Extraction of Interaction Information among Genes from Gene Expression Time Series Data

Debasish Datta
Dept. of Information Technology, St. Thomas' College of Engineering & Technology, Khidirpur,
Kolkata-700023, India.
debasishresearchpaper@rediffmail.com

Amit Konar
Artificial Intelligence Laboratory, Dept. of Electronics and Telecommunication Engineering, Jadavpur University,
Kolkata-700032, India.
konaramit@yahoo.co.in

R. Janarthanan
Dept. of Information Technology,
Jaya Engineering College,
Chennai – 602 024.
India.
srmjana_73@yahoo.com

***Abstract* — Gene regulatory network gives the idea about the nature of interaction among the genes present in the DNA of a living species. Recently detection of gene regulatory network from gene expression data is of prime interest to the researchers. This paper considers modelling of the Gene regulatory network identification problem using a fuzzy recurrent neural network, and obtains the interaction weights among the neuron using differential evolution algorithm. A cost function is designed, the minimization of which yields the solution to the problem. In order to improve the solution further, a heuristic based local search is proposed. Computer simulation of the proposed inference algorithm revels that it is able to predict the signs of all the existing weights accurately.**

Keywords- Recurrent Neural Network,Differential Evolution,Fuzzy Distribution,Gene expression micro array data

1. INTRODUCTION

At the time of evolution of a new generation of living species from the older one, the genetic information stored in the parent's DNA is copied into child's DNA. The procedure through which genetic information is copied from generation to generation is popularly known as central dogma in molecular biology. An apparent contradiction to central dogma lies in the difference between the physical and the psychological aspects of two children in the same family. Researches believe that there is some biological controlling factor, which controls the central dogma procedure. This is indeed the case. Some genes in parent's DNA influenced by different environmental factors (atmospheric pressure, temperature, acidic condition etc.) controls the behavior of other genes. As a result, some gene of the parent, are expressed in children leading to the presence of a physical characteristic; on the other hand the same gene may remain silent in another child, leading to the absence of that physical characteristic. These interactions among genes form a virtual network, called gene regulatory network (GRN). Recently identification of gene regulatory network is of primary interest to the researchers. This is due to the fact that gene regulatory network is a deciding factor of many biological diseases. For example, if it is possible to detect the controlling genes responsible for cancer, then it may be possible to stop abnormal cell divisions.

Identification of gene regulatory network is a data driven problem. We need data of the expression of the genes to detect the reason of its over- or under- expression. This platform is provided by the recent development of DNA micro-array technology. This technology allows us to investigate thousands of gene samples simultaneously, and with the availability of the gene expression time series data extracted from DNA micro array [6], it becomes possible to infer gene regulatory network by soft computing techniques. There are varieties of micro array platforms that have been developed to accomplish this gene expression profiling. The basic idea is simple: a glass slide or membrane is spotted or "arrayed" with DNA fragments or oligoneucleotides that represent a specific gene-coding region. Purified RNA is then fluorescently or radioactively labeled, and hybridized to the slide/membrane. In some cases, hybridization is done simultaneously with reference RNA to facilitate the comparison of data across multiple experiments. After through washing, the raw data is obtained by laser scanning or auto radiographic imaging. At this point, the data are entered into a database, and analyzed by a number of statistical methods. Researchers attempted different approaches to infer gene regulatory network, including Boolean Networks [7], [8], Linear differential model [9], [10], Bayesian networks [11], [12], Linear additive regulation model [19], [20], and the like. A brief overview of these models is given below.

According to Boolean network, a gene can have two states: 1 for active and 0 for inactive. To specify the state change of a gene due to the interaction of the other genes, Boolean functions are used. Detection of gene regulatory network using Boolean functions is the pioneering idea in this field. The main drawback of this technique is that it ignores the intermediate levels of gene expressions, and because of that it incurs information loss. Also Boolean network presumes that the transition between genes' activation state are synchronous, which is biologically implausible. The investigation of modeling of dynamic behavior of gene regulatory network using this technique is under progress [14], [15].

Bayesian network models the gene regulatory network as a directed acyclic graph, where vertices represent genes, and arcs represent the conditional dependence between genes. This technique is [11], [12] capable of handling noise, incompleteness, and stochastic nature of the gene expression data. Main problem with this approach is that it is unable to model the dynamic aspect of gene regulatory network. To overcome this limitation of Bayesian networks, an alternative approach called Dynamic Bayesian model [16], [17], [18] has been proposed in the literature. Dynamic Bayesian networks are capable of handling the 'hidden variable', 'prior knowledge', and 'missing data'.

Linear additive regulation models [19], [20] infer the gene regulatory network by describing the expression level of a gene at time t as a weighted sum of expression levels of other genes at time (t-1). However linear additive model fails to capture the nonlinear dynamics of gene regulation.

The results obtained so far from the existing models are partially correct. The reason of failure of the existing models to capture the stochastic behavior of the genes accurately is the limited availability of the gene expression time series data. Moreover these data are erroneous and incomplete, and there is no prior knowledge about the topology of the network or about the parameters. Further, weights of the gene regulatory network are multimodal functions of the gene expression time series data. While modeling gene

regulatory network as an optimization problem, researchers attempt to minimize the mean square error obtained by taking the squared difference of the measured time series data, and desired response of the network nodes. Since the error function too is a nonlinear multimodal surface, the solution set of weights is non unique, and naturally the solution does not guarantee the optimal selection of weights of the network. An alternative formulation of the problem is to presume that the interactions among the network nodes have a stochastic/fuzzy distribution. Although there is a scope of multiplicity of such distributions for individual weights, the justification of introducing stochastic/fuzzy distribution lies in searching a narrow space for individual probability or fuzzy memberships in [0,1], thereby minimizing the possibility of multiplicity. In this paper, we propose a model to infer gene regulatory network using fuzzy recurrent neural network (FRNN). In recurrent neural network, the output feeds into processing units. As a result, it provides dynamic aspect, which is most essential for gene regulatory network. The neuron in the FRNN acts as genes of GRN and the weights between the neurons represents the quantitative interaction among genes. The weights in the network are represented using fuzzy membership distribution. In this paper, we used differential evolution algorithm (DE) to find the optimum membership distribution. The evaluated membership distribution is defuzzified using centroidal defuzzification technique, and those defuzzified weights are used to calculate the response of the neurons of the network. This way the training of the FRNN continues until the stopping criteria is meet. Though there are many techniques available to train recurrent neural network, such as back propagation through time we have chosen DE because of its fast convergence rate and simplicity [4]. To improve the results found in each step of the differential evolution algorithm, we used a heuristic based local search technique around the best result found from it. We have tested our inference technique using simulated gene expression data, and real gene expression data, and found almost all signs of the existing weights are detected correctly for a known network.

The remaining part of the paper organizes as follows. Section 2 explains the mathematical model we used here. In section 3, we discussed about a well-known 4-gene network. This 4-gene network is used to test the accuracy of our proposed inference technique. In section 4, we introduce the proposed objective function. Section 5 pictorially describes the used fuzzy distribution of the weights of the inferred network. Section 6 outlines the local search technique around the best individual of DE. In section 7, we introduce the extended differential evolution algorithm used in our paper and show the results obtained. In section 8, we apply our proposed technique to identify the gene regulatory network of S.O.S DNA repair system of bacteria E.Coli using our model.

2. PROPOSED MODEL

The gene expression micro array data available in reality gives us the expression that it changes with time. We assumed that, this change happens according to the differential of the gene expression with respect to time and proposed the mathematical model of equation (1).

$$\overset{*}{T}_i \frac{dx_i}{dt} = f\left(\sum_{j=1}^{N} \overset{*}{w}_{ij}\, x_i - \overset{*}{b}_i\right) - p_i x_i \qquad (1)$$

Here, $\overset{*}{T}_i$ is the i^{th} fuzzy time constant, $\overset{*}{b}_i$ is the i^{th} fuzzy bias term, p_i is the i^{th} decay constant, x_i is the expression of i^{th} gene. $\overset{*}{w}_{ij}$ is the fuzzy interaction value among the i^{th} and j^{th} gene, $f(x) = 1/(1+e^{-x})$. All the fuzzy terms have fuzzy membership distribution. For example, the fuzzy distribution corresponding to $\overset{*}{w}_{ij}$ can be represented using the doublet $\{ \overset{k}{w}_{ij} \mid \mu(\overset{k}{w}_{ij}, t) \}$, k=1, 2, 3… C. C is the cardinality of the set, $\overset{k}{w}_{ij}$ is the k^{th} element of the set, and $\mu(w^k_{ij}, t)$ is its fuzzy membership value at t^{th} iteration For example the w_{23} can be represented as {-30|0.54, -15|0.65, 0.00|0.7, 15|0.8, 30|0.9}. Here, the range of value of weights is assumed to be –30 to 30, and set cardinality C= 5. The right side of the equation (1) demonstrates the characteristics of a recurrent neural network, and the left side of it shows how the expression of a gene changes with time. Therefore the whole equation (1) shows how the expression of a single gene changes in response to the other genes present in the network. The defuzzyfied value of $\overset{*}{w}_{ij}$ can be calculated using the centroidal defuzzyfication technique of expression (2).

$$\overset{*}{w}_{ij}(t) = \frac{\sum_{k=1}^{C} \overset{k}{w}_{ij} \times \mu(\overset{k}{w}_{ij}, t)}{\sum_{k}^{C} \mu(\overset{k}{w}_{ij}, t)} \qquad (2)$$

The bias terms and the time constants are also defuzzified using the same centroidal defuzzification technique as of expression (2). As the actual gene expression data is discrete we change our model as shown in expression (3).

$$\overset{*}{T}_i \frac{dx_i}{dt} = f\left(\sum_{j=1}^{N} \overset{*}{w}_{ij} x_i(t) - \overset{*}{b}_i\right) - p_i x_i(t)$$

$$\Rightarrow \overset{*}{T}_i \frac{x_i(t+\Delta t) - x_i(t)}{\Delta t} = f\left(\sum_{j=1}^{N} \overset{*}{w}_{ij} x_i(t) - \overset{*}{b}_i\right) - p_i x_i(t)$$

$$\Rightarrow x_i(t+\Delta t) - x_i(t) = \frac{\Delta t}{\overset{*}{T}_i} f\left(\sum_{j=1}^{N} \overset{*}{w}_{ij} x_i(t) - \overset{*}{b}_i\right) - p_i x_i(t)$$

$$\Rightarrow x_i(t+\Delta t) = \frac{\Delta t}{\overset{*}{T}_i} f\left(\sum_{j=1}^{N} \overset{*}{w}_{ij} x_i(t) - \overset{*}{b}_i\right) + x_i(t)\left(1 - \frac{\Delta t}{\overset{*}{T}_i} p_i\right) \qquad (3)$$

Expression (3) demonstrates how expression of a gene changes with time in response to other genes present in the network.

3. GENE EXPRESSION GENERATION FOR A KNOWN NETWORK

Before going into direct implementation of gene regulatory network using real world data, we evaluated the accuracy of our model in simulation environment. First, the gene expression time series data is generated for a known network, and then reverse engineering is performed to find out the used known network using only those simulated time series data. The known network used for this purpose is shown below in TABLE I.

TABLE I
USED NETWORK TO SIMULATE GENE EXPRESSION DATA

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	25	-20	0	0	0	10
Gene2	10	-15	0	0	-3	7
Gene3	0	5	-10	0	0	5
Gene4	0	0	-5	10	0	4

The meanings of the values of TABLE I are as follows: from TABLE I we can see cell (2, 1) has value 10; it indicates that gene2 has 10 unit of effect on gene1. Expression (3) is used to generate the simulated gene expression using TABLE I. These generated gene expressions are shown using graph in Fig.1.

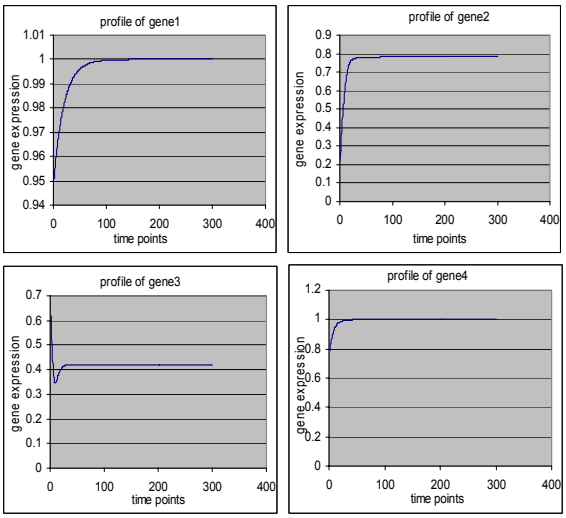


Fig.1. Simulated gene expression using the network of TABLE I.

In Fig.1 we used time duration $\Delta t = 0.5$, we can see that nearly after 100 points the gene expression get saturated. In reverse engineering process [8], we used different time durations, for example Δt =0.1, 0.4, 0.7 etc. The reason is that instead of taking more number of points from one single graph, taking points from different trajectory of graph convey more information. We used this time series data to get back the TABLE I.

4. PROPOSED OBJECTIVE FUNCTION

Cost function or objective function for the problem of detection of gene regulatory network should take into account two aspects, (i) the accuracy of detected topology of the network, (ii) the accuracy of the detected network parameters. Handling these two issues is a difficult task, because there is no prior information available except the gene expression time series data. To handle the first issue we introduce the cost function of equation (4).

$$COST_1 = p\sum_{i=1}^{N}\sum_{j=1}^{N}|w_{ij}^{*}| \qquad (4)$$

Here, w^{*}_{ij} is the defuzzified interaction value between the gene i and gene j, N is the number of genes in the network. The motivation of $COST_1$ came from the observation of experimentally detected gene regulatory networks. Studies of these detected networks reveal that, very few genes in the network regulate a gene. That means among the probable networks for a set of genes the most suitable one is the network, which contains fewer number of interconnections. Therefore $COST_1$ penalize an inferred network according to its number of nonzero interconnections. But introduction of this term has a serious problem. If we try to minimize it, then it will lead the whole system to an all zero solution. To overcome that situation and to get some sort of control over it we used the constant p in equation (4). In this paper we tested our model in a simulated 4-gene network environment, where the interaction values among genes are in the range of –20 to 25. For this case we experimented with the value of p in the range of [0.0001, 0.3]. This range of value for p we chosen by trial and error method for better accuracy of the final result.

To handle the second issue we introduce the cost function of equation (5).

$$COST_2 = \frac{1}{MND}\sum_{i=1}^{N}\sum_{j=1}^{D}\sum_{k=1}^{M}\{(x^{j}{}_{ter_i})_k - (x^{j}{}_{cal_i})_k\}^2 \qquad (5)$$

Equation (5) is basically the square error between the target gene expression and calculated gene expression using our inferred network. Here, $(x^{j}{}_{ter_i})_k$ is the target gene expression of ith gene at jth time point in kth time series, and $(x^{j}{}_{cal_i})_k$ is the calculated gene expression of the same using our inferred network. Both the symbols of reference and calculated expression are extension of the symbol of the gene expression $x_i(t+\Delta t)$ of equation (3). The extra subscripts (i.e. ter, cal, k), and superscript (i.e. j) are used to indicate the whether they are calculated gene expression or reference gene expression, the time series, and the time point they belongs to. M, D, N represents total number of time series used number of data points present in each time series, and total number of genes present in the network. Using $COST_2$ we measure the accuracy of the produced gene expression using our inferred network, and that gives the measure of accuracy of the network parameters.

Our final equation for the cost function or the objective function is shown in equation (6).

$$COST = COST_1 + COST_2$$
$$\Rightarrow COST = p\sum_{i=1}^{N}\sum_{j=1}^{N}|w_{ij}^{*}| + \frac{1}{MND}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\{(x^{j}{}_{ter_i})_k - (x^{j}{}_{cal_i})_k\}^2 \qquad (6)$$

Using the cost function of equation (6), we chose a specific parameter set as the final solution, if it has the minimum cost value among all other available.

5. PROPOSED FUZZY DISTRIBUTION OF THE INTERCONNECTIONS AMONG THE GENES

In this paper, we represented the interconnection among the genes using fuzzy distribution due to the reason that fuzzy distribution reduces the search space of the weights, as described in the introduction section. We used an extended differential evolutionary (DE) algorithm (introduced in section 7) to find the optimum distribution of the fuzzy membership value. Every individual solution of population pool of the DE represents a complete solution of the target gene regulatory network. For example, an individual solution for the 4-gene network introduced in section 3 consists of 24 × 5 = 120 fields (16 fields for the interaction weights among genes, 4 fields for bias terms, 4 fields for time constants, and each field is represented using distribution set of cardinality assumed to be 5). The structure of a candidate solution for the 4-gene network is shown in Fig.2.

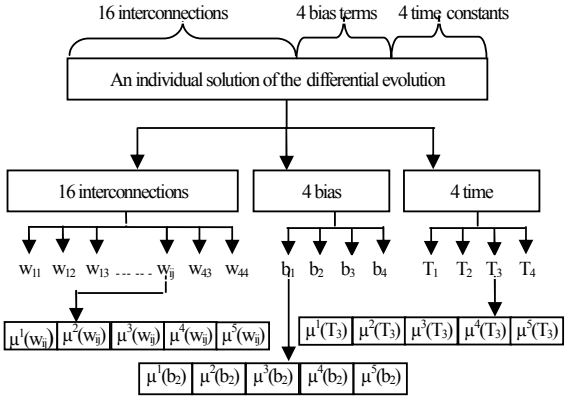


Fig.2. Pictorial representation of an individual solution of DE.

6. USED LOCAL SEARCH

The ideas behind this step is to improve the best solution found by DE by detecting the sparse structure of the network and force the detected nonexistent connection to zero. We perform a local search (ls) to detect the non-existing connections. The algorithm for (ls) is shown using the block diagram of Fig.3.

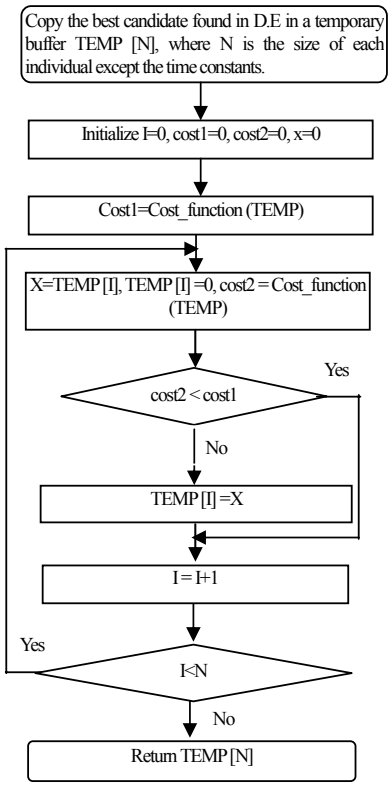


Fig.3. Block diagram of the local search technique.

What we are doing here is that, we are making each connection of the solution zero one after another and checking whether it improves the result or not, if it is then we are persisting with the result otherwise we restoring the previous result. The best solution found by DE is first defuzzified using the expression (2), and then this local search is applied to improve the solution. The detail block diagram of the whole algorithm is shown in Fig 4. Here equation (5) has been used as cost function for the local search. The justification of using equation (5) instead of equation (6) is that, one can see that if we use equation (6) then this local search technique will provide an all zero solution.

7. USED EXTENDED DIFFERENTIAL EVOLUTION AND THE RESULTS

In this paper, we used an extended differential evolution algorithm to find the inferred network from gene expression time series data. Differential evolution algorithm [4] is a well-known population based evolutionary algorithm. A population of solution vectors successively updated using mutation, recombination, and selection until the population converges to the optima. We used an improved initialize technique to distribute the initial population properly in the solution search space, so that some of the initial solution may become close to the original solution of the problem. For this purpose we used a chaos system [2], [3]. The equation of the chaos system is shown in equation (7).

$$R_{k+1} = \Omega R_k - \Omega R_k^2 \qquad (7)$$

Here k =1, 2… α, α is the number of chaotic iterations. R_{k+1} is the value of the chaos at k+1iteration. Ω is the control parameter. R_k takes any value between 0 and 1. When the initial value is chosen as $\Omega = 4$, and $R_0 \neq \{0, 0.25, 0.5, 0.75, 1\}$, the value of R_k distributes with proper irregularity, and randomness. We indeed found that this new initialization technique improve the overall performance of the differential evolutionary algorithm.

In the following we briefly describe the steps of differential evolution algorithm.

(i) **Initialization:** Initialize the population pool with N number of random individuals.

(ii) **Mutation:** For each candidate solution C_i in the population pool, form a mutant vector $V_i = C_{r1} + \lambda (C_{r2} - C_{r3})$ where r1, r2, r3 are three mutually distinct randomly drawn indices from 1 to N and λ is the mutation factor, $0 < \lambda <= 2$.

(iii) **Recombination:** For each candidate solution C_i and corresponding mutant vector V_i form a trial vector α_i as follows: for each component of the candidate solution draw a random number τ_i between 0 and 1. If $\tau_i <= P_r$ then the corresponding component of V_i appears in α_i, otherwise component of candidate solution appears in α_i. P_r is a predefined value and typically $P_r = 0.9$.

(iv) **Selection:** If COST (α_i) is better than COST (C_i) then αi replaces C_i in the next generation otherwise keep C_i. COST () is the defined cost function for a particular problem.

(v) If stop criterion is met then show the result otherwise go to step (ii)

Each individual solution of the population pool of the differential evolution represents a complete solution. The pictorial representation of a complete solution is shown in section 5. Each fields of every solution is described by a fuzzy distribution. We used equation (7) to initialize the population pool, perform the evolution of the population using differential evolution algorithm, defuzzy each field of every candidate solution using expression (2), generate gene expression for every candidate solution using the expression (3), calculate its fitness using the equation (6) and take appropriate decision whether to persist with this individual or not. The whole process is shown in the Fig.4.

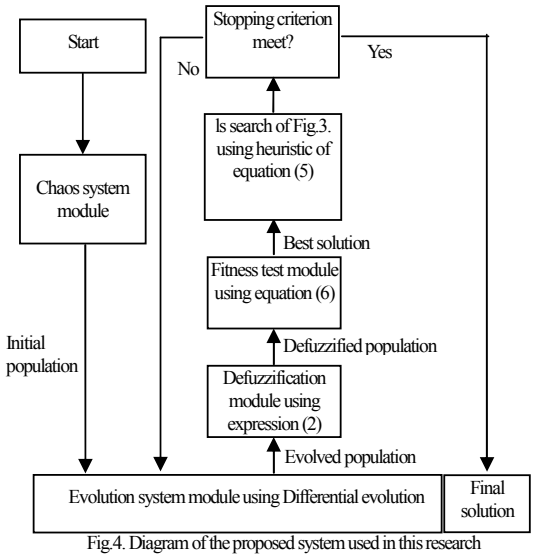


Fig.4. Diagram of the proposed system used in this research

Using our model of Fig.4 we performed the reverse engineering technique on the 4-gene network introduced in Section 3. The results are shown in the TABLE II to TABLE V. Further we have shown a graph in Fig.5, which shows the accuracy of the

detected interconnection among the genes depending upon the cardinality of the distribution of the weight set. For the sake of better understanding we have shown the TABLE I here once again.

TABLE I

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	25	-20	0	0	0	10
Gene2	10	-15	0	0	-3	7
Gene3	0	5	-10	0	0	5
Gene4	0	0	-5	10	0	4

TABLE II.
SIMULATED NETWORK USING 4-TIME SERIES DATA, WEIGHT DISTRIBUTION CARDINALITY C=5, AND TOPOLOGY CONSTANT P = 0.25, POPULATION SIZE=70, DE -ITERATION=3000

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	-3.1	-1.9	-0.39	20.57	-16.5	26
Gene2	0.4	-0.7	-0.2	-2.8	-5.6	14.9
Gene3	8.56	-0.53	16.9	0.2	28.5	7.1
Gene4	-0.85	0.73	15.9	-14.3	-24.9	15.9

TABLE III.
SIMULATED NETWORK USING 4-TIME SERIES DATA, WEIGHT DISTRIBUTION CARDINALITY C = 9, AND TOPOLOGY CONSTANT P = 0.1, POPULATION SIZE = 50, DE -ITERATION = 2500

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	22.9	-4.5	0.0	19.9	-21.3	9.9
Gene2	1.6	7.5	-0.08	8.4	-2.9	13.7
Gene3	-2.99	3	0.0	0.0	9.5	8.6
Gene4	0.0	2.9	-2.8	6.7	0.0	22.6

TABLE IV.
SIMULATED NETWORK USING 4-TIME SERIES DATA, WEIGHT DISTRIBUTION CARDINALITY C =13, AND TOPOLOGY CONSTANT P = 0.5, POPULATION SIZE = 30, DE- ITERATION = 2500

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	5.3	-27.6	0.0	7.6	-0.1	16.2
Gene2	15.4	-10.2	0.0	5.6	22.5	28.6
Gene3	-4.4	0.11	-0.2	-8.1	11.13	26.7
Gene4	0.0	2.7	-8.3	0.0	25.1	7.9

TABLE V.
SIMULATED NETWORK USING 4-TIME SERIES DATA, WEIGHT DISTRIBUTION CARDINALITY C = 19, AND TOPOLOGY CONSTANT P = 0.95, POPULATION SIZE = 35, DE- ITERATION = 3500

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	15.7	-6.5	0.0	0.0	-26.4	30.3
Gene2	4.7	-8.2	0.0	0.0	-14.2	28.9
Gene3	-6.9	0.1	-8.4	1.8	-26.9	18.5
Gene4	0.0	0.0	-6.3	12.9	-24	9.5

For better understanding of the results we presented TABLE I to TABLE V in a different manner once again,

SIGN OF TABLE I

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	+	-	0	0	0	10
Gene2	+	-	0	0	-3	7
Gene3	0	+	-	0	0	5
Gene4	0	0	-	+	0	4

SIGN OF TABLE II.

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	-	-	-0.39	20.57	-16.5	26
Gene2	+	-	-0.2	-2.8	-5.6	14.9
Gene3	8.56	-	+	0.2	28.5	7.1
Gene4	-0.85	0.73	+	-	-24.9	15.9

SIGN OF TABLE III.

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	+	-	0	19.9	-21.3	9.9
Gene2	+	+	-0.08	8.4	-2.3	13.7
Gene3	-2.99	+	0	0	9.5	8.6
Gene4	0.0	2.9	-	+	0.0	22.6

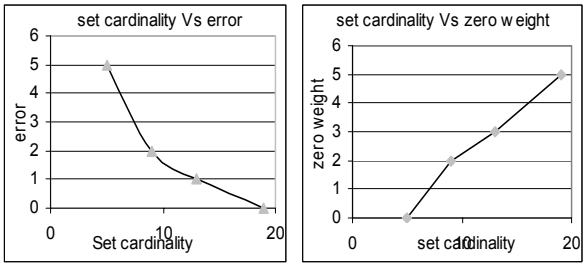
SIGN OF TABLE IV.

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	+	-	0	7.6	-0.1	16.2
Gene2	+	-	-16.2	5.6	22.5	28.6
Gene3	-4.4	+	-	-8.1	11.1	26.7
Gene4	0.0	2.7	-	0	25.1	7.9

SIGN OF TABLE V.

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	+	-	0.0	0.0	-26.4	30.3
Gene2	+	-	0.0	0.0	-14.2	28.9
Gene3	6.9	+	-	1.8	-26.9	18.5
Gene4	0.0	0.0	-	+	-24	9.5

In Fig.5 we have shown a graph to demonstrate the change of error of detection of the network interconnections with respect to the cardinality of the weight distribution set.



(a) (b)

Fig.5. (a) graph of the value with respect to the cardinality of weight distribution, (b) graph of the detected nonzero connection with respect to cardinality of weights

8. SIMULATION OF THE GENE REGULATORY NETWORK OF S.O.S DNA REPAIR SYSTEM OF BACKTERIA E.COLI

We have used our model to infer the S.O.S DNA repair network of bacteria E.Coli [5]. This network consists of nearly 30 genes regulated at the transcription level. Four experiments have been conducted with different UV light intensities. Experiment 1, and 2 using UV= 5 jm^{-2}, and experiment 3, and 4 using UV=20jm^{-2}. Using these experiments, expressions of eight major genes have been documented. These genes are uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA, polB.

E.Coli. S.O.S. DNA repair network works as follows: lexA acts as a balancing factor for the whole network. In absence of any DNA damage it binds to the promoter region of the genes, suppressing the S.O.S genes in the network. When DNA damage occurs, RecA (one of the S.O.S protein) becomes activated. It decreases the level of lexA. As a result the S.O.S genes become activated. Once the damage has been repaired or bypassed, the levels of RecA decrease. As a result, the level of lexA increases, and again deactivates the S.O.S genes. This regulatory mechanism is shown pictorially in Fig.6.

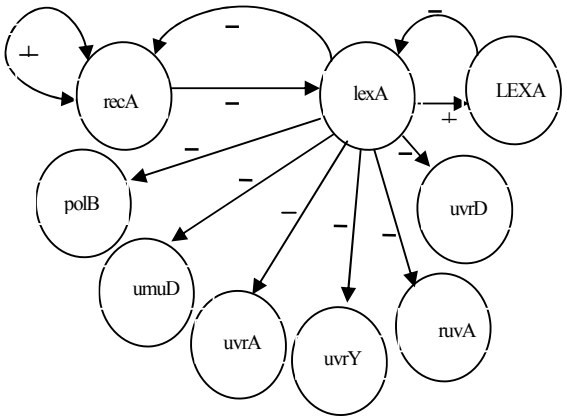


Fig.6. E.Coli S.O.S. DNA repair network, activation is represented by '+' sign, and inhibitation by'-'; genes are written with small letter, and protein with capital letter

Gene profiles of E.Coli S.O.S DNA repair network consists of 50 data points, sampled every 6 minutes. The data set consists of four 8 × 50 matrix. Each column represents the observation of expression value at a particular time instant of eight genes, and each row represents the fifty-expression value of a particular gene at different time instants. This data set is available in the website (http://www.weizmann.ac.il/mcb/UriAlon/). It is one of the best data set, which fits our model. We have conducted the same experiment, as with the above artificial data, and the identified interaction values between genes are represented in TABLE VI.

TABLE VI
IDENTIFIED INTERECTION VALUES OF E.COLI. S.O.S DNA REPAIR NETWORK

	uvrD	lexA	umuD	recA	uvrA	uvrY	ruvA	polB
uvrD	-14.8	6.7	-10.4	17.0	0.0	-5.4	-15.5	16.1
lexA	-14.1	-7.9	0.0	-6.8	-12.3	13.9	1.1	-14.6
umuD	-18.5	-10.2	-24.8	9.2	-17.3	-21.2	-11.2	-20.2
recA	-31.8	-3.8	-11.6	-8.0	-12.8	-7.7	2.6	-16.1
uvrA	4.9	6.7	13.1	17.8	13.0	-20.6	-6.6	12.4
uvrY	-0.2	-7.5	-14.1	5.2	-12.5	0.0	-19.9	3.1
ruvA	-10.0	12.7	0.0	0.0	-7.6	-29.2	19.1	3.9
polB	-2.3	1.5	7.9	-3.6	-6.1	1.7	-18.8	0.0

We have fixed the lower and upper range of the interaction value as -30, and +30; the values of other parameters of the algorithm are the same as used in our model. The result of our algorithm has shown in TABLE VI. Here the reader should keep in mind that the cost function used contains multiple minima, and also the available gene expression time series data contains only 50 data points.

CONCLUSION

This work proposed a fuzzy recurrent neural network based novel technique for inference of gene regulatory network from gene expression time series data. Recurrent neural network provides the dynamic aspect and fuzzy distribution of weight capable of handling the noise. Because of these reasons, the proposed algorithm is able to detect the signs of almost all existent weights and most of the nonexistent weights of the network, which is a challenge to the existent literature [1]. The only information available for inference of gene regulatory network is the gene expression time series data, which are also erroneous, and there is no guideline regarding the structure of a particular gene regulatory network. Hence, the result obtained is not 100% accurate, but considering the challenges, and limited availability of information, our model provides a good result. If more knowledge can be incorporated in the cost function, then the model will be able to provide more accurate results. Our next goal is to incorporate a few more constraints in our cost function to make it more biological accurate.

REFERENCE

[1] Rui Xu, Donald C. Wunsch II and Ronald L. Frank, " Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization" IEEE/ACM transaction on computational biology and bioinformatics,vol.4, No.4,pp 681-692, 2007.

[2] C. Lng, S.Q. Li. Chaotic spreading sequences with multiple access performance better than random sequences. IEEE transaction on Circuit and System -I, Fundamental Theory and Application, 47(3):394-397, 2000.

[3] R.May. Simple mathematical models with very complicated dynamics. Nature, 261: 459 - 467, 1976.

[4] Storn, Rainer, and Kenneth Price. Differential Evolution –A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization11, 1997, pp. 341-359.

[5] B. Perrin, L. Ralaivola, A. Mazurie, S. Battani, J. Mallet, and F.d'Alche´-Buc, "Gene Networks Inference Using Dynamic Bayesian Networks," Bioinformatics, vol. 19, pp. ii138-ii148, supplement 2, 2003.

[6] L. F. A. Wessels, E. P. Van Someren, M.J.T. Reinders, "A comparison of genetic network models," Pac Symp Biocomput, 508-519, 2001.

[7] T. Akutsu, S. Miyano, S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under the Boolean network model," Pac Symp Biocomput, 17-28, 1999.

[8] S. Liang, S. Fuhrman, R. Somogyi, "Reveal, a general reverse engineering algorithm for inference of genetic network architectures," Pac Symp Biocomput, 18-29, 1998.

[9] J. L. Michael De Hoon, S. Imota, K. Kobayashi, N. Ogasawara, S.Miyano, "Inferring gene regulatory networks from time-ordered gene expression data of Bacillus subtills using differential equations," Pac Symp Biocomput, 17-28, 2003.

[10] T. Chen, H. L. He, G. M. Church, "Modeling gene expression with differential equations," Pac Symp Biocomput, 4:29-40, 1999.

[11] N. Friedman, M. Linial, I. Nachman, D. Pe'er, "Using Bayesian network to analyze expression data," J. Comp. Biol., 7, 601-620, 2000.

[12] S. Imoto, T. Gota, S. Miyano, "Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression," Pac Symp Biocomput, 175-186, 2002.

[13] R. Xu, X. Hu, D. Wunsch, "Inference of genetic regulatory networks with recurrent neural network models," Engineering in Medicine and Biology Society, 2004. EMBC 2004. Conference Proceedings. 26th Annual International Conference of the Volume 2, 1-5 Sept. 2004 Page(s):2905 – 2908 Vol.4.

[14] J. Hallinan and J. Wiles, "Evolving Genetic Regulatory Networks Using an Artificial Genome," Proc. Second Asia-Pacific Bioinformatics Conf., vol. 29, pp. 291-296, 2004.

[15] J. Hallinan and J. Wiles, "Asynchronous Dynamics of an Artificial Genetic Regulatory Network," Proc. Ninth Int'l Conf. Simulation and Synthesis of Living Systems, 2004.

[16] D. Husmeier, "Sensitivity and Specificity of Inferring Genetic Regulatory Interactions from Micro array Experiments with Dynamic Bayesian Networks," Bioinformatics, vol. 19, no. 17, pp. 2271-2282, 2003.

[17] B. Perrin, L. Ralaivola, A. Mazurie, S. Battani, J. Mallet, and F. d'Alche´-Buc, "Gene Networks Inference Using Dynamic Bayesian Networks," Bioinformatics, vol. 19, pp. ii138-ii148, supplement 2, 2003.

[18] Y. Tamada, S. Kim, H. Bannai, S. Imoto, K. Tashiro, S. Kuhara, and S. Miyano, "Estimating Gene Networks from Gene Expression Data by Combining Bayesian Network Model with Promoter Element Detection," Bioinformatics, vol. 19, pp. ii227-ii236, supplement 2, 2003.

[19] P. D'haeseleer, "Reconstructing Gene Network from Large Scale Gene Expression Data," PhD dissertation, Univ. of New Mexico, 2000.

[20] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Linear Modelling of mRNA Expression Levels during CNS Development and Injury," Proc. Pacific Symp. Bio computing, pp. 41-52, 1999.