Computational Biology and Bioinformatics
2016; 4(5): 37-44
http://www.sciencepublishinggroup.com/j/cbb
doi: 10.11648/j.cbb.20160405.11
ISSN: 2330-8265 (Print); ISSN: 2330-8281 (Online)

Gene Regulatory Network Inference Using Prominent Swarm Intelligence Methods

Md Julfikar Islam, M. S. R. Tanveer, M. A. H. Akhand

Dept. of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh

Email address:
julfikar_islam@yahoo.com (M. J. Islam), sr.tanvir29@gmail.com (M. S. R. Tanveer), akhand@cse.kuet.ac.bd (M. A. H. Akhand)

To cite this article:
Md Julfikar Islam, M. S. R. Tanveer, M. A. H. Akhand. Gene Regulatory Network Inference Using Prominent Swarm Intelligence Methods. *Computational Biology and Bioinformatics.* Vol. 4, No. 5, 2016, pp. 37-44. doi: 10.11648/j.cbb.20160405.11

Received: November 25, 2016; **Accepted**: December 12, 2016; **Published**: January 16, 2017

Abstract: Genes are the basic blue print of life in an organism containing the physiological and behavioral characteristics. A gene regulatory network (GRN) is a set of genes, or parts of genes, that interact with each other to control a specific cell function. GRN inference is the reverse engineering approach to predict the biological network from the gene expression data. Biochemical system theory based S-System is a popular model in GRN inference and the model is defined with its different parameters. The task of S-System based GRN inference is its parameter estimation which is an optimization problem. Several studies employed Particle Swarm Optimization (PSO) and other pioneer optimization techniques to estimate S-System model. In this paper several prominent swarm intelligence (SI) techniques have been studied and adapted for S-System parameter estimation. They are Group Search Optimizer, Grey Wolf Optimizer and PSO. Proficiency of optimization techniques are compared to infer GRN from SOS DNA real gene expression data and DREAM 4 Silico data.

Keywords: Gene Regulatory Network (GRN), GRN Inference, Swarm Intelligence, S-System Model

1. Introduction

Genes are the basic blue print of life in an organism containing the physiological and behavioral characteristics [1]. There are thousands of genes situated in the DNA double helix, contained in the chromosomes inside cell nucleus in wounded form. Each gene codes for a protein. The protein is produced from a gene through a sequence of bio-chemical reaction when the cell environment is undergone some internal or external change [2]. The process of protein production is called the gene expression and the rate of expression is called the gene expression level. The protein from one gene may affect the expression of any other gene contained in the same cell or in any other cell of the organism. This phenomenon is called the regulation of the first gene to the second. For a set of genes the pairwise regulatory relationship forms the network called Gene Regulatory Network (GRN).

GRN inference is the reverse engineering approach to uncover the dynamic and intertwined nature of gene regulation in cellular systems [3]. In GRN inference the gene regulation network is inferred from gene expression data. The inference of genetic networks faces a great challenge in which mutual interactions among genes are estimated using time-series data of gene expression patterns. The inferred model of a genetic network is considered as an idea tool which helps biologists generate hypotheses as long as facilitate the design of their experiments.

For GRN inference researchers have used a number of approaches [4, 5]. They proposed numerous models to describe biochemical networks have ranged from simple Boolean networks to detailed sets of differential equations of an arbitrary form [6, 7]. Nowadays, S-System model is a popular model derived from biochemical system theory to infer GRN from gene expression data [8, 9].

The GRN inference through S-System means estimation of its parameters values from time-series data of gene expression patterns. S-System has many different parameters and they have different optimal values for a unique set of genes. The number of S-System parameters is proportional to the number of network components. Hence, S-System model based GRN inference is an optimization problem and any

optimization technique can be used. For this reason, researchers have used optimization techniques such as genetic algorithms (GA) [10], global optimization methods [11], Particle Swarm Optimization (PSO) [26], and linear time variant models [12] to estimate parameters of S-System model for different gene expression data.

In recent years nature inspired Swarm Intelligence (SI) has become popular in the field of optimization. PSO is the pioneer SI based optimization technique and used for S-System model [26]. Other recently developed SI based methods might also interesting for S-System model and the aim of this study is to investigate several prominent SI based methods for S-System parameter estimations. In this study, PSO, Group Search Optimizer and Grey Wolf Optimizer are investigated to estimate the S-System parameters.

This paper is organized in the following way. A brief description about S-System based GRN inference technique is given in Section 2. In Section 3, adaptation of selected SI techniques for S-System model is explained. Experimental studies have been discussed in Section 4. Finally the conclusion is drawn in Section 5.

2. S-System Based GRN Inference Technique

A number of approaches have been investigated to infer GRNs from gene expression data with the aim of improving the network inference accuracy and scalability. Basically, the methods can be categorized into two types: information theoretic approaches and model based approaches. In the information theoretic approach, the network is inferred through measuring the dependences or causalities between transcription factors and target genes. A number of prominent methods in this category use Mutual Information (MI) and its variants [13, 14, 15]. On the other hand, in a model based approach nonlinear differential equations are used to express the chemical reaction of transcription, translation and other cellular processes. In model based approach, S-System model is one of the most widely used models for GRN inference.

The S-System model is provided by the Biochemical system theory (BST) [16] to represent and analyze biological systems. The model is a nonlinear differential equation model of GRNs, and it can describe various dynamics of the relationships among genes. It represents a network as a set of differential equations:

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{N} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{N} X_j^{h_{ij}}, \quad (1)$$

where, X_i represents the expression level of the ith gene of the network; N is the number of genes in the network; $\alpha_i, \beta_i \in \mathbb{R}_+^N$ are rate constants; and $g_{ij}, h_{ij} \in \mathbb{R}_+^N$ are kinetic orders. It is to mention that the kinetic orders g_{ij} and h_{ij} regulate the synthesis and degradation of X_i due to X_j. The GRN shown in Figure 1 contains 5 genes.

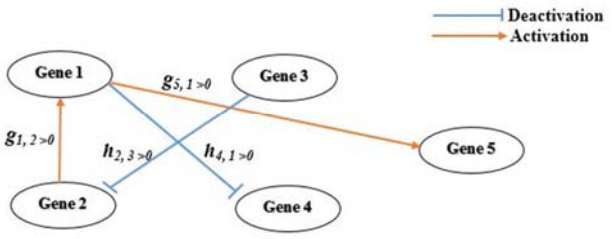


Figure 1. *A GRN with 5 genes.*

In this network the rate constants are $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$ and $\{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5\}$ and the kinetic orders are $\{g_{i,1}, g_{i,2}, g_{i,3}, g_{i,4}, g_{i,5}\}$ and $\{h_{i,1}, h_{i,2}, h_{i,3}, h_{i,4}, h_{i,5}\}$ where i=1 to 5. The kinetic orders g_{ij} and h_{ij} determine the structure of the regulatory network. If $g_{ij} > 0$ gene j induces the synthesis of gene i. If $g_{ij} < 0$ gene j inhibits the synthesis of gene i. Analogously, a positive (negative) value of h_{ij} indicates that gene j induces (suppresses) the degradation of the mRNA level of gene i. So in Figure 1, as $g_{12} > 0$, so gene 2 activates gene 1. As well as, for being $h_{23} > 0$, gene 3 deactivates gene 2.

For the purpose of evaluation of an S-System model for GRN, a numerical solver has to be used such as Runge–Kutta. In the Runge–Kutta method, we have the following differential equation:

$$\frac{dx}{dt} = f(t, x), \quad (2)$$

$$y(t_0) = x_0, \quad (3)$$

where $f(t, x)$ is a nonlinear differential equation. For the case of the S-System

$$f(t, x) = \frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{N} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{N} X_j^{h_{ij}}, \quad (4)$$

Using a fourth-order Runge–Kutta method, we can integrate the solution as

$$X_{n+1} = X_n + \frac{1}{6}(k_1 + k_2 + k_3 + k_4), \quad (5)$$

$$t_{n+1} = t_n + h, \quad (6)$$

where the factors are defined as

$$k_1 = hf(t_n, x_n), \quad (7)$$

$$k_2 = hf(t_n + \frac{1}{2}h, x_n + \frac{1}{2}k_1), \quad (8)$$

$$k_3 = hf(t_n + \frac{1}{2}h, x_n + \frac{1}{2}k_2), \quad (9)$$

$$k_4 = hf(t_n + \frac{1}{2}h, x_n + \frac{1}{2}k_3), \quad (10)$$

The numerical solutions often take a long time to calculate given that (4) depends on each of the N variables to do an update when the networks are large. On the other hand, to optimize the parameters with an optimization technique, it is necessary to evaluate the S-System for multiple candidates and multiple iterations. Linear Lagrange polynomial was used by Tsai and Wang [8] to reduce the calculation's time

by introducing experimental while they used the Runge–Kutta method for solving the S-System of a network. Thus, each new step in the numerical solution of the S-System is defined as:

$$X_{n+1} = X_n + 0.5 * \eta * (g\left[X_{n+1,exp}, \theta\right] + g[X_n, \theta]), \quad (11)$$

where the $X_{exp}(t)$ on the right side of the equation represents the value of the experimental values at time t and $g[x, \theta]$ is evaluated at x with parameters θ according to equation (1). The parameter η denoted the smoothness rate which is to set small, as a result the approximation does not overshoot. In this paper, for notation convenience θ is named to the set of all parameters $\{g_{ij}, h_{ij}, \alpha_i, \beta_i | i, j \in 1 \dots N\}$ in the S-System. With a view to decreasing the calculations and the inference time, equation (11) is used instead of the update in (5). Instead of performing the joint solution of all genes in the classical Runge–Kutta method in (4), the experimental values are used for the solution of $x(t)$ so that the system can solve each gene's solution independently.

Since the aim of this study to find the best parameters θ for the network, it is necessary to formulate it as an optimization problem. Tominaga *et al.* [17] standardized the use of the mean squared error (MSE) evaluation to measure a candidate's fitness in the S-System. Thus, the fitness function is

$$f = \sum_{t=1}^{T} \sum_{i=1}^{N} \left(\frac{X_{i,cal}(t) - X_{i,exp}(t)}{X_{i,exp}(t)} \right)^2, \quad (12)$$

where, T represents the number of time samples in the experimental data, N is the number of genes, and ***cal*** and ***exp*** refer to the calculated and experimental values of the gene expression's data, respectively. Since different networks can have the same time series data, in this study experimental time series have been divided in M sets. This will force the parameters θ to recreate the same dynamics at different initial points, which helps to infer the true parameters of the network. Each of the N genes' time expression is divided on M sets, which will create $N \times M$ training sets for the system. In this study, the values of the parameters θ have been inferred using several SI techniques with regularization parameters.

A regularization term $\lambda E(\theta)$ is often introduced in an error function to avoid over fitting as well as to restrict the search space. It achieves this by restraining the growth of the parameters. There are different regularization terms, which are chosen according to the data. Ng [18] showed that an L1 regularizer is a good choice when trying to infer sparse parameters in a logistic regression. Thus, the regularization term for each of the N genes i will take the form,

$$\lambda E_i(\theta^*) = \lambda \sum_{i,j=1}^{N} |g_{ij} + h_{ij}|, \quad (13)$$

where, the parameters θ^* for large GRNs. The L1 regularization is only used on sparse parameters. It has been applied to the h_{ij} and g_{ij} parameters, while the parameters α and β are left without regularization. Thus, with the regularization term and the decoupling included the following optimization function is defined for each gene i as

$$f = \sum_{t=1}^{T} \sum_{i=1}^{N} \left(\frac{X_{i,cal}(t) - X_{i,exp}(t)}{X_{i,exp}(t)} \right)^2 + \lambda E_i(\theta^*), \quad (14)$$

where, M represents the different time series in which the data have been divided and $X_{i,cal}(t)$ is calculated using (11). In this study S-System has been decoupled so that this value has been calculated for each individual gene. The problem then has two objective functions, where the parameters $\{h_{ij}, g_{ij}\}$ have been constrained to be small and at the same time to have the best MSE fitness for the original time series data.

In this study, experimental time series have been divide in M sets because different networks can have the same time series data. As a result, this has forced the parameters θ to recreate the same dynamics at different initial points and helped us to infer the true parameters of the network. Each of the N genes' time expression is divided on M sets and they create $N \times M$ training sets for the system.

3. Adaption of Prominent SI Techniques for S-System Model

In general, the GRN inference problem is formulated as a function optimization problem to minimize the sum of the squared relative error by Tominaga et al. [17] in (12). Where $X_{i,exp}(t)$ is an experimentally observed gene expression level at time t of the ith gene, $X_{i,cal}(t)$ is a numerically computed gene expression level acquired by solving (1). Where, N is the number of components in the network and T is the number of sampling points of observed data. Since $2N(N+1)$ S-System parameters need to be determined in order to solve (1), so this function optimization problem is $2N(N+1)$ dimensional.

In this study, Particle Swarm Optimization, Group Search Optimizer and Grey Wolf Optimizer are investigated to estimate the S-System parameters. Table 1 demonstrates the parameters for network with 5 genes as shown in Figure 1. For 5 network components the dimension of the problem (for N=5) is D= 2×N (N+1) = 2×5 (5+1) = 60. Therefore, it is required to optimize 60 parameter values for network presented in Figure 1. It is notable that number of parameters will be larger for network with more genes. The following section briefly describe the SI based methods and their adaptation to estimate S-System parameters.

Table 1. *S-System parameters for GRN with 5 genes.*

1	**..**	**5**	**6**	**..**	**10**	**11**	**..**	**35**	**36**	**..**	**60**
α_1	..	α_5	β_1	..	B_5	g_1	..	g_{25}	h_1	..	h_{25}

3.1. Particle Swarm Optimization (PSO)

PSO was proposed by Eberhart and Kennedy [21]. The algorithms is evaluated according to the idea of swarm intelligence based on the swarming habits by certain kinds of animals (such as birds and fish). The basic operations of PSO are performed simultaneously maintaining several candidate solutions in the search space. During each iteration of the

algorithm, the fitness of each candidate solution is determined by the objective function being optimized. In PSO, each particle represents a potential solution. At every iteration each particle moves to a new position (i.e., search a new point) based on the calculated velocity.

In PSO, each particle updates position based on the calculated velocity comparing the best solution of population and its own best solution. Population of particles is distributed uniformly for multi-dimension search space optimization problem. Equation (15) is to calculate velocities of particles and (16) to update positions.

$$V_j^{t+1} = W * V_j^t + R_1 * C_1\left(P_j - X_j^t\right) + R_2 * C_2\left(P_g - X_j^t\right) \quad (15)$$

$$X_j^{t+1} = X_j^t + V_j^{t+1} \quad (16)$$

In (1) the global best location is denoted by P_g and P_j represents the best location ever encountered by this particle. An inertia weight W is included in (1) to avoid the swarm being trapped into a local minimum. Both $C_1$and C_2 are learning parameters and R_1, R_2 are random parameters in a range of [0, 1].

In PSO the position of each particle is considered as the solution of the problem. So that to adapt PSO for S-System model, each particle position X_j^t has been adapted according to Table 1. As well as the velocity will be adapted according to the equations (15) and (16). Finding the best particle position in PSO denotes the estimation of proper S-System model. In every particle, value of each dimension, must be within the pre-defined range. For getting the best particle fitness function is shown in (14).

3.2. Group Search Optimizer (GSO)

GSO is a novel optimization technique developed inspired by animal searching behavior [22]. Animal searching behavior may be described as an active movement through which animal can find resources such as foods, mates, nesting sites. One major consequence of living together is that it is group searching strategy which allows group members to increase patch finding rates. Simply this has led to the adoption of two foraging strategies within groups which are 1) producing, e.g., searching for food; and 2) scrounging, e.g., joining resources uncovered by others. The second one is also referred to as conspecific attraction, kleptoparasitism.

In GSO algorithm population is called a group and each individual animal is called a member. In an *n* dimensional search space, the *i*th member at the *k*th searching bout has a current position which is $x_i \in R^n$ and a head angle $\theta_i^k = \left(\theta_{i_1}^k, \ldots., \theta_{i_{(n-1)}}^k\right) \epsilon R^{n-1}$. The search direction of the *i*th member is a unit vector $D_i^k(\theta_i^k) = \left(d_{i_1}^k, \ldots., d_{i_{(n)}}^k\right) \epsilon R^n$ which is measured from θ_i^k polar to Cartesian coordinate transformation.

$$d_{i_1}^k = \prod_{q=1}^{n-1} \mathrm{Cos}\left(\theta_{i_q}^k\right),$$

$$d_{i_j}^k = \mathrm{Sin}\left(\theta_{i_{(j-1)}}^k\right).\prod_{q=j}^{n-1} \mathrm{Cos}\left(\theta_{i_q}^k\right) \ (j = 2, \ldots., n-1),$$

$$d_{i_n}^k = \mathrm{Sin}\left(\theta_{i_{(n-1)}}^k\right) \quad (17)$$

Practically in a 3-D search space, if at the *k*-th searching bout, the *i*th member's head angle is $\theta_i^k = (\pi/3, \pi/4)$, then the search direction unit vector is $D_i^k = (1/2, \sqrt{6}/4, \sqrt{2}/2)$. Each iteration, a group member located in the most promising area and which confers the best fitness value chosen as the producer. It then stops and scans the environment to seek resources actually that is the optima. At the *k*th iteration producer x_p will scan at zero degree in (18) and then scan laterally by randomly sampling other two points in right according to (19) and left by (20).

$$x_z = x_p^k + R_1 l_{max} D_p^k(\theta^k), \quad (18)$$

$$x_r = x_p^k + R_1 l_{max} D_p^k(\theta^k + R_2 \varphi_{max}/2), \quad (19)$$

$$x_r = x_p^k + R_1 l_{max} D_p^k(\theta^k - R_2 \varphi_{max}/2), \quad (20)$$

$R_1 \epsilon R^1$ is a normally distributed random number with mean 0 and standard deviation 1 and $R_2 \epsilon R^{n-1}$ is random number sequence in the range (0, 1). If the resource of the best point among the three is better than producer's current position, then it will fly to that point; otherwise it will have to stay in its present position and turn its head to a new randomly generated angle, $\theta^{k+1} = \theta^k + R_2 \alpha_{max}$. If the producer fails to find a better area after *a* iterations, it will have to turn its head back to zero degree, $\theta^{k+a} = \theta^k$. In GSO, some group of members are selected as scroungers and they try to get opportunities from the producer. The random walk toward the producer of the *i*th scrounger is modeled in (21) at *k*th iteration.

$$x_i^{k+1} = x_i^k + R_3 o\left(x_p^k - x_i^k\right), \quad (21)$$

$R_3 \epsilon R^n$ is constant in a uniform random sequence which range is (0, 1). In this equation "o" is the Hadamard product, which calculates the entry wise product of the two vectors. A few worst members in GSO are considered as dispersed or ranger members who perform random walks. At the *k*-th iteration, ranger generates a random head angle θ_i; chooses a random distance $L_i = a.R_1 L_{max}$ and move to the new point using (22).

$$x_i^{k+1} = x_i^k + L_i D_i(\theta^{k+1}), \quad (22)$$

In GSO the expected solution is denoted through the efficient location X_i^{k+1} of the producer. For GRN inference, GSO has been adapted to S-System model through the parameter according to the declaration of Table 1. The best location found by the producer is identified by (14).

3.3. Grey Wolf Optimizer (GWO)

GWO is the most recently developed SI technique based on the hunting behaviors of Grey wolf (*Canis lupus*) which belongs to *Canidae* family [23]. It is seen in nature that Grey

wolves prefer to live in a pack. The main phases of their hunting are i) tracking, chasing, and approaching the prey; ii) Pursuing, encircling, and harassing the prey until it stops moving; and iii) attack towards the prey. In the pack, wolves are categorized into several different types and each type perform specific task.

In GWO, the wolf with fittest solution mark as the alpha (α). Consequently, the second and third best ones are named beta (β) and delta (δ) respectively. The rest of the candidate solutions are assumed to be omega (ω). In the GWO, the hunting (optimization) is guided by *α, β,* and *δ*. The *ω* wolves have to follow these three types of wolves. During hunting Grey wolves encircle the prey. The mathematical model of encircling behavior is illuminated through the following (23) and (24).

$$\vec{d} = \left|\vec{c}.\overrightarrow{y_p}(t) - \overrightarrow{y_p}(t)\right|, \quad (23)$$

$$\vec{y}(t+1) = \vec{y}_p(t) - \vec{A}.\vec{d}, \quad (24)$$

Here, t shows the current iteration, $\vec{A}$ and $\vec{C}$ are both coefficient vectors, $\vec{y}_p$ represents the position vector of the prey, and the position vector of a grey wolf is indicated by $\vec{y}$. The vectors $\vec{A}$ and $\vec{C}$ are calculated through (25) and (26).

$$\vec{A} = 2\vec{\psi}.\vec{R}_1 - \vec{\psi}, \quad (25)$$

$$\vec{C} = 2.\vec{R}_2, \quad (26)$$

In (25) and (26), component $\vec{\psi}$ is decreased from 2 to 0 linearly over the course of iterations and $\vec{R}_1$, $\vec{R}_2$ are both random vectors in [0, 1]. The mathematical simulation of the hunting behavior of grey wolves illustrates that the alpha, beta, and delta have fair knowledge about the potential location of prey. As a result, the first three best solutions obtained so far are saved and also oblige the other search agents (including the omegas). The following equations are used to fulfill this purpose.

$$\vec{d}_\alpha = \left|\vec{C}_1.\vec{y}_\alpha - \vec{y}\right|, \vec{d}_\beta = \left|\vec{C}_2.\vec{y}_\beta - \vec{y}\right|, \vec{d}_\delta = \left|\vec{C}_3.\vec{y}_\delta - \vec{y}\right|, \quad (27)$$

$$\vec{y}_1 = \vec{y}_\alpha - \vec{A}_1.\left(\vec{d}_\alpha\right), \vec{y}_2 = \vec{y}_\beta - \vec{A}_2.\left(\vec{d}_\beta\right),$$

$$\vec{y}_3 = \vec{y}_\delta - \vec{A}_3.\left(\vec{d}_\delta\right) \quad (28)$$

$$\vec{y}(t+1) = \frac{\vec{y}_1 + \vec{y}_2 + \vec{y}_3}{3} \quad (29)$$

According to GWO the best solution is the position of alpha wolf $\overrightarrow{y_\alpha}$. And to adapt GWO for S-System model, every position of wolf has been adapted according to the Table 1. The best position of alpha is identified by the fitness function in (14). The beta and gamma wolf also identified by the same fitness function.

4. Experimental Studies

This section gives experimental settings S-System model parameters and gene expression data. After that settings of PSO, GSO and GWO are explained. Finally experimental results have been presented and discussed accordingly.

4.1. Gene Expression Benchmark Datasets

In this study, both synthetic and real gene expression benchmark data are considered. The gene expression data is available in a two dimensional matrix form in which each column represents an individual gene and each row represents the expression level of all genes within an experiment. Table 2 shows the brief description of the datasets which shows a considerable variety in the number of types, gene number, series, and sample size.

Table 2. *Benchmark datasets for GRN inference.*

Network Name	Gene Size	Series	Samples	Type	Source
SOS DNA network	8	4	50	Real	Uri Alon [24]
InSilico_Size10_1	10	5	21	Synthetic	DREAM4 [25]

SOS DNA network is a well-known real genetic network published by the Uri Alon [24] group. It is the time series data of different multi array experiments. In their experiments, eight genes are expressed (uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA, and polB). They irradiate their DNA with ultraviolet light, which will affect some genes, and the network will repair itself, thus expressing auto regulation. They did four experiments for different light intensities. Each experiment had 50 time steps spaced by 6 min. During each time step, they take measures of the eight genes.

In DREAM*4* [25] the datasets are provided for InSilico Network Challenge. The goal of the in silico network challenge is to reverse engineer gene regulation networks from given in silico gene expression datasets. Network topologies are obtained by extracting subnetworks from transcriptional regulatory networks of E. coli and S. cerevisiae. They adapted the subnetwork extraction method to preferentially include parts of the network with cycles. Auto-regulatory interactions were removed, i.e., there are no self-interactions in the in silico networks. All networks and data are generated with version 2.0 of GNW. There are three different datasets provided in this challenge. They are InSilico_Size10, InSilico_Size100, and InSilico_Size100_ Multifactorial.

In silico network challenge the dataset InSilico_Size_10 contains the gene expression data of the network that contains 10 genes. There are 5 different datasets (InSilico_Size_10_1, InSilico_Size_10_2, InSilico_Size_10_3, InSilico_Size_10_4 and InSilico_Size_10_5). In this study InSilico_Size_10_1 dataset has been used.

4.2. Experimental Setup

In this study search region has been set in the range of [−2, 2] for the kinetic orders g_{ij}, h_{ij} and [0, 5] for the rate constants α and β. The search region is set in such a manner that the search will be fast to converge. Population size and

generation are set to 40 and 5000 respectively to draw the comparison among these algorithms. In PSO acceleration coefficients are 2.0 and the inertia weight ω is in range [0. 2, 0.9]. In GSO the initial head angle of each individual, θ^0 is set to be $(\pi/4,..., \pi/4)$. The constant a was given by $round(\sqrt{n+1})$ where n is the dimension of the search space. The maximum pursuit angle φ_{max} is π/a^2. The maximum turning angle α_{max} is set to be $\varphi_{max}/2$. The maximum pursuit distance l_{max} is calculated from the following equation:

$$l_{max} = \|\mathrm{U} - \mathrm{L}\| = \sqrt{\sum_{i=1}^{n}(U_i - L_i)^2}, \quad (30)$$

where, L_i and U_i are the lower and upper bounds for the ith dimension. In this study the result is taken from the best among 5 trials. The experiments have been done on a PC (Intel Core i7 @4.40GHz CPU, 8GB RAM, Windows 7 OS, MATLAB 2015).

4.3. Evaluation Technique

To evaluate the performance any GRN inference techniques the inferred network is compared with the true network parameters. Also when the true network parameters are not given then the network is inferred from time series gene expression data. After that, time series data is generated from the new inferred network. Finally the newly generated time series data is matched with the corresponding previous time series data. If the two datasets get matched then the network is inferred correctly.

The performance evaluated by receiver operator characteristic (ROC) curve. In general, ROC curve is a graphical tool for depicting true hit rate along the vertical axis (the number of target events correctly classified as targets) as compared to false alarm rate along the horizontal axis (the number of target events incorrectly classified as non-targets). In GRN inference evaluation, the ROC curve is created by plotting the fraction of true positive rate (i.e., true positives out of the total actual positives) vs. false positive rate (i.e., the fraction of false positives out of the total actual negatives), at various threshold settings. The following equations are used to calculate true positive rate (*TPR*) and false positive rate (*FPR*).

$$TPR = TP/(TP + FN) \quad (31)$$

$$FPR = FP/(FP + TN) \quad (32)$$

Here TP=True Positive (i.e., links are correctly identified), FP=False Positive (i.e., identified links are not correct), TN=True Negative (i.e., correctly identified that there is no links between genes), FN=False Negative (i.e., failed to identify links between genes).

4.4. Evaluation on SOS DNA Real Gene Expression Data

To infer SOS DNA network, at first the S-System parameters have been estimated from time-series data provided by the Uri Alon Laboratory [24]. Among the 8 genes, 2 genes (uvrY, ruvA) have little involvement in regulations [26]. Therefore, this study considered 6 genes as like study of Leon *et al.* [26] and actual network is shown in Figure 2.

The S-System parameters are estimated first then the network is reconstructed. The kinetic orders g_{ij} and h_{ij} determine the structure of the regulatory network. In the case $g_{ij} > 0$, gene j induces the synthesis of gene i. If $g_{ij} < 0$, gene j inhibits the synthesis of gene i. analogously, a positive (negative) value of h_{ij} indicates that gene j induces (suppresses) the degradation of the mRNA level of gene i. Now the graphical representation of the SOS network and the networks inferred by the SI techniques are represented in Figure 3. The performance evaluation of the algorithms used in this study on the SOS DNA dataset are shown in the Table 3. And the ROC plot is shown in Figure 4.

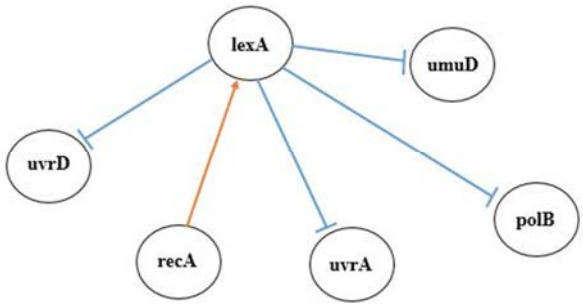
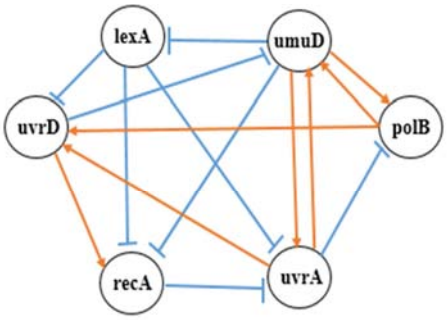


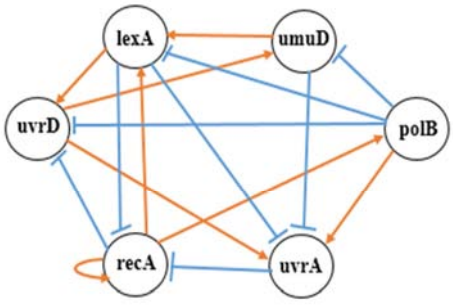
Figure 2. *The graphical representation of the actual SOS DNA network [24].*

Table 3. *Summary of inferred connections through GSO, GWO, PSO for SOS DNA Network.*

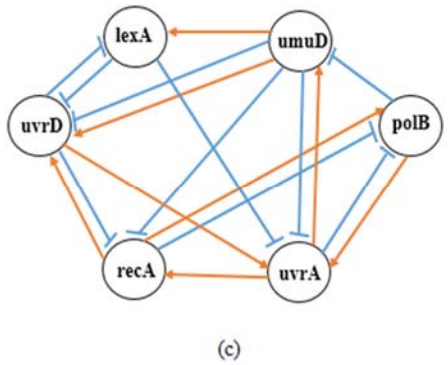
Connection Status	True Network	GSO	GWO	PSO
True Positive (TP)	5	2	2	2
True Negative (TN)	31	18	17	18
False Positive (FP)	-	14	14	13
False Negative (FN)	-	2	3	3



(a)



(b)



(c)

Figure 3. *Inferred SOS DNA networks by (a) PSO, (b) GSO, (c) GWO.*

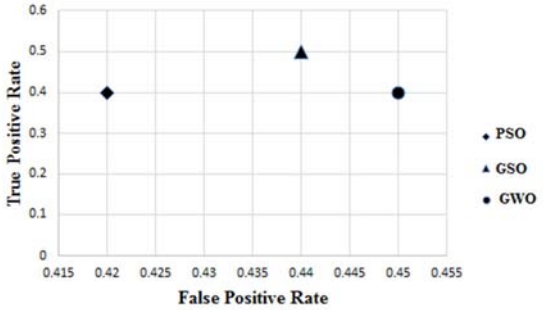


Figure 4. *ROC plot of the results in the SOS DNA network.*

From Table 3 it is observed that TP and TN in SOS DNA true network are 5 and 31, respectively. In experimental result, it is seen that all the methods identified TP as 2; GSO and PSO identified TN as 18; and GWO showed TN 17. For good inference of GRN, FP and FN are expected to minimum. From this point of view, PSO and GSO are competitive and GWO is worse. To visualize these result ROC plot has been drawn in Figure 4, where the *x*-axis represents the *FPR* and the *y*-axis represents the *TPR*. In the ROC plot, the algorithm having a high *TPR* and a low *FPR* GSO is the best in SOS DNA experiment.

4.5. Evaluation on DREAM4 Data

Synthetic DREAM4 InSilico_Size10_1 dataset is for 10 genes and is relatively larger than SOS DNA. Table 4 shows the summary of true and inferred networks. From Table 4 it is observed that, in InSilico_Size10_1 true network, TP is 15 and TN is 75. It is noticeable that GSO identified all true connections showing TP value as 15. On the other hand, GSO and PSO showed TP values 10 and 12, respectively. Moreover, FN is zero for GSO; whereas FN values for GWO and PSO are 5 and 3, respectively. From the ROC plot drawn in Figure 5, GSO is also shown the best among the three SI methods for DREAM4 data.

Table 4. *Summary of inferred connections through GSO, GWO, PSO for InSilico_Size10_1.*

Connection Status	True Network	GSO	GWO	PSO
True Positive (TP)	15	15	10	12
True Negative (TN)	75	6	17	17
False Positive (FP)	-	69	58	58
False Negative (FN)	-	0	5	3

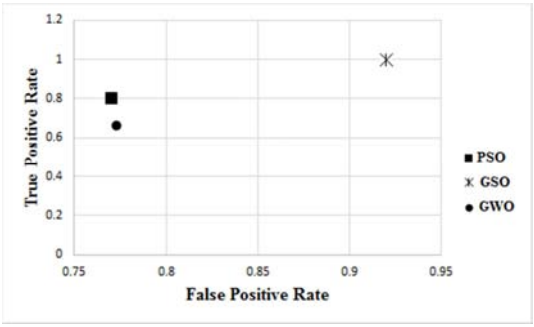


Figure 5. *ROC plot of the results in the InSilico_Size10_1.*

5. Conclusions

Network systems are easy to understand and visualize the genetic interactions in cell organisms. So GRN is a significant way to represent the gene regulations. S-System model is considered as an effective way of GRN inference. S-System parameter estimation as an optimization task and three prominent SI based methods are investigated in this study for to estimate its parameter. In this study both real (i.e., SOS DNA) and synthetic (i.e., DREAM4 InSilico_Size10_1) datasets have been used for S-System model based GRN inference. Among these techniques GWO has shown fair performance in both datasets for GRN inference. Finally, the study revealed the scope of SI based optimization in GRN inference through S-System parameter estimation.

References

[1] Berg, J. M., Tymoczko, J. L. and Stryer, L.(2002). Biochemistry, 5th Edition.

[2] Pierce, B. A. (2010). Genetics: A Conceptual Approach, 4th edition.

[3] Akhand, M. A. H., Nandi, R. N., Amran, S. M. and K. Murase (2015). Gene Regulatory Network Inference using Maximal Information Coefficient. I*nternational Journal of Bioscience, Biochemistry and Bioinformatics,* vol. 5, no. 5, pp. 296-310.

[4] Bower, J. M. and Bolouri, H. (2001). Computational modeling of genetic and biochemical networks. MIT Press.

[5] Jong, H. de (2002). Modeling and simulation of genetic regulatory systems: a literature review. *J. Comp. Bio.*, vol. 9, no. 1, pp. 67–103.

[6] Akutsu, T., Miyano, S. and Kuhara, S. (2000), "Inferring qualitative relations in genetic networks and metabolic pathways", Bioinformatics, vol. 16, pp. 727–734.

[7] Chen, T., He, H. L. and Church, G. M. (1999), Modeling gene expression with differential equations. Proc. Pac. Symp. Biocomput., 4, 29–40.

[8] Tsai, K. Y. and Wang, F. S. (2005). Evolutionary optimization with data collocation for reverse engineering of biological networks. *Bioinformatics*, vol. 21, no. 7, pp. 1180-1188.

[9] Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K. and Tomita, M. (2003). Dynamic modeling of genetic networks using genetic algorithm and S-system. *Bioinformatics*, vol. 19, pp. 643–650.

[10] Ueda, T., Ono, I. and Okamoto, M. (2002). Development of system identification technique based on real-coded genetic algorithm. *Genome Inform.* vol. 13, pp. 386–387.

[11] Noman, N. and Iba, H. (2008). Accelerating differential evolution using an adaptive local search. *IEEE Trans. Evol. Computat.*, vol. 12, no. 1, pp. 107–125.

[12] Liu, P. K. and Wang, F. S. (2008). Inference of biochemical network models in S-system using multiobjective optimization approach. *Bioinformatics*, vol. 24, no. 8, pp. 1085-1092.

[13] Butte, A. J. and Kohane, I. S. (2000). Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing*, vol. 5, pp. 418-429.

[14] Meyer, P. E., Kontos, K. and Bontempi, G. (2007). Biological Network Inference Using Redundancy Analysis, Bioinformatics Research and Development. *Lecture Notes in Computer Science*, vol. 4414, pp. 16-27.

[15] Faith, J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., et al. (2007). Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PloS Biology*, 5 (1), e8 (13 pages).

[16] Savageau, M. A. (1969). Biochemical systems analysis: I. Some mathematical properties of the rate law for the component enzymatic reactions. *J. Theoretic. Biol.,* vol. 25, no. 3, pp. 365–369.

[17] Tominaga, D., Koga, N. and Okamoto, M. (2000). Efficient numerical optimization algorithm based on genetic algorithm for inverse problem. in *Proc. Genet. Evol. Computat. Conf.*, vol. 251. 2000, pp. 251- 258.

[18] Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. *in Proc. 21st ICML*, pp. 78–86.

[19] Noman N. and Iba, H. (2005). Inference of gene regulatory networks using S-system and differential evolution. *in Proc. Conf. Genet. Evol. Computat.* pp. 439-446.

[20] Gonzalez, O. R., K¨uper, C., Jung, K., Naval, P. C. and Mendoza, E. (2007). Parameter estimation using simulated annealing for S-system models of biochemical networks. *Bioinformatics*, vol. 23, no. 4, pp. 480–486.

[21] Kennedy, J. and Eberhart, R. C. (1995). Particle swarm optimization. In Proceedings of the 1995 IEEE International Conference on Neural Networks, vol. 4, pp. 1942-1948.

[22] He, S., Wu, Q. H., and Saunders, J. R. (2009). Group Search Optimizer: An Optimization Algorithm Inspired by Animal Searching Behavior. IEEE Transactions On Evolutionary Computation, vol. 13, no. 5, pp. 973-990.

[23] Mirjalili, S., Mirjalili, S. M., Lewis, A. (2014). Grey Wolf Optimizer. Advances in Engineering Software, vol. 69, pp. 46–61.

[24] Ronen, M., Rosenberg, R., Shraiman, B. I. and Alon, U. (2002). Assigning numbers to the arrows: Parameterizing a gene regulation network by using accurate expression kinetics. Proc. Nat. Acad. Sci., vol. 99, no. 16, p. 10555-60.

[25] DREAM*4* In Silico Network Challenge, Available: http://dreamchallenges.org/projectclosed/dream4 [Accessed: 27- Apr- 2016].

[26] Palafox, L., Noman, N., and Iba, H. (2013). Reverse Engineering of Gene Regulatory Networks Using Dissipative Particle Swarm Optimization. IEEE Trans. on Evolutionary Computation, vol. 17, no. 4, pp. 577-586.