

Bibliography

[1] Soumya Kanti Datta, Srirupa Dasgupta, Sounak Mitra and Dr. Goutam Saha,*"**Determination of Genetic Network from Micro-Array Data using Neural Network Approach**"* ,International Conference of Communication, Computers and Devices 2010, Kharagpur, INDIA December 10-12 ,PAPER IDENTIFICATION NUMBER: 218, 2010.

[2] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome – wide expression data", Proc. Nat. Acad. Sci, 95(25): 14863 – 14868, 1998.

[3] N. Friedman, M. Linial, I. Nachman and D.Pe'er." Using Bayesian networks to analyze expression data." J. Comput. Biol. , 7: 601 – 620, 2000.

[4] D. Husmeier. "Reverse Engineering of genetic networks with Bayesian networks. ", Biochem. Soc. Trans., 31: 1516 – 1518, 2003.

[5] J. C. Liao et al. ," Network component analysis: reconstruction of regulatory signals in biological systems." , Proc. Nat. Acad. Sci., 100(26): 15522 – 15527, 2003

[6] M. K. Yeung, J. Tegner and J. Collins," Reverse engineering genetic networks using singular value decomposition and robust regression.", Proc. Nat. Acad. Sci., 99(9): 6163 – 6168, 2002.

[7] Sayan day and Dr. Goutam Saha, ***"Determination and study of Genetic Network responsible for growth of a fungus using the concepts of Baysian algorithm"***. International Conference on Systems in Medicine and Biology 16-18 December 2010, liT Kharagpur, INDIA, 71-80.

Bibliography

[8] Joshua Stender, “Microarrays to Functional Genomics: Generation of Transcriptional Networks for Microarray experiments”, December 3, 2002, Department of Biochemistry.

[9] Patrix D’Haeseller , Shoudan Liang and Ronald Somogyi, “Genetic Network Interface: From Co-Expression Clustering to Reverse Engineering”, lecture thesis.

[10] Niranjan Baisakh and Swapan Datta, “Metabolic Pathway Engineering for Nutrition Enrichment”, chapter 19.Plant breeding, Genetics, Biochemistry division, International Rice Research Institute, Philippines.

[11] McCulloch, W.S. and Pitts, W., “A logical calculus of the ideas immanent in the nervous acitivity,” Bull. Math. Biophys., vol. 5, pp. 115 – 133, 1943.

[12] M. Minisky, and S. Papert, Perceptrons, MIT Press, Cambridge, 1988.

[13] F. Rosenblatt “The Perceptron: a perceiving and recognizing automation”, Technical Report 85-460-1, Cornell Aeronautical Laboratory, 1957.

[14] F. Rosenblatt ,“The Perceptron: a probabilistic model for information storage in the brain”, Psych. Rev., vol. 65, pp. 365-408, 1958.

[15] Hartemink et aI., "Construction of networks using Bayesian belief algorithms", Supplement 1, 18th Edition, S216-S224,2002.

[16] J. Cheng, D. A. Bell and W. Liu: "An algorithm for Bayesian Belief network construction from data", In proceedings of AI and STAT, Florida, pp. 83-90, 1997.

[17] Y. Jing, V. A. Smith, P. P. Wang, A. 1. Hartemink and E. D. Jarvis, "Using Bayesian Network inference algorithms to recover molecular Genetic regulatory networks", 12th Edition, 18 June, 2004.

[18] D. Heckerman, "A Tutorial on learning with Bayesian Networks", 1996 Technical report MSR-TR-95-06, Microsoft Research, March, 1995 (Revised November, 1996).

[19] Blagoj Ristevski,” ***A survey of models for inference of gene regulatory networks*****”**, Nonlinear Analysis: Modelling and Control, 2013, Vol. 18, No. 4, 444–465.

[20] Chao Sima, Jianpong Hua ,Sungwon Jung , “Inference of gene regulatory networks using time series data : A survey”,Current Genomics, 2009,416-429.

Bibliography

[21] Barker NA, Myers CJ, Kuwahara H, “Learning genetic regulatory network connectivity from time series data.” IEEE/ACM Trans Comput Biol Bioinform.2011 Jan-Mar;8(1):152-65. doi: 10.1109/TCBB.2009.48.