RECONSTRUCTION OF GENE REGULATORY NETWORKS USING A HYBRID MODEL

Submitted by

DEBAPRIYA PAUL

Registration Number: D01-1211-0094-19

Roll Number: 97/CSM/191019

In partial fulfilment for the award of the degree of

Master of Technology

In

Computer Science and Engineering

Under the supervision of

Prof. Rajat Kumar Pal

Department of Computer Science and Engineering

University of Calcutta

2019–21

ABSTRACT

The ‘Central Dogma’ of molecular biology states that there are some regulatory interactions between genes, which can be represented by a complex network structure, known as a *gene regulatory network*. However, it does not throw any light on the nature and strength of such interactions. It is also nearly impossible to extract such information from the ‘Wet Lab’. Thus, computational tools are required to determine the nature and strength of regulatory interactions between genes from the underlying information present in time-series gene expression datasets. The computational tools used for this purpose can be broadly divided into two groups, namely power-law based tools, like *S-system* and *half-system*, and exponential-law based tools like *recurrent neural network*. The aim of this work is to reconstruct gene regulatory networks from temporal expression profiles using a hybrid formalism based on *half-system* and *recurrent neural network*. The hybrid model has been employed to accurately extract the underlying information from the given time-series datasets. The model parameters have been trained using *particle swarm optimization*. The proposed methodology will be used for the reconstruction of small-scale and medium-scale networks in the next part of our work.

CONTENTS

CHAPTER 1: INTRODUCTION

The basic functional unit of every living organism is cell. Every biological process occurring in a living organism can be explained by the functionality of these cells and their interactions. The interaction between genes can explain all the biological processes. We can also explain the reason behind various diseases appearing in living organisms by identifying the wrong interactions between the genes. The gene expression level is the measure of the regulatory effect between a pair of genes or a group of genes. The micro array data experiments help the researchers to get the time series gene expression [1] value at different time points with different genes to understand to understand the regulatory relationships of the genes in the living organisms. With the help of this dataset, we are trying to construct a regulatory network which can predict the original biological interactions between the genes in a living organism. This process of reconstruction of the regulatory network is called Reverse Engineering. For this reverse engineering, the researchers have used various methods. For making the continuous time series dataset into a discrete one, they proposed some differential equations which are treated as the objective functions. They used various methods such as Mean Square Estimation to estimate the error. Then they used some meta-heuristic optimization algorithms to optimize the error. We have followed the same pathway to reconstruct the GRN. We have represented the network as a directed graph G (V, E), where V, the vertex, represents the genes and E, the edges between the vertices, represent the interactions between the genes. A weighted directed graph has been used to represent the interaction between the genes where 0 means no interaction, +1 means activation (i.e., the expression level of the target gene increases) and -1 means inhibition (i.e., the expression level of the target gene decreases).

A GRN suffers from two major problems, (i) Noise and (ii) Curse of Dimensionality. Noise occurring in a dataset is the impurities present in it due to the effect of some environmental hazards or some external or internal disturbances. Curse of Dimensionality [2] occurs if we have a dataset where the number of genes present is more than the number of time points. Curse of Dimensionality occurs in a large network where a large number of genes are present, but it does not occur in comparatively smaller networks where the number of genes (4 to 20 genes) is much less than the number of time points.

CHAPTER 2: LITERATURE SURVEY

In this part we have discussed about some models on which the main construction of GRN is based. Later we have discussed about some optimization techniques and then the methodology, how we have processed our work to reconstruct the network.

2.1. Bayesian Network

Bayesian network is a probabilistic graphical model based on the two mathematical areas, graph theory and probability. The Bayesian network is represented by a directed acyclic graph (DAG), G (X, E), where X is the set of vertices, i.e., the gene expressions for the GRN and E is the set of all the edges that represents the regulatory relationship between the genes. This network implicitly represents the Markov's assumptions. Basically, Bayesian networks are used to assign conditional probabilities to the genes' regulation parameters.

Friedman *et al*. [3] was one of the first to use Bayesian network for the GRN modelling, closely followed by Pe'er *et al*. [4]. Bayesian networks can handle noise and uncertainty and integrate prior knowledge to reinforce the casual relationships amongst the genes but fail to capture the underlying dynamics in temporal expression data. Bayesian networks are time-consuming, cannot handle large-scale networks due to their high computational complexity, and cannot infer self-regulations and feedback loops. To resolve this issue and also to incorporate the dynamic or temporal nature of gene expression, Dynamic Bayesian networks [5] were proposed. The introduction of the dynamic Bayesian network approach in the gene regulation increased the accuracy of the network and reduced the computational time. They selected those regulators (either activators or inhibitors) which are expressed previously. Furthermore, they also evaluated the time difference between the regulator gene and the target gene which helped to get an idea about the transcriptional time lag. They applied this approach on the yeast cell cycle and obtained a much better result with less computational time. Dynamic Bayesian networks can effectively handle the problems like hidden variables, prior knowledge, and missing data.

In [6], the authors used Bayesian network approach in three different ways for gene expression data. Firstly, they induced Bayesian classifier from micro array data. Then they proposed a pre-processing scheme to induce the Bayesian classifier for the gene expression data. Finally, they evaluated different types of Bayesian classifier to evaluate this kind of data. They applied the proposed methodology on nine sets of various cancer dataset.

2.2. Boolean Network

Boolean network is a dynamic model of synchronous interactions between the nodes in a network. In Boolean network Boolean variables are used to represent the interaction between the genes. Basically, Boolean networks are binary models, which consider that a gene can have only two states, namely, 1 for active and 0 for inactive [7]. The effect of other genes on the state change of a given gene is described through a Boolean function.

The representation of genetic network through Boolean network was first proposed by S. A. Kaufmann in the year of 1969 [8]. In his paper, he considered a directed graph consisting of n nodes and the degree per node will be maximum k, he defined such a network as NK Boolean network. In this paper, the author showed that binary genes had the stability comparable with that of the living organism. Low per degree connected NK Boolean network showed more similar properties of biological system like robustness, perturbation, and periodic behavior.

In “A Tutorial on Analysis and Simulation of Boolean Gene Regulatory Network Models” proposed by Yufei Xiao, in the year of 2009 *et al.* [9] the author used Boolean network and probabilistic Boolean network, a dynamic approach in order to analyze GRN. The paper has also explained the relationship between the probabilistic Boolean network and the dynamic Bayesian network, using Markov analysis.

Although Boolean networks make it possible to explore the dynamic behavior of a gene regulatory system, they ignore the effect of genes at intermediate levels and inevitably cause information loss. Moreover, Boolean networks assume the transitions between genes’ activation states are synchronous, which is biologically implausible.

2.3. Recurrent Neural Network (RNN)

The inter-regulatory effect between the genes can be represented with the help of the Recurrent Neural Network (RNN). In using RNNs for genetic network inference, we are mainly concerned with their ability to interpret complex temporal behaviour, which is an important characteristic of time series gene expression data and makes them different from static expression data [10]. The recurrent structure of RNNs effectively reflects the existence of feedback, which is essential for gene regulatory system.

D’haeseleer discussed a realization of RNNs in modelling gene networks using synthetic data [11]. Vohradsky investigated the dynamic behaviour of a 3-gene network in the framework of RNNs [12]. In 2007, Xu *et al*. proposed an RNN based model where a meta-heuristic optimization algorithm, Particle Swarm Optimization (PSO) was used as the training algorithm [13]. In this paper, the author applied this model to a 4-gene artificial network and a real-world experimental dataset generated from an 8-gene DNA SOS Repair network of *E.coli*. In 2007, the authors developed an RNN based model [14], in which optimization techniques like Differential Evolution (DE) and PSO were applied on the same datasets as mentioned above. In 2012, an RNN, Ant Colony Optimization (ACO) and PSO based model were proposed by Kentzoglanakis *et al*. [15]. The authors used RNN for modeling the dynamical behaviour of gene regulatory systems. More specifically, ACO was used for searching the discrete space of network architectures and PSO for searching the corresponding continuous space of RNN model parameters. The authors did this experiment on a 4-gene synthetic dataset and a 10-gene real-world network from which dataset was artificially generated using Gene Net Weaver (GNW). GNW is nothing but an easy to use JAVA tool. The results obtained from the experiments demonstrated the relative advantage of utilizing problem-specific knowledge regarding biologically plausible structural properties of gene networks over conducting a problem-agnostic search in the vast space of network architectures.

In 2016, Mandal *et al*. [16] proposed an RNN, Bat algorithm (BAT) and PSO based model. The authors here analyzed a synthetic dataset of a small-scale artificial network, both synthetic and real-world dataset of DNA SOS Repair Network of *E.coli* and an artificial dataset generated from a real-world network of yeast. In 2017, an RNN and Bat Algorithm based model were proposed by Mandal *et al*. [17]. First, this model was used to analyze 6 sets of noise-free data each of 4 genes. Then this methodology was also applied on 4-gene noisy data. 5 percent Gaussian noise was added to this dataset.

2.4. S-System

In 1988 Michael A. Savageau [18] first proposed the S-system, which is a set of non-linear differential equations. S-system is based on the Biochemical System Theory (BST), used for modelling and analyzing biological system provides a good comparison in between the biological relevance and mathematical flexibility. It can also express the saturable and synergistic characteristics of a biological system and other complex systems, hence the name S-system. It consists of a combination of simple derivatives and power function. S-system can be called as the canonical form of many differential equations. This S-system can be used as a tool to infer gene regulatory network. In 2010, H. Wang has proposed S-System to infer gene regulation [19]. Just as S-system can be applied to a very small network, their unified approach makes it possible to apply to a comparatively large scale network by fast parameter estimation for finding the interaction. In 2013, Leon Palafox *et al*. [20] proposed S-System and Dissipative Particle Swarm Optimization (DPSO) based model to analyze a small 5-gene network and also two *in-silico* real-world datasets of yeast and SOS DNA Repair Network of *E. coli*. In 2015, Jer-Nan Juang *et al*. [21] proposed an S-system based model with hybrid parameter estimation algorithm to investigate the gene regulatory network. In 2016, Khan *et al*. [16] proposed a model based on the combination of S-system and Bat Algorithm (BA) which can infer GRNs. The authors used a 5-gene artificial noise-free dataset and an *in-vivo* noisy dataset of 8-genes of *E. coli* DNA SOS Repair Network and a real-world 20-gene network extracted from Gene Net Weaver (GNW).

2.5. Particle Swarm Optimization (PSO)

Till now various approaches have been used for the reconstruction of GRN using the micro-array time series dataset. ‘No Free Lunch’ (NFL) theory [22] stated that “for any algorithm any elevated performance over one class of problems is offset by performance over another class.” Thus, considering this NFL theory, there are several techniques to solve this meta-heuristic problem but none of them are efficient to correctly detect the regulation of a GRN.

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995. It is an evolutionary optimization algorithm which improves the candidate solutions in an iterative process. PSO is implemented with a memory mechanism, which can retain the information of previous best solutions that may get lost during the population evolution. It is based on the behaviour of bird flocking or fish schooling.

PSO has many desirable characteristics, such as flexibility in balancing of global and local searches, computational efficiency and ease of implementation. It is observed that PSO can achieve faster convergence than conventional back-propagation in training feed-forward neural networks for non-linear function approximation [23]. In 2007, Rui *et al*. [14] has proposed a hybrid of PSO and Differential Evolution (DEPSO) to optimize the problem of GRN. In 2009, a noble hybrid model of Particle Swarm Optimization and Recurrent Neural Network (PSO-RNN) was proposed for gene inference method.

2.6. Bat Algorithm inspired Particle Swarm Optimization Algorithm

BAPSO algorithm is Bat Algorithm inspired Particle Swarm Optimization technique. Khan *et al*. [24] proposed a new methodology for the reconstruction of GRN from any given noisy temporal genetic expression dataset using a statistical paradigm based on the theory of combination. The methodology is based on a hybrid swarm intelligence framework which is basically a Bat Algorithm (BA) inspired Particle Swarm Optimization (PSO) algorithm, named BAPSO by the authors. In this paper, the authors have used Recurrent Neural Network (RNN) as the tool for modelling the required network dynamics. The authors assumed a maximum number of regulators to be 4. The methodology has been applied on an artificial network with 4 genes, an 8-gene *E. coli* SOS DNA Repair Network of the real-world and two networks extracted from the genome of yeast (10 genes) and *E. coli* (20 genes) with the help of GNW. The fundamental mathematical theory of combination has been applied to search all possible candidate solutions in the discrete search space of network construction exhaustively. The corresponding RNN model parameters have been trained by the proposed BAPSO technique that can replicate the original network dynamics faithfully. The results obtained in this paper show that prediction errors are much reduced. A new inertia weight update technique has been proposed in this paper that has been able to produce better results than individual PSO or BA algorithms. Another change has been made inspired by the virtual bats in BA that has been incorporated in BAPSO technique is the initialization of the velocity vector of each particle to zero instead of a random vector. The authors have observed that this has helped in preventing the particles from having an initial unguided velocity that may divert them away from a potential optimal solution in the search space.

CHAPTER 3: MOTIVATION AND METHODOLOGY

Genes are the basic functional units of cells in living organisms. Each cell consists of several genes where some are activated whereas some are inhibited. Those genes which are activated are responsible for production of proteins through two stages transcription and translation. The entire process of production of proteins from genes is called Central Dogma. Hence Central Dogma, a term used in molecular biology, means that DNA makes RNA and RNA makes proteins.

Transcription is the process by which DNA (Deoxyribonucleic acid) is copied to RNA (Ribonucleic acid), specifically mRNA (messenger RNA). Translation is the process by which mRNA is used to produce protein. In translation mRNA is decoded into specific ribosome from which the required amino acid chain is produced. A transcription factor (or sequence-specific DNA-binding factor) is protein that controls the production of mRNA from DNA by binding to a specific DNA sequence. For prokaryotic cells, transcription and translation processes are coupled whereas in eukaryotic cells, transcription and translation processes are spatially and temporally separate. Gene expression basically shows the outcome of the process of synthesis of proteins from a gene. The function of transcription factor is to regulate the gene expression by turning on and off genes in order to make sure they are expressed in the right cell at the right time and in the right amount throughout the life of the cell and the organism.

Transcription factors work alone or with other proteins in a complex, by promoting as an activator or by blocking as an inhibitor the recruitment of RNA polymerase, the enzyme performing transcription of genetic information from DNA to RNA, to specific genes. Hence activation means increasing the rate of synthesis of protein and inhibition means decreasing the rate of synthesis of protein. Biologically we easily get the knowledge about how and when transcription and translation take place. But the main motivation behind the reconstruction of gene regulatory network from the available time series gene expression dataset is to understand the true interaction between the genes which in regard helps us in finding out the reason behind various diseases occurring in the cellular living organisms and to understand how a biological system works. The reason behind developing the model is also to understand which protein is produced by which particular gene and why that gene is producing the protein, how the genes are activated, whether directly by a specific protein or as a result of some genetic interactions.

Time series gene expression data (DNA micro array data) is stored in a 2D matrix, where rows represent the number of genes and the columns represent the number of time points, and each element of the matrix represents the amount of gene expression value at a particular time instant.

3.1. Recurrent Neural Network (RNN)

In this paper we are planning to use Recurrent Neural Network as the tool to reconstruct gene regulatory network from time series gene expression data. RNN helps to interpret the complex temporal behavior of the gene regulatory network. The presence of the property of feedback makes RNN a particularly efficient tool in modeling GRNs.

The regulation of the expression of any particular gene by another gene or a group of genes can be expressed with the help of the RNN formalism. In such an RNN model, each node represents a particular gene and the edges between the nodes represent regulatory interactions among the genes. Each tier of the neural network defines the genetic expression level of the genes at a specified time *t*. The level of expression of any particular gene at a time $t' = t + \Delta t$ depends upon the genetic expression level of all the genes (x_j) at the preceding time t and the weights of their corresponding edges (w_{ij}) with that particular gene. The intensity and the type of a particular interaction between a target gene (i) and a regulator gene (j) are defined by w_{ij}, where a positive value denotes activation and a negative value denotes inhibition while a zero value implies that there is no interaction between i and j.

For a continuous time system, the gene regulation model can be represented through a recurrent neural network formulation.

$$\tau_i \frac{dx_i}{dt} = f_i\left(\sum_{j=1}^{N} w_{ij}x_j + \beta_i\right) - x_i \dots (1)$$

$$\tau_i \frac{dx_i}{dt} = \frac{1}{1 + e^{-\left(\sum_{j=1}^{N} w_{ij}x_j + \beta_i\right)}} - x_i \dots (2)$$

where x_i is the gene expression level for the ith gene $(1 \leq i \leq N)$, N being the total number of genes in the system; f_i is a non-linear function, usually a sigmoid function as shown in equation (2); w_{ij} represents the effect of the jth gene on the ith gene $(1 \leq i, j \leq N)$. x_j denotes the gene expression level for the jth gene $(1 \leq j \leq N)$ at a particular time t. τ_i is the time constant of *i*th gene; β_i is the bias term for the ith gene. $\frac{dx_i}{dt}$ is the rate of expression of any gene i. The model can also be described in a discrete form (for computational convenience, since we only measure at certain time points):

$$\frac{x_i(t+\Delta t)-x_i(t)}{\Delta t}=\frac{1}{\tau_i}\left[\frac{1}{1+e^{-(\sum_{j=1}^{N}w_{ij}x_j(t)+\beta_i)}}-x_i(t)\right]\dots(3)$$

$$x_i(t+\Delta t)=\frac{\Delta t}{\tau_i}\left[\frac{1}{1+e^{-(\sum_{j=1}^{N}w_{ij}x_j(t)+\beta_i)}}\right]+\left(1-\frac{\Delta t}{\tau_i}\right)x_i(t)\dots(4)$$

3.2. S-System

S-system, a power-law based formalism, is preferred for modelling of gene regulatory networks because it can model both synergy and saturation. The mathematical representation of the S-system model is as follows:

$$\frac{dx_i}{dt}=\alpha_i\prod_{j=1}^{N}x_j{}^{g_{ij}}-\beta_i\prod_{j=1}^{N}x_j{}^{h_{ij}}\dots(5)$$

Where α_i and β_i are the rate constants for the production and degradation terms, respectively; g_{ij} and h_{ij} are the kinetic orders of the system, also known as the exponential parameters and *N* is the number of genes in the network. $g_{ij}>0$ or $h_{ij}<0$ denotes activation of gene *i* by gene *j*, while $g_{ij}<0$ or $h_{ij}>0$ denotes inhibition of gene *i* by gene *j* .

Assuming$\frac{dx_i}{dt}\approx\frac{\Delta x_i}{\Delta t}=\frac{x_i(t+\Delta t)-x_i(t)}{\Delta t}$, we get:

$$x_i(t+\Delta t)=x_i(t)+(\Delta t.\alpha_i).\prod_{j=1}^{N}[x_j(t)]^{g_{ij}}-(\Delta t.\beta_i).\prod_{j=1}^{N}[x_j(t)]^{h_{ij}}\dots(6)$$

The total number of parameters that are needed to be estimated for each gene in the S-system formalism is (2*N* + 2). Thus, the model requires 2*N* (*N* + 1) parameters for the case of an *N*-gene network.

3.3. Half-System (HS)

S-system modelling poses a few problems. It is computationally expensive to train the model due to the large number of parameters required compared to RNN, i.e. 2*N* (*N* + 1) compared to *N* (*N* + 2). Also a problem arises when both the predicted g_{ij} and h_{ij} are of the same sign, which suggests dual regulation. This is not possible in reality as a gene cannot activate as well as inhibit another gene at the same time. To avoid these issues, we have employed the half-system formalism, in place of S-system. The mathematical representation of half-system is as follows:

$$\frac{dx_i}{dt} = \alpha_i \prod_{j=1}^{N} x_j{}^{w_{ij}} \dots (7)$$

Where α_i is rate constant; w_{ij} is the only kinetic order of the system; *N* is the number of genes in the network. Assuming $\frac{dx_i}{dt} \approx \frac{\Delta x_i}{\Delta t} = \frac{x_i(t+\Delta t)-x_i(t)}{\Delta t}$, we get:

$$x_i(t + \Delta t) = x_i(t) + (\Delta t.\alpha_i).\prod_{j=1}^{N}[x_j(t)]^{w_{ij}} \dots (8)$$

The total number of parameters required for training is thus (*N* + 1). Thus, the model requires *N* (*N* + 1) parameters for an *N*-gene network, which is exactly half of the number of parameters of S-system.

Two modifications have been made to the traditional half-system formalism to improve the stability of the model as well as increase the prediction accuracy. Firstly, a self-degradation term has been added to equation (7). The product of gene expression is not accumulated at the reaction site but is used up in the process of regulating the expression of other genes. The depletion of the expression product is dependent on the current level of expression of a gene. For all these reasons, equations (7) and (8) are modified as follows:

$$\frac{dx_i}{dt} = \alpha_i \prod_{j=1}^{N} x_j{}^{w_{ij}} - \gamma_i.x_i \dots (9)$$

$$x_i(t + \Delta t) = (\Delta t.\alpha_i).\prod_{j=1}^{N}[x_j(t)]^{w_{ij}} + (1 - \Delta t.\gamma_i).x_i(t) \dots (10)$$

where $\gamma_i > 0$. Equation (10) has been further modified as:

$$x_i(t + \Delta t) = (\Delta t.\alpha_i).\prod_{j=1}^{N}[x_j(t)]^{w_{ij}} + (1 - \Delta t.\gamma_i).x_i(t) + q_i.df \dots (11)$$

Where df is the difference between the original and the predicted expression values of the associated gene at the last sampling instance and q_i is a positive constant. The term $q_i.df$ is an error corrector. Now the total number of parameters to be trained for each gene in the modified half-system formalism is (N + 3).

3.4. Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a meta-heuristic optimization algorithm used to optimize RNN model parameters. PSO consists of a swarm of particles, each of which represents a candidate solution. Each particle i with a position p_i move in the multidimensional problem space with a corresponding velocity v_i. The basic idea of PSO is that each particle randomly searches through the problem space by updating itself with its own memory and the social information gathered from other particles. These components are represented in terms of two best locations during the evolution process: one is the particle's own previous best position, represented as vector p_i^{best}, according to the calculated fitness value, and the other is the best position in the entire swarm, represented as G_{best}. G_{best} is basically the minimum of the p_i^{best}. The corresponding canonical equations for updating particle velocity and particle position are written as,

$$v_i^{t+1} = w \cdot v_i^t + r_1 c_1 \cdot [pbest_i^t - p_i^t] + r_2 c_2 \cdot [gbest^t - p_i^t] \ldots (12)$$

$$p_i^{t+1} = p_i^t + v_i^{t+1} \ldots (13)$$

In equation (12), v_i^{t+1} represents updated velocity vector of particle i in the current generation; w is the inertia weight; v_i^t is the particle velocity vector at time point t; r_1 and r_2 are uniform random numbers in the range of [0,1]; c_1 and c_2 are the cognitive coefficient and social coefficient, respectively; p_i^t is the position vector of the ith particle in the tth generation; $pbest_i^t$ is the ith particle's previous best position vector; and $gbest_i$ is the best position vector in the entire swarm in the tth generation. In equation (13), p_i^{t+1} is the updated position vector of the particle i in the tth iteration. A fitness function, which is used to measure the deviation of the predicted expression data $\widetilde{x_\iota}(t)$ from the original expression data $x_i(t)$, is defined as:

$$mse = \frac{1}{N}\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\left(x_i(t) - \widetilde{x_\iota}(t)\right)^2 \ldots (14)$$

In equation (14), to estimate the deviation of predicted gene expression values from that of the original dataset, we used *mean squared error or mse* as the fitness function.

The basic procedure of PSO-based RNN training can be summarized as follows:

(i) Initialise a population of particles with random positions and velocities of D dimensions.

(ii) Calculate the estimated gene expression time series based on the RNN model and evaluate the optimization fitness function for each particle.

(iii) Update the velocity and position of the particles with equations (5) and (6).

(iv) Return to step (ii) until a stopping criteria is met.

CHAPTER 4: PROPOSED WORK

In the next work of action, our aim is to combine the two modelling techniques. One from the exponential law based technique RNN and another one is the power law based model half-system for the reconstruction of GRN from the time-series gene expression datasets. For perceiving the underlying dynamics of the temporal gene expression profile and also resolving the problems of reverse engineering, this hybridization method becomes effective compared to previous works [reference]; as this method is integrating the advantages of both techniques.

Our target is to use Particle Swarm Optimization (PSO), a swarm intelligent based optimization technique for the estimation of the model parameters. We will implement the proposed methodology on the real experimental datasets (in *vivo*) of the SOS DNA repair network of *E.coli* (8 genes). After that we will implement the above mentioned methodology in DREAM3 challenge network (10 genes) to check the competency of the work. Later on for the validation of the work we will use the traditional statistical metrics.

BIBLIOGRAPHY

[1] M Kathleen Kerr, Mitchell Martin, and Gary A Churchill. Analysis of variance for gene expression microarray data. Journal of computational biology, 7(6):819-837, 2000.

[2] Jerome H Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. Data Mining and Knowledge Discovery, 1(1):55-77, 1997.

[3] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyse expression data. Journal of Computational Biology, v 7(3-4):601–620, 2000.

[4] D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnet-works from perturbed expression profiles. Bioinformatics, 17(suppl_1): S215–S224, 2001

[5] Min Zou and Suzanne D Conzen. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. Bioinformatics, 21(1):71-79, 2004.

[6] Luis M De Campos, Andrés Cano, Javier G Castellano, and Serafín Moral. Bayesian networks classifiers for gene-expression data. In2011 11th International Conference on Intelligent Systems Design and Applications, pages 1200-1206. IEEE, 2011.

[7] J. Hallinan and P. Jackway. Network motifs, feedback loops and the dynamics of genetic regulatory networks. In Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 1-7, 2005.

[8] Stuart A Kaufmann. Metabolic stability and epigenesis in randomly constructed genetic nets. Journal of Theoretical Biology 22(3):437-467, 1969.

[9] Yufei Xiao. A tutorial on analysis and simulation of Boolean gene regulatory network models. Current Genomics, 10(7):511#525, 2009.

[10] Z. Bar-Joseph. Analysing time series gene expression data. Bioinformatics, 20(16), pp. 2493-2503, 2004.

[11] P. D'haeseleer. Reconstructing gene network from large scale gene expression data. Dissertation, University of New Mexico, 2000.

[12] J. Vohradský. Neural network model of gene expression. The FASEB Journal, vol. 15, pp. 846-854, 2001.

[13] Rui Xu, Donald Wunsch II, and Ronald Frank. Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 4(4):681-692, 2007.

[14] Rui Xu, Ganesh K Venayagamoorthy, and Donald C Wunsch II. Modelling of gene regulatory networks with hybrid differential evolution and particle swarm optimization. Neural Networks, 20(8):917-927, 2007.

[15] Kyriakos Kentzoglanakis and Matthew Poole. A swarm intelligence framework for reconstructing gene networks: searching for biologically plausible architectures. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 9(2):358-371, 2012.

[16] Sudip Mandal, Abhinandan Khan, Goutam Saha, and Rajat Kumar Pal. Reverse engineering of gene regulatory networks based on s-systems and bat algorithm. Journal of Bioinformatics and Computational Biology, 14(03):1650010, 2016.

[17] Sudip Mandal, Goutam Saha, and Rajat Kumar Pal. Recurrent neural network based modelling of gene regulatory network using bat algorithm. arXiv preprint arXiv:1509.03221, 2015.

[18] Michael A Savageau. Introduction to s-systems and the underlying power-law formalism. Mathematical and Computer Modelling, 11:546-551, 1988.

[19] Haixin Wang, Lijun Qian, and Edward Dougherty. Inference of gene regulatory networks using s-system: a unified approach. IET Systems Biology, 4(2): 145-156,2010

[20] Leon Palafox, Nasimul Noman, and Hitoshi Iba. Reverse engineering of gene regulatory networks using dissipative particle swarm optimization. IEEE Transactions on Evolutionary Computation, 17(4):577-587, 2013.

[21] Yuji Zhang, Jianhua Xuan, Benildo G de los Reyes, Robert Clarke, and Habtom W Ressom. Reverse engineering module networks by PSO-RNN hybrid modelling. Genomics, 10(1):S15, 2009.

[22] Tor Lattimore and Marcus Hutter. No free lunch versus Occam's razor in supervised learning. In Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence, pages 223-235. Springer, 2013.

[23] V. Gudise and G. Venayagamoorthy. Comparison of particle swarm optimization and back-propagation as training algorithms for neural networks. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium, pp. 110-117, 2003.

[24] Abhinandan Khan, Sudip Mandal, Rajat Kumar Pal, and Goutam Saha. Construction of Gene Regulatory Network using Recurrent Neural Networks and Swarm Intelligence, 2016.