Volume 4, Issue 1, January 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Biclustering of Gene Expression Using Glowworm Swarm Optimization and Neuro-Fuzzy Discriminant Analysis

K.Sathishkumar*, M.Ramalingam
Asst.Professor in Information Technology,
Gobi Arts & Science College(Autonomous),
Gobichettipalayam, TamilNadu, India

Dr.V.Thiagarasu
Associate Professor in Computer Science,
Gobi Arts & Science College(Autonomous),
Gobichettipalayam, TamilNadu, India

Abstract--- The advent of DNA microarray technologies has revolutionized the experimental study of gene expression. Biclustering is the most popular approach of analyzing gene expression data and has indeed proven to be successful in many applications. In recent years, several biclustering methods have been suggested to identify local patterns in gene expression data. Most of these algorithms represent greedy strategies that are heuristic in nature: an approximate solution is found within reasonable time bounds. The quality of a biclustering, though, is often considered more important than the computation time required to generate it. To overcome the problems in gene expression analysis, novel algorithms for finding the coregulated clusters, dimensionality reduction and clustering have been proposed. The coregulated clusters are determined using biclustering algorithm, so it is called as coregulated biclusters. The coregulated biclusters are two or more genes which contain similarity features. The dimensionality reduction of microarray gene expression data is carried out using Neuro- fuzzy Discriminant Analysis (NFDA). To maintain bond between the neighborhoods in locality, NFDA is used and an efficient metaheuristic optimization algorithm called Glowworm Swarm Optimization clustering is used for clustering the gene expression based on the pattern. The experimental results shows that proposed algorithm achieve a higher clustering accuracy and takes lesser less clustering time when compared with existing algorithms.

Keywords: Glowworm Swarm Optimization, Neuro- fuzzy Discriminant Analysis, biclustering, gene expression data

I. INTRODUCTION

Modern technology provides efficient methods for data collection. The advent of DNA microarray technologies has revolutionized the experimental study of gene expression. Thousands of genes are routinely probed in a parallel fashion. The expression levels of their transcribed mRNA are reported. By repeating such experiments under different conditions (e.g, different patients, different tissues, or varying cells' environments) [1], data from tens to hundreds of experiments can be gathered. The analysis of these large datasets poses numerous algorithmic challenges. Clustering is the most popular approach of analyzing gene expression data and has proven successful in many applications, such as discovering gene pathway, gene classification, and function prediction. There is a very large body of literature on clustering in general and on applying clustering techniques to gene expression data in particular. Several representative algorithmic techniques have been developed and experimented in clustering gene expression data, which include but are not limited to hierarchical clustering [2], self-organizing maps [3], and graphic theoretic approaches (e.g., CLICK [4]).

The concept of bicluster is analogous to that of subspace clustering in data mining [5,6], though there exist important differences regarding the criteria adopted to measure the coherence among the rows and the type of biclusters found, that generally do not overlap, i.e. a row or a column can participate to only one bicluster.

A *bicluster* is a subset of genes that show similar activity patterns under a subset of conditions. The research on biclustering started in 1972 with Hartigan's work, in which the way of dividing a matrix in sub–matrices with the minimum variance was studied [7] (Hartigan *et al*., 1972). In that approach the perfect bicluster was the submatrix formed by constant values, i.e., with variance equal to zero. Hartigan's algorithm, named *direct clustering*, divides the data matrix into a certain number of biclusters, with the minimum variance value, so the fact of finding a number of sub-matrices equal to the number of elements of the matrix is avoided. Another way of searching biclusters is to measure the coherence between their genes and conditions. Cheng & Church [8](Cheng *et al*., 2000) introduced a measure, the mean squared residue (MSR), that computes the similarity among the expression values within the bicluster. The ideas of Cheng and Church were further developed by Yang [9,10] (Yang *et al*., 2002, 2003) who dealt with missing values in the matrices. As a result of this approach an algorithm named FLOC was designed. Other works [11] (e.g (Wang *et al*., 2002)) are based in a quality value as well, calculated using the expression values of biclusters, so to measure their coherence.

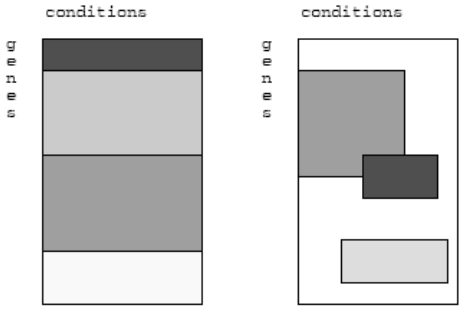

Fig. 1. Left: Traditional clustering searches a partition of all genes into *k* disjoint groups. Right: Biclustering searches for one or a set of blocks containing a consistent local pattern. Three biclusters are shown. (Note that it is not generally possible to display several biclusters at the same time as contiguous blocks.)

II. RELATED WORKS

Other alternatives in the searching for biclusters have been studied in the last years. We might also consider that a value in the data matrix is the sum of the contributions of different biclusters. Based on the previous idea, [12]Lazzeroni (Lazzeroni *et al*., 2000) presents the *plaid models*, in which the data matrix is described as a linear function of layers corresponding to its biclusters and shows how to estimate a model using an iterative maximization process. Shamir [13] (Shamir *et al*., 2002) proposes a new method to obtain biclusters based on a combination of graph theoretical and statistical modelling of data. In this model, a gene responds to a condition if its expression level changes significantly at that condition witch respect to its normal level. In a recent work [14](Liu *et al*., 2004), a generalization of OPSM model, introduced by [15] (Ben *et al*., 2002), is presented. The OPSM model is based on the search of biclusters in which a subset of genes induce a similar linear ordering along a subset of conditions. Some techniques search for specific structures in data matrix to find biclusters : [16] (Gerstein *et al*., 2003) creates a method for clustering genes and conditions simultaneously based on the search of "checkerboard" patterns in matrices of gene expression data. Previously the data is processed by normalization in a spectral framework model (several schemes all built around the idea of putting the genes on the same scale so that they have the same average level of expression across conditions, and the same for the conditions). Evolutionary computation techniques have also been used in this research area. These techniques use aspects from natural selection within computer science, including genetic algorithms, genetic programming and evolutionary strategies. In [17] (Aguilar *et al*., 2005) an evolutionary technique, based on the search of biclusters following a sequential covering strategy and measuring the mean squared residue, is used.

Another approach is pattern-based clustering, that captures the similarity of the patterns exhibited by a bicluster. In general, given a set of objects, a subset of these objects form a pattern-based cluster if these objects follow a similar pattern in a subset of dimensions. Wang et al. [11] proposed a depth-first algorithm for detecting patternbased clusters. In order to speed up the process and to avoid the repetition of computations, the algorithm uses a suffix tree to efficiently enumerate the possible combinations of row and column sets that represent a bicluster. Liu and Wang [18] also proposed an exhaustive bicluster enumeration algorithm, which is based on a model that generalizes the order preserving submatrix model [10]. The objective of finding all biclusters that, after column reordering, represent coherent evolutions of the symbols in the matrix is achieved by using a pattern discovery algorithm inspired in sequential pattern mining algorithms [19]. Another algorithm that uses the pattern-based clustering model is proposed in [20]. This algorithm mines only the maximal pattern-based clusters. It conducts a depthfirst, divide and conquer search and prunes unnecessary branches smartly.

Most of these previous techniques search for one or two types of biclusters among four that have been identified in the literature [21]: biclusters with constant values, biclusters with constant values on rows or columns, biclusters with coherent values, and biclusters with coherent evolution. Constant biclusters in a gene expression matrix identify subsets of genes with equal expression values within a subset of conditions. Biclusters with constant values along rows indicate a subset of genes with expression levels that do not change across a subset of conditions, irrespective of the actual expression levels of the individual genes. Biclusters with constant columns isolate a subset of conditions for which a subset of genes have constant expression values that may differ from condition to condition. Much of the prior work, however, has focused on finding more complex relations between genes and conditions by looking for biclusters with coherent values or evolutions. Such biclusters allow for variation in the actual numerical values of the gene expression levels. Instead, they focus on the behavior of the gene expression levels across subsets of genes or conditions. Gene expression levels in biclusters of coherent values obey additive or multiplicative models on rows or columns. Our focus here, and indeed that of most researchers, is on finding subsets of genes that are upregulated or downregulated across a subset of conditions irrespective of their actual expression values or subsets of conditions that have always the same or opposite effects on a subset of genes. As mentioned previously, and as we shall see later, finding such biclusters provides a starting point for elucidating genetic pathways. Most previous techniques are also greedy and will miss some biclusters that satisfy their definition of a valid bicluster. Many of these pioneering approaches used a cost function to define biclusters. In many cases, the cost function will measure the square deviation from the sum of the mean value of

expression levels in the entire bicluster and the mean values of expression levels along each row and column in the bicluster [22].

The gene microarray data are arranged based on the pattern of gene expression using various clustering algorithms and the dynamic natures of biological processes are generally unnoticed by the traditional clustering algorithms. To overcome the problems in gene expression analysis, novel algorithms for finding the coregulated clusters, dimensionality reduction and clustering have been proposed. The coregulated clusters are determined using biclustering algorithm, so it is called as coregulated biclusters. The coregulated biclusters are two or more genes which contain similarity features. The dimensionality reduction of microarray gene expression data is carried out using Locality Sensitive Discriminant Analysis (LSDA). To maintain bond between the neighborhoods in locality, LSDA is used and an efficient metaheuristic optimization algorithm called Artificial Bee Colony (ABC) using Fuzzy c Means clustering is used for clustering the gene expression based on the pattern. The experimental results shows that proposed algorithm achieve a higher clustering accuracy and takes lesser less clustering time when compared with existing algorithms.

III. METHODOLOGY

The proposed approach consists of three stages namely finding of coregulated biclusters using Bimax algorithm, dimensionality reduction using Locality Sensitive Discriminant Analysis (LSDA) and clustering using MoABC.

Finding of Coregulated Clusters using Bimax Algorithm:

The diagram for finding the coregulated clusters using algorithm is shown in Fig.1. The input is gene sequence f the micro array data. Enhanced Bimax algorithm is used to display a maximal biclusters value and displays a coregulated biclusters. The Enhanced Bimax algorithm is used to measure a particular gene is present or not. It also finds the transcription sites of the coregulated biclusters. Normalization technique used to specify genes are presented in the particular group or not. The output is display the transcription factors.

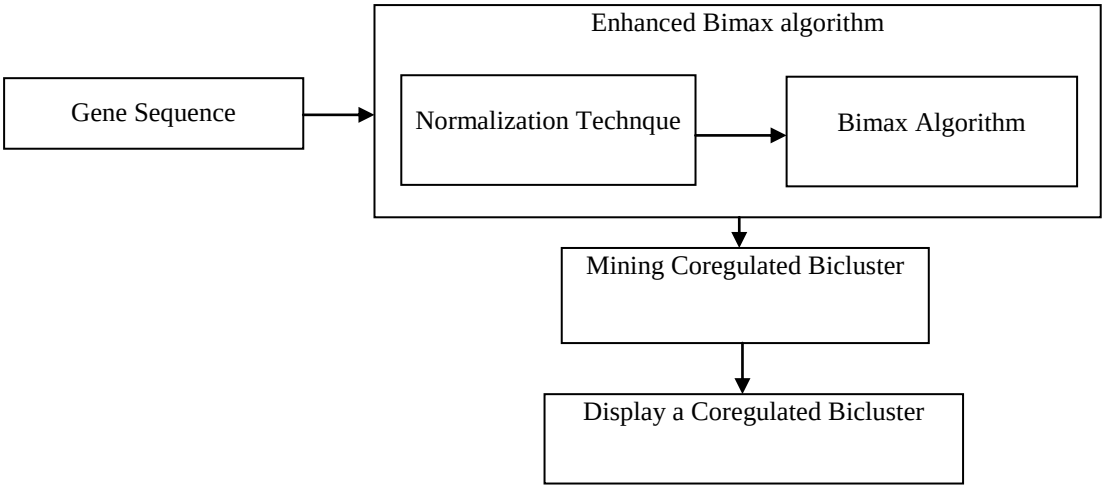

Figure 1: Block diagram for mining coregulated bicluster

Bimax Algorithm

The Bimax algorithm needs to guarantee that only optimal, inclusion-maximal biclusters are generated. The problem arises because V contains parts of the biclusters found in U, and as a consequence we need to ensure that the algorithm only considers those biclusters in V that extend over CV . The parameter Z serves this goal. It contains sets of columns that restrict the number of admissible biclusters. It is used to specify the genes and conditions. It is used to specify that analysis of DNA chips and gene networks. The algorithm realizes the divide-and-conquer strategy. Fig. 1 describes an original Bimax algorithm. It consists of three procedures. They are Enhanced Bimax. Conquer and Divide. Conquer function is call and check the condition is if the genes and conditions are equal then the partitioning is begin, otherwise it stop the process. Second step is split the data and normalization technique is used to group the splited data. It is used to find all add the maximum groups in general gene expression data. Each coregulated genes are grouping together the particular expression value and the particular situation [23].

Proposed Enhanced Bimax Algorithm

Binary Space Partitioning (BSP) is a method for recursively subdividing a space into convex sets by hyperplanes. This subdivision gives rise to a representation of the scene by means of a tree data structure known as a BSP tree. Normalization is the process of isolating statistical error in repeated measured data. Quintile normalization for instance, is normalization based on the magnitude of the measures. The goals in doing eliminate all the redundant data and ensure data dependencies. The numbers of genes that reproducibly showed and the unnormalized data and normalized data are displayed on the coregulated biclusters. Enhanced Bimax algorithm is applied data mining technique on clustering. In the clustering similar samples and similar gene probes are organized in a fashion so that they would lie close together. It consists of three procedures. They are Enhanced Bimax, Breadth-First Search (BFS) and BSP [23].

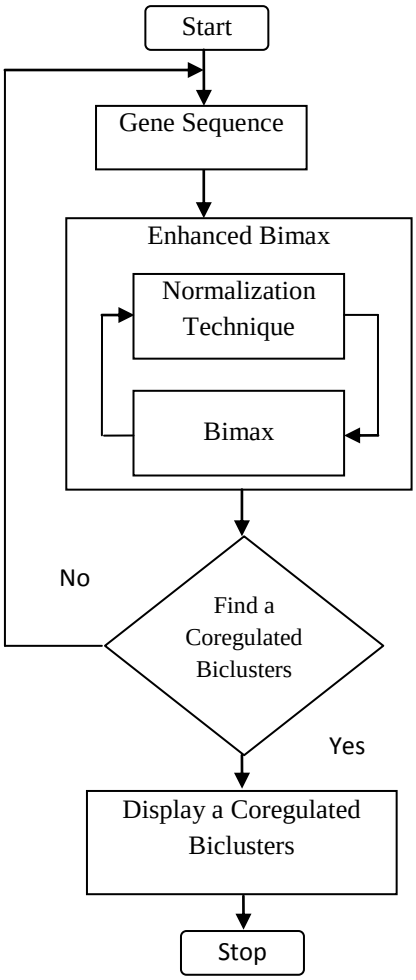

Figure 2: Flow Chart for Proposed Enhanced Bimax Algorithm

They are BFS and BPS combination of sequences searches the entire graph c nodes of a graph or ords, it exhaustively y Space Partitioning. First step is normalization technique used to remove the redundant data and then grouping genes in the specific conditions. Binary Space Partitioning function is call and check the condition is if the genes and conditions are equal then the partitioning is begin. Otherwise it stop the process. It specifies that a particular gene is present in the given group then it is represents a one. With these maximum groups in general gene expression data can be found. Each co regulated genes are grouping together the particular expression value and the particular situation. Otherwise the gene is not present in the given group then it is representing as zero. Fig. 2 describes a proposed Enhanced Bimax algorithm.

IV. NEURO-FUZZY DISCRIMINANT ANALYSIS

An artificial neural network (ANN) model is an information processing paradigm inspired by the way the biological nervous system process information. It consists of many nonlinear computational elements operating in parallel and arranged in patterns reminiscent of biological neural networks. These computational elements, known as the nodes or the neurons, are connected via weights that are typically adapted during the use to improve the performance. The ANNs have long been utilized in a great variety of tasks. However, at present, their main practical applications have been for classification tasks. Earlier studies on the relation between discriminant analysis and the Multilayer Perceptrons (MLP) used for classifications date back long time ago. Several studies were made to illustrate why nonlinear adaptive feed-forward layered networks with linear output units can perform well for pattern classification [24], [25]. These studies proved that within MLP, each layer of weights can be thought of as performing projections that try to separate as best as possible the different classes, so they can be linearly separable by the cells in the last layer. All of these studies suggest that the MLP actually consist of two projections: A Non-linear projection from input-to-hidden and from each hidden-to-hidden layer and a second projection being linear from the final hidden-to-output layer.

Several studies followed, but the main trend was decomposed into two parts. The first focused on enhancing the functionality of multilayer feed-forward neural networks performing the nonlinear discriminant analysis [26], [27]. The second trend, as mentioned earlier, focused on Fisher's discriminant analysis itself as a statistical technique and mixing this technique with kernel function to perform the nonlinear mapping [28], [29], [30]. Although many of these studies does actually perform well as a nonlinear discriminant analysis tool, but up to the authors' knowledge there were no studies that combined neural networks with the statistical discriminant analysis to form a dimensionality reduction tool. Thus the main focus of this paper is to combine these two techniques and compare the performance of the proposed nonlinear method with other techniques.

Differential Evolution based Weights Optimization

Differential evolution is a simple optimization technique having parallel, direct search, easy to use, good convergence, and fast implementation properties [31]. The crucial idea behind DE is a new scheme for generating trial parameter vectors by adding the weighted difference vector between two population members $\boldsymbol{x_{r1}}$and $\boldsymbol{x_{r2}}$ to a third member $\boldsymbol{x_{r0}}$ The following equation shows how to combine three different, randomly chosen vectors to create a mutant vector, $\boldsymbol{v}_{i,g}$ from the current generation g:

$$\boldsymbol{v}_{i,g} = \boldsymbol{x_{r0.g}} + \boldsymbol{F} \times \left(\boldsymbol{x_{r1.g}} - \boldsymbol{x_{r2.g}}\right)$$

where $F \in (0,1)$ is a scale factor that controls the rate at which the population evolves. The index g indicates the generation to which a vector belongs. In addition, each vector is assigned a population index, i, which runs from 0 to Np - 1. Parameters within vectors are indexed with j, which runs from $0\ to\ D\ -\ 1$.

Extracting both distance and direction information from the population to generate random deviations results in an adaptive scheme that has excellent convergence properties. In addition, DE also employs uniform crossover, also known as discrete recombination, in order to build trial vectors out of parameter values that have been copied from two different vectors. In particular, DE crosses each vector with a mutant vector, as given below:

$$u_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} & if\ rand(0,1) \le C_r\ or \\ x_{j,i,g} & otherwise \end{cases}$$

where uj,i,g is the i_th trial vector along j_th dimension from the current population g. The crossover probability Cr ∈ [0, 1] is a user defined value that controls the fraction of parameter values that are copied from the mutant. If the newly generated vector results in a lower objective function value (higher fitness) than the predetermined population member, then the resulting vector replaces the vector with which it was compared [32].

Each member of the population hold two pieces of information. The first is a possible representation for the weights attached to each connection in the network, and the second is variable z to be added to the diagonal value of the within class scatter matrix to prevent it from being singular. In simple words, the connection weight matrix is represented by a linear genome formed by concatenating each of its rows. A population of 100 members was initially randomly generated. In order to bound the search space, the weight values were limited to a range between -1 and +1. This constraint also helps reduce the chance that the evolutionary process will produce a forced model with extreme weight values. The evolution process starts after initialization according the to DE equations (A modified version of the one published by [7]). After computing the values of the connection weights for each node, the output of each node will be computed according to the following equation:

$$\mu_j(t) = f_t\left(\sum_{i=0}^{n-1} w_{ij}\, x_i - \theta_j\right)$$

In this equation, $\mu_j(t)$ is the output of node j at time $t, \boldsymbol{x_i}$ is the element i of the input, and f_t is the nonlinear transfer function chosen as the sigmoid function in this paper. θ_j is the threshold value associated with each neuron, that can also be included in the genome linear representation.

One of the points that should be taken into consideration with feed-forward neural networks employing a sigmoid function is that care should be taken so that the maximum input to the nonlinear transfer function will not cause the output to saturate. Another point that affects the performance of the network is the number of nodes or neurons in the hidden layer. It is important to use enough neurons to capture the nonlinearities in the input, however, using too many neurons may cause an over fitting. In such a case the proposed neural network will not be able to generalize well on unseen data [33].

Since the weights of the proposed neural network that are evolved using the DE optimization technique require a fitness function to evaluate the importance of each member of the population, then the classification accuracy achieved by a suitable classifier is used here as a fitness function.

A. New Fuzzy Linear Discriminant Analysis Projection Technique

Consider a classification problem with c classes, in which the data set of labelled training samples is given as:

$$S = \{(x_1, y_1), (x_2, y_2), \ldots., (x_l, y_l)\} \sqsubseteq (X, Y)^l$$

Where X is the input space and Y is the output space. $X \subseteq \Re^n$, and l is the number of samples. Each training point x_i, where $i = \{1, 2, 3, \ldots, c\}$, originally belongs to one of the c classes and is given a label $y_i \in \{1, 2, 3, \ldots, c\}$. The goal is to find an optimal hyper-plane using the training samples that can recognize the test points, i.e., the classifier will have a good generalization capability. In FLDA each point, x_i, belongs to each of the c classes with a certain membership. The fuzzy within class scatter matrix SW, fuzzy between class scatter matrix SB, and the fuzzy total class scatter matrix ST are given as follows [34]:

$$S_w = \sum_{i=1}^{c} \sum_{k=1}^{l_i} \mu_{ik}^m (x_k - v_i)\,(x_k - v_i)^T$$

$$S_B = \sum_{i=1}^{c} \sum_{k=1}^{l_i} \mu_{ik}^m (v_i - \bar{x})\,(v_i - \bar{x})^T$$

$$S_T = \sum_{i=1}^{c} \sum_{k=1}^{l} u_{ik}^m (x_k - \bar{x})\,(x_k - \bar{x})^T$$

where m ($given\ that\ m > 1$) is the fuzzification parameter, $u_i k$ is the membership of pattern k in class i, x_{kj} is the value of the $k'th$ sample across the $j'th$ dimension, v_i is the mean of the patterns belonging to class $i, and\ v_{ij}$ is its value across the $j'th$ dimension. x is the mean of the training samples.

$$\bar{x} = \frac{1}{l}\sum_{k=1}^{l} x_k$$

In this paper, the value of the membership uik is calculated using a possibilistic fuzzy clustering approach. The cost function of the possibilistic clustering approach is adopted from [14], as given in Eq.(9) below.

$$J(\theta, U) = \sum_{k=1}^{l}\sum_{i=1}^{c} u_{ik}^{m} d(x_k, \theta_i) + \sum_{i=1}^{c} \eta_i \sum_{k=1}^{l} (1 - u_{ik})^m$$

where θ_i is the i'th cluster center, η_i is a positive constant that is suitably chosen for each class. The first term in Eq. (9) is the same objective function used in the probabilistic clustering approach, while the second term is added to reduce the effect of outliers. In order to find the membership values from the above equation, then the values of the clusters centers are needed. A direct way would be to differentiate Eq. (9) with respect to θ_i, but this in turn would cancel the second term leaving only the first term. A general look at the first term of Eq. (9) reveals that it represents the classical within class scatter matrix SW given in Eq. (5) if the weight is removed. Thus applying the values of the clusters means ensures that the objective function given by Eq. (9) would settle at a global optimum value. Then in order to compute the membership values, a differentiation of the resultant function with respect to u_{ik} needs to be done as follows.

$$\frac{\partial J(\theta, U)}{\partial u_{ik}} = m u_{ik}^{m-1} d(x_k, v_i) - m\eta_i (1 - u_{ik})^{m-1} = 0$$

This would in turn result in the following function

$$u_{ik} = \frac{1}{1 + \left(\frac{d(x_k, v_i)}{\eta_i}\right)^{\frac{1}{m-1}}}$$

The values of $\eta_i, i = \{1, 2, 3, \ldots, c\}$ were chosen to be equal to the maximum distance between the samples belonging to that class and the class center.

After computing all the variables, FLDA finds the vector G that would maximize the ratio of the between class scatter matrix to the within class scatter matrix by solving the following equation:

$$G = \arg\max_{G} trace\left(\frac{G^T S_B G}{G^T S_w G}\right)$$

The solution can be readily computed by applying an Eigendecomposition on $S_W^{-1} S_B$, provided that the within class scatter matrix SW is nonsingular. In this paper, we are using a regularized version of SW given by $= SW + zI$, for some $z > 0$ that is included in the weight representation in Figure. 2, where I is an identity matrix. In this way the scatter matrix is guaranteed to be nonsingular. Since the rank of the between class scatter matrix is bounded from above by $c - 1$, there are at most $c - 1$ discriminant vectors by FLDA.

B. Proposed Glowworm swarm optimization clustering

It is necessary to introduce GSO with optimizing multi-modal functions, as it is helpful to understand the working principle of the algorithm. When using GSO optimizing multi-modal functions, physical agents i ($i = 1, \ldots, n$) are initially randomly deployed in the objective function space. Each agent in the swarm decides its direction of movement by the strength of the signal picked up from its neighbours. This is similar to the luciferin induced glow of a glowworm which is used to attract mates or prey in nature. The brighter the glow, the more is the attraction. Therefore, the authors use the glowworm metaphor to represent the underlying principles of the GSO algorithm.

GSO algorithm to optimize the multi-modal function include 5 major steps:

1) According to the formula (2), each glowworm *i* encodes the objective function value $J(x_i(t))$ at its current location $x_i(t)$ into a luciferin value $l_i(t)$;
2) Constructing neighborhood set $N_i(t)$;
3) According to the formula (3),each glowworm i calculate moves toward j probability $p_{ij}(t)$;
4) Select the moving objects j^* , using the formula (4) calculate the new location $x_i(t\ 1)s$,is the moving step;
5) According to the formula (5) update the radius of the dynamic decision domain.

$$l_i(t) = (1 - \rho) l_i(t - 1) + \gamma J\big(X_i(t)\big)$$

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)}$$

$$X_i(t+1) = X_i(t) + s^*\left(\frac{X_j(t) - X_i(t)}{\|X_j(t) - X_i(t)\|}\right)$$

$$r_d^i(t+1) = \min\left\{r_s, max\{0, r_d^i(t) + \beta(n_t - |N_i(t)|)\}\right\}$$

A full parameters analysis is found in [35]Krishnanand and Ghose (2008b), show that the choice of these parameters has some influence on the performance of the algorithm. In terms of the total number of peaks captured, they suggest the parameter selection as shown in Table 1. Thus, only *n* and *s r* need to be selected. These parameters value brings more convenience to people to apply the GSO algorithm.

Table 1. The GSO algorithm parameter selection

ρ	γ	β	n_t	s	l_0
0.4	0.6	0.08	5	0.03	5

C. GSO clustering

The previous section we have brief introduced the GSO algorithm, the GSO clustering algorithm will be proposed in this section, abbreviation GSOCA.

One of the most important components of a clustering algorithm is the measure of similarity used to determine how close two patterns are to one another. In this paper, we use the local space relative density to reflect the local data similarity

After above definitions, the GSO clustering process can be described as follows: each glowworm *i* represents a cluster data object, according to the formulate (5) calculate it's local space relative density , use the formulate (6) calculate its attraction, then encodes its current attraction value into a luciferin value by the formulate (1), and broadcasts the same within its neighbourhood; Within its dynamic decision domain range, select have a relatively higher luciferin value agent to constitute its neighbours; According to the formula (2), glowworm *i* calculate moves toward neighbour *j* probability *p* (*t*) *ij* ; According to probability select move objects moves toward it, update its location by the formulate (3); According to the formula (4) update the radius of the dynamic decision domain. Through repeatedly carries out the above process, and finally realize the data objects self-organization cluster. Algorithm symbolic description: $X_i(t)$ is the glowworm i in t iteration location; $l_i(t)$ is the luciferin of the glowworm i in t iteration; $N_i(t)$ is the neighbourhood set of glowworm i in t iteration; $r_i^d\ (t)$ is the dynamic decision domain radius of glowworm i in t iteration; r_s is the upper bound of the $r_i^d\ (t)$; $p_{ij}(t)$ is the probability of glowworm i selects neighbour j ; n_t is the threshold of the number of agents include in the neighbourhood set; ρ is the evaporation rate of luciferin ; γ is the replacement rate of luciferin; β is the rate of change of the neighbourhood range.

V. RESULTS AND DISCUSSION

The proposed technique for microarray gene clustering has been implemented in the working platform of MATLAB (version 7.11). For evaluating the proposed technique, the microarray gene samples of human acute leukemia and colon cancer data are utilized. [36] The high dimensional gene expression data has been subjected to dimensionality reduction and so a dimensionality reduced gene data with dimensions has been obtained. Thus LSDA is applied to identify informative genes and reduce gene dimensionality for clustering samples to detect their phenotypes.

TABLE I: MICROARRAY GENE DATA DIMENSION UTILIZED FOR THE EVALUATION PROCESS

Types of Gene Data	Number of Samples	Number of Genes	Dimensionality Reduced Data with the aid of LPP
ALL	41	7139	41X41
AML	36	7128	36X40
COLON	68	3000	62X42

Source: Jacinth Salome and Suresh [37]

A sample of microarray gene dataset of three classes that has been used for testing is given in the Table II. Clustering for microarray gene expression data whose amount is large can be fully calculated by determining the boundary of the clusters.

TABLE II: A SAMPLE OF THE MICROARRAY GENE DATA TO TEST THE PROPOSED TECHNIQUE

Class	ALL		AML		COLON	
Sample gene	ALL 16125 TA-Norel	ALL 23668 TA-Norel	AML SH-5	AML SH-13	AFFX-MurIL2	AFFX-MurIL10
AFFX-CreX-5_at(endogenous control)	- 172A	-93A	-271A	-11A	20.6	-16
AFFX-CreX-3_at(endogenous control)	52A	10A	-12A	112A	-8.7	41.2
AFFX-BioB-5_st(endogenous control)	-134A	159A	-104A	-176A	4880	26.2

Source: Jacinth Salome and Suresh [37]

While testing, when a gene dataset is given, the proposed technique has to identify its belonging cluster. Existing clustering algorithms, such as Fuzzy C-means and Fuzzy Possibilistic C-Means Algorithm using EM Algorithm approaches and also MoABC are applied both to group genes, to partition samples in the early stage and have proven to be useful. The performance of each clustering algorithm may vary greatly with different data sets. Complete-link clustering method uses the smallest similarity within a cluster as the cluster similarity, and every data object within the cluster is related to every other with at least the similarity of the cluster. In order to test the performance of the data, N artificial m-dimensional feature vectors from a multivariate normal distribution having different parameters and densities were generated. Situations of large variability of cluster shapes, densities, and number of data points in each cluster were simulated.

TABLE III: PERFORMANCE COMPARISON IN PERCENTAGE BETWEEN THE PROPOSED MOABC CLUSTERING TECHNIQUE AND OTHER EXISTING TECHNIQUES

Type of Gene Data	Accuracy				Correlation				Distance				Error rate			
	FPCM	EMFPCM	MoABC	GSO and NFDA	FPCM	EMFPCM	MoABC	GSO and NFDA	FPCM	EMFPCM	MoABC	GSO and NFDA	FPCM	EMFPCM	MoABC	GSO and NFDA
ALL	83.9	85.69	87.25	90.54	0.368	0.412	0.4852	0.4962	0.00346	0.00263	0.00142	0.00098	0.20	0.18	0.12	0.09
AML	81.02	83.84	85.12	88.63	0.029	0.0315	0.0396	0.0462	0.00331	0.00201	0.00185	0.00101	0.29	0.24	0.16	0.11
COLON	79.9	81.96	83.04	87.98	0.125	0.139	0.215	0.368	0.02011	0.0126	0.0099	0.0048	0.03	0.01	0.006	0.002

Source: Jacinth Salome and Suresh [37]

From the Table III, it can be seen that the proposed technique MoABC has provided more accuracy, correlation and less distance and error rate rather than the other gene clustering techniques like FCM, FPCM etc. More accuracy and less error rate leads to effective clustering of the given microarray gene data to the actual class of the gene.

VI. CONCLUSION

This paper has introduced a general global search framework for biclustering of gene expression data. As global optimizer, an evolutionary algorithm was used that can be an effective microarray gene data clustering technique has been proposed with the aid of Bimax algorithm, NFDA and GSO. Initially, the micro array data genes are given to the Bimax algorithm to find coregulated biclusters to reduce the space complexity of the genes, and then the dimensionality of the microarray data has been reduced with the help of NFDA mechanism. The technique has been tested by clustering the microarray gene expression data of human acute leukemia and colon cancer data. The approaches are compared based on their performances and it can be noticed that the proposed approach yields equally good results for the entire functional category. The comparative results have shown that the proposed technique possesses better accuracy, correlation and lesser distance, error rate than FCM, FPCM gene clustering techniques. Hence, the proposed GSO approach for gene clustering has paved the way for effective information retrieval in the microarray gene expression data.

REFERENCES

[1] Jinze Liu1, Jiong Yang2, and Wei Wang1 "Biclustering in Gene Expression Data by Tendency" Computational Systems Bioinformatics Conference, 2004. IEEE

[2] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein.Cluster analysis and display of genome-wide expression patterns. In Proc Natl Acad Sci U S A,95(25):14863-8, 1998.

[3] S. Kaski, J. Nikkil, and G. Wong. Analysis And Visualization Of Gene Expression Data Using Self- Organizing Maps, Proceedings of NSIP-01, IEEEEURASIP Workshop on Nonlinear Signal and Image Processing, 2001.

[4] R. Sharan and R. Shamir. Click: A clustering algorithm with applications to gene expression analysis. In ISMB, pages 307-216, 2000.

[5] R. Agrawal, J.C. Gehrke, D. Gunopulos, P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, in: Proceedings of the ACM International Conference on Management of Data (SIGMOD'98), 1998, pp. 94–105.

[6] C.C. Aggarwal, P.S. Yu, Finding generalized projected clusters in high dimensional spaces, in: Proceedings of the ACM International Conference on Management of Data (SIGMOD'00), 2000, pp. 70–81.

[7] Hartigan,J.A. (1972) Direct clustering of a data matrix, Journal of the American Statistical Association, 67(337), 123-129.

[8] Cheng,Y., Church,G.M. (2000) Biclustering of expression data, Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology, ISMB'00, 93-103.

[9] Yang,J. ,Wang,W., Haixun,W., Yu,P. (2002) Improving Performance of Bicluster Discovery in a Large Data Set, 6th ACM International Conference on Research in Computational Molecular Biology, RECOMB2002, Poster.

[10] Yang,J. ,Wang,W., Haixun,W., Yu,P. (2003) Enhanced biclustering on expression data, 3rd IEEE Conference on Bioinformatics and Bioengineering, 321-327.

[11] Wang,H. ,Wang,W., Haixun,W., Yu,P. (2002) Clustering by pattern similarity in large data sets, ACM SIGMOD International Conference on Management of Data, 394-405.
[12] Lazzeroni,L., Owen, A. (2000) Plaid models for gene expression data, Technical Report Stanford University.
[13] Shamir,R., Sharan,R., Tanay,A. (2002) Discovering statistically significant biclusters in gene expression data, Bioinformatics, vol. 19, Suppl. 1 2002, 136-144.
[14] Liu,J., Yang, J., Wang,W. (2004) Biclustering in Gene Expression Data by Tendency, *IEEE Computational Systems Bioinformatics Conference 2004*, 183-193.
[15] Ben-Dor,A., Chor,B., Karp,R., Yakhini,Z. (2002) Discovering local structure in gene expression data : The Order Preserving Submatrix Problem, 6th ACM International Conference on Research in Computational Molecular Biology, RECOMB2002.
[16] Gerstein,M., Chang,J., Basri,R. ,Kluger,Y. (2003) Spectral Biclustering of Microarray Data : Coclustering Genes and Conditions, *Journal Genome Research*, vol. 13(4), 703-716.
[17] Aguilar,J.S., Divina,F. (2005) Evolutionary Biclustering of Microarray Data, *3rd European Workshop on Evolutionary Bioinformatics*.
[18] J. Liu and W. Wang, "Op-Cluster: Clustering by Tendency in High Dimensional Space," Proc. Third IEEE Int'l Conf. Data Mining,p. 187-194, 2003.
[19] J. Hipp, U. Gu¨ ntzer, and G. Nakhaeizadeh, "Algorithms for Association Rule Mining—A General Survey and Comparison," SIGKDD Explorations Newsletter, vol. 2, no. 1, pp. 58-64, 2000.
[20] J. Pei, X. Zhang, M. Cho, H. Wang, and P.S. Yu, "Maple: A Fast Algorithm for Maximal Pattern-Based Clustering," Proc. Third IEEE Int'l Conf. Data Mining, p. 259-266, 2003.
[21] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: A survey," *IEEE Trans. Comput. Biol. Bioinform.*, vol. 1, no. 1, pp. 24–45, Jan.–Mar. 2004.
[22] Ahmed H. Tewfik, Alain B. Tchagang, Laura Vertatschitsch "Parallel Identification of Gene Biclusters With Coherent Evolutions" IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 54, NO. 6, JUNE 2006.
[23] C.P. Chandran and K. IswaryaLakshmi, "biclustering analysis of coregulated biclusters from gene expression data", International Journal of Computational Intelligence and Informatics, Vol.2, No.1, 2012.
[24] A.R. Webb and D. Lowe, *The optimised internal representation of multilayer classifier networks performs nonlinear discriminant analysis*, Neural Networks, Vol. 3, No. 4,pp. 367-375, 1990.
[25] P. Gallinari, S. Thiria, F. Badran, and F. Fogelman-Foulie, *On the relations between discriminant analysis multilayer perceptrons*, Neural Networks, Vol. 4, No. 3, pp. 349–360, 1991.
[26] D. Casasent, and X. Chen, Radial basis function neural networks for nonlinear fisher discrimination and neyman-pearson classification, Neural Networks, Vol. 16, Volume 16 , No. 5-6, pp. 529-535, 2003.
[27] Y. Kwon, and B. Moon, Nonlinear feature extraction using a neuro genetic hybird, Proceedings of the 2005 conference on Genetic and evolutionary computation', Washington DC, USA, pp. 2089–2096, 2005.
[28] J. W. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, *Face recognition using kernel direct discriminant analysis algorithms*, IEEE Transactions on Neural Networks, Vol. 14, No. 1, pp. 117–126, 2003.
[29] T. Xiong, J. Ye, and V. Cherkassky, *Kernel uncorrelated and orthogonal discriminant analysis: A unified approach*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 125-131, 2006.
[30] Z. Liang and P. Shi, *Uncorrelated discriminant vectors using kernel method*, Pattern Recognition, Vol. 38, No. 2, pp. 307–310, 2005.
[31] *K. V. Price, R. M.Storn, and J. A. Lampinen, Differential evolution: A practical approach to global optimization, Springer, 2005.*
[32] *K. Palit and D. Popovic, "Computational intelligence in time series forecasting: theory and engineering applications", Springer, 2005.*
[33] *L. H. Tsoukala, and R. E. Uhrig, Fuzzy and Neural Approaches in Engineering (Adaptive and Learning Systems for Signal Processing, Communications and Control Series), John Wiley and Sons, 1997.*
[34] *H. X. Wua and J. J. Zhoua, Fuzzy discriminant analysis with kernel methods, Pattern Recognition, Vol. 39, No. 11, pp.22362239, 2006.*
[35] *K.N.Krishnand, D.Ghose. "Glowworm swarm optimisation: a new method for optimising multimodal functions". Int. J. Computational Intellingence Studies, vol.1,no.1,pp.93~119. 2009*
[36] Jian Wen, "Ontology Based Clustering for Improving Genomic IR", Twentieth IEEE International Symposium International Journal of Data Mining and Bioinformatics,Vol. 3, No. 3, pp.229-259, 2009.
[37] Jacinth Salome J and R M Suresh, "Efficient Clustering for Gene Expression Data", International Journal of Computer Applications (0975 – 888), Volume 47– No.5, June 2012.

