Multiobjective Fuzzy Biclustering in Microarray Data: Method and a New Performance Measure

Ujjwal Maulik, Senior Member, IEEE, Anirban Mukhopadhyay, Sanghamitra Bandyopadhyay, Senior Member, IEEE, Michael Q. Zhang and Xuegong Zhang

Abstract—Objective of any biclustering algorithm in microarray data is to discover a subset of genes that are expressed similarly in a subset of conditions. The boundaries of biclusters usually overlap as genes and conditions may belong to different biclusters with different membership degrees. Hence the notion of fuzzy sets is useful for discovering such overlapping biclusters. In this article an attempt has been made to develop a multiobjective genetic algorithm based approach for probabilistic fuzzy biclustering that minimizes the residual and maximizes cluster size and expression profile variance. A novel variable string length encoding has been proposed in this regard that encodes multiple biclusters in a single string. Also a new performance measure that reflects how a bicluster is statistically distinguished from the background is proposed. Performance of the proposed algorithm has been compared with some well known biclustering algorithms.

Index Terms—Fuzzy biclustering, fuzzy K-medoids, mean squared residue, expression profile variance, multiobjective genetic algorithm, variable string length, statistical difference from background.

I. INTRODUCTION

GENOMIC data generated by the microarray technology is generally huge. Therefore computational analysis is required to extract the knowledge hidden in a microarray database. Clustering [1, 2] is an exploratory unsupervised pattern classification technique that partitions the input space in a number of groups depending on some similarity/dissimilarity matrix. Standard clustering methods, such as K-means [1], hierarchical methods [2], Self Organizing Maps (SOM) [3], and genetic algorithm (GA) based clustering methods [4], [5], [6], discover the cluster of genes over all conditions. Thereby, they may fail to discover the genes that have similar expression pattern over a subset of conditions. On the other hand, biclustering algorithms aim to find a subset of genes that have similar expression profile over a subset of experimental conditions, and thus, they provide better reflection of the biological reality [7], [8].

Microarray gene expression data usually contains noise. It is also highly likely that a gene or condition may belong to different biclusters simultaneously with different degree of belongingness. Therefore a lot of uncertainty is involved in microarray data sets. Hence the concept of fuzzy set theory is useful to discover such overlapping biclusters from noisy background. This article makes such an attempt which is motivated by fuzzy K-medoids clustering [9].

In many real world situations there may be several objectives that must be optimized simultaneously in order to solve a certain problem. This is in contrast to the problems tackled by conventional GAs which involve optimization of just a single criterion. Unlike single objective technique, in multiobjective optimizations (MOO) [5], [10], [11], [12], [13], search is performed over a number of, often conflicting, objective functions. The final solution set contains a number of Pareto-optimal solutions, none of which can be further improved on any one objective without degrading it in another. In this article, a multiobjective GA-based probabilistic fuzzy biclustering algorithm has been presented that proposes a novel variable string length encoding scheme based on indices of the genes and the conditions, where each string encodes a number of possible biclusters in the data set. The proposed algorithm clusters both the dimensions simultaneously using fuzzy K-medoids clustering. NSGA-II [14] multiobjective algorithm is used as the underlying optimization strategy. Three objective functions, namely, mean squared residue and cluster size or the volume and the expression profile variance of the biclusters are optimized simultaneously.

In recent years, several studies have been made by the researchers in the context of biclustering of microarray data. Biclustering was first introduced in [15] in the name of direct clustering. The goal was to discover a set of submatrices with zero variance, i.e., with constant values. One of the prior works on biclustering can be found in [7], where the measure to compute the coherence among a group of genes, i.e., the mean squared residue, has been introduced. The algorithm developed in [7] was based on a greedy search technique guided by a heuristic. Some other recent approaches to biclustering are Flexible Overlapped Biclustering (FLOC) [16] and Random Walk biclustering (RWB) [17]. In [18], a coupled two-way clustering (CTWC) method has been proposed that uses hierarchical clustering in both dimensions and combines them to obtain the biclusters. A bipartite graph based model called Statistical-algorithmic Method for Bicluster Analysis (SAMBA) has been proposed.
for biclustering in [19].

The performance of the proposed algorithm has been demonstrated on two benchmark real life gene expression data sets, viz., Yeast cell cycle and Human large B-cell lymphoma. A new performance measure, called Statistical Difference from Background (SDB) has been proposed to evaluate biclusters to measure how much they are statistically different from the background data matrix. The performance of the proposed fuzzy biclustering technique is also compared with that of Cheng and Church’s algorithm (CC) [7] and RWB algorithm [17]. Also biological significance test based on Gene Ontology (GO) has been carried out for an example bicluster to show the effectiveness of the proposed approach.

II. MULTIOBJECTIVE OPTIMIZATION

The main difficulty in considering multiobjective optimization is that there is no accepted definition of optimum in this case, and therefore it is difficult to compare one solution with another one. In general, these problems admit multiple solutions, each of which is considered acceptable and equivalent when the relative importance of the objectives is unknown. The best solution is subjective and depends on the need of the designer or decision maker. The multiobjective optimization can be formally stated as [10], [20], [11], [21]:

Find the vector \( \mathbf{x}^* = [x_1, x_2, \ldots, x_n]^T \) of decision variables which will satisfy the \( m \) inequality constraints:

\[
g_i(\mathbf{x}) \geq 0, \quad i = 1, 2, \ldots, m,
\]

the \( p \) equality constraints

\[
h_i(\mathbf{x}) = 0, \quad i = 1, 2, \ldots, p,
\]

and optimizes the vector function

\[
\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})]^T.
\]

The constraints given in Eqs. 1 and 2 define the feasible region \( \mathcal{F} \) which contains all the admissible solutions. Any solution outside this region is inadmissible since it violates one or more constraints. The vector \( \mathbf{x}^* \) denotes an optimal solution in \( \mathcal{F} \). In the context of multiobjective optimization, the difficulty lies in the definition of optimality, since it is only rare that we will find a situation where a single vector \( \mathbf{x}^* \) represents the optimum solution for all the objective functions.

The concept of Pareto optimality is useful in the domain of multiobjective optimization. A formal definition of Pareto optimality from the viewpoint of minimization problem may be given as follows: A decision vector \( \mathbf{x}^* \) is called Pareto optimal if and only if there is no \( \mathbf{x} \) that dominates \( \mathbf{x}^* \), i.e., there is no \( \mathbf{x} \) such that

\[
\forall i \in \{1, 2, \ldots, k\}, f_i(\mathbf{x}) \leq f_i(\mathbf{x})^* \quad \text{and} \quad \exists i \in \{1, 2, \ldots, k\}, f_i(\mathbf{x}) < f_i(\mathbf{x})^*.
\]

In other words, \( \mathbf{x}^* \) is Pareto optimal if there exists no feasible vector \( \mathbf{x} \) which causes a reduction on some criterion without a simultaneous increase in at least one other. In this context, two other notions, weakly non-dominated and strongly non-dominated solutions are defined [11]. A point \( \mathbf{x}^* \) is a weakly non-dominated solution if there exists no \( \mathbf{x} \) such that \( f_i(\mathbf{x}) < f_i(\mathbf{x}^*) \), for \( i = 1, 2, \ldots, k \). A point \( \mathbf{x}^* \) is a strongly non-dominated solution if there exists no \( \mathbf{x} \) such that \( f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*) \), for \( i = 1, 2, \ldots, k \), and for at least one \( i \), \( f_i(\mathbf{x}) < f_i(\mathbf{x}^*) \). In general, Pareto optimum usually admits a set of solutions called non-dominated solutions.

There are different approaches for solving multiobjective optimization problems [10], [11], e.g., aggregating, population based non-Pareto and Pareto-based techniques. In aggregating techniques, the different objectives are generally combined into one using weighting or goal based method. Vector evaluated genetic algorithm (VEGA) is a technique in the population based non-Pareto approach in which different subpopulations are used for the different objectives. Multiple objective GA (MOGA), non-dominated sorting GA (NSGA), niched Pareto GA (NPGA) constitute a number of techniques under the Pareto based non-elitist approaches [10]. NSGA-II [14], SPEA [22] and SPEA2 [21] are some popular multi-objective elitist techniques. Recently, a simulated annealing based multiobjective optimization technique called AMOSA has been proposed in [13]. The present article uses NSGA-II as underlying multiobjective framework for developing the proposed biclustering algorithm.

III. BICLUSTERING MODEL

A microarray data set can be considered as a \( G \times C \) matrix \( M \) that represents the expression level of a set of \( G \) genes \( G = \{I_1, I_2, \ldots, I_g\} \) over a set of \( C \) conditions \( C = \{J_1, J_2, \ldots, J_c\} \). Each element \( m_{ij} \) of matrix \( M \) represents the expression level of the \( i \)th gene at the \( J \)th condition, where \( i \in G \) and \( j \in C \).

A bicluster is a submatrix \( B = (I, J) \) of matrix \( M \), where \( I \subseteq G \) and \( J \subseteq C \). The volume \( \text{vol}(I, J) \) of a bicluster \( B = (I, J) \) is the total number of elements in the bicluster, i.e.,

\[
\text{vol}(I, J) = |I| \times |J|.
\]

If \( B = (I, J) \) is a bicluster then the residue \( r_{ij} \) of any element \( a_{ij} \) of the bicluster is defined as:

\[
r_{ij} = a_{ij} - a_{iJ} - a_{Ij} + a_{IJ},
\]

where \( a_{ij} = \frac{1}{|I|} \sum_{j \in J} a_{ij} \) (mean of the \( i \)th row), \( a_{iJ} = \frac{1}{|C|} \sum_{i \in I} a_{ij} \) (mean of the \( j \)th column), and \( a_{IJ} = \frac{1}{|I| \times |J|} \sum_{i \in I, j \in J} a_{ij} \) (mean of all the elements in the bicluster).

The mean squared residue (MSR) of a bicluster \( B = (I, J) \) is defined as:

\[
\text{MSR}(I, J) = \frac{1}{\text{vol}(I, J)} \sum_{i \in I, j \in J} r_{ij}^2.
\]

The mean squared residue score of a bicluster represents the level of coherence among the elements of the bicluster. Lower residue score means higher coherence and thus better quality of the bicluster.
IV. FUZZY K-MEDOIDS CLUSTERING

The fuzzy k-medoids [9] algorithm is the extension of well-known fuzzy C-means [23] algorithm replacing cluster means with cluster medoids. A medoid is defined as follows:

Let $Y = \{y_1, y_2, \ldots, y_v\}$ be a set of $v$ objects. The medoid of $Y$ is an object $O \in Y$ such that the sum of distances from $O$ to other objects in $Y$ is minimum, i.e., the following criterion is minimized.

$$D(O, Y) = \sum_{i=1}^{v} D(O, y_i).$$

Here $D(O, y_i)$ denotes the distance measure between $O$ and $y_i$.

The aim of fuzzy k-medoids algorithm is to cluster the data set $X = \{x_1, x_2, \ldots, x_n\}$ into $K$ partitions so that the following criterion is minimized.

$$J_m(U, Z : X) = \sum_{k=1}^{n} \sum_{i=1}^{K} u_{ik}^m D(z_i, x_k),$$

where $m$ is the fuzzy exponent, $U = [u_{ik}]$ denotes the $K \times n$ fuzzy partition matrix and $u_{ik}$ (between 0 and 1) denotes the membership degree of $k$th object to the $i$th cluster. $Z = \{z_1, z_2, \ldots, z_K\}$ represents the set of cluster medoids. For probabilistic clustering, $\sum_{i=1}^{K} u_{ik} = 1, \forall k \in \{1, 2, \ldots, n\}$.

Fuzzy k-medoids algorithm involves in iteratively estimating the partition matrix followed by computation of new cluster medoids. It starts with random initial $K$ modes, and then at every iteration it finds the fuzzy membership of each object to every cluster using the following equation [9]:

$$u_{ik} = \frac{1}{\sum_{j=1}^{K} \left( \frac{D(z_j, x_k)}{D(z_j, x_k)} \right)^{\frac{1}{m}}}, \quad \text{for} \quad 1 \leq i \leq K; \quad 1 \leq k \leq n,$$

Note that while computing $u_{ik}$ using Eqn. 10, if $D(z_j, x_k)$ is equal to zero for some $j$, then $u_{ik}$ is set to zero for all $i = 1, \ldots, K$, $i \neq j$, while $u_{ik}$ is set equal to one.

Based on the membership values, the cluster medoids are recomputed as following: The medoid $z_i$ of the $i$th cluster will be $z_i = x_p$, where,

$$p = \arg \min_{1 \leq j \leq n} \sum_{k=1}^{n} u_{ik}^m D(x_j, x_k).$$

The algorithm terminates when there is no significant improvement in $J_m$ value (Eqn. 9). Finally, each object is assigned to the cluster to which it has maximum membership.

V. MULTIOBJECTIVE FUZZY BICLUSTERING

The proposed multiobjective GA based biclustering technique has been discussed in detail here. The proposed technique uses NSGA-II as the underlying multiobjective optimization technique.

A. String Representation

Each string has two parts: one for clustering the genes, and another for clustering the conditions. The first $M$ positions represent the $M$ cluster centers for the genes, and the remaining $N$ positions represent the $N$ cluster centers for the conditions. Thus a string looks like following:

$$\{gc_1, gc_2, \ldots, gc_M, cc_1, cc_2, \ldots, cc_N\},$$

where each $gc_i, i = 1 \ldots M$, represents the index of a gene that acts as a cluster medoid of a set of genes, and each $cc_j, j = 1 \ldots N$, represents the index of a condition that acts as a cluster medoid of a set of conditions. For a data set having $n$ points, it is usual to assume that the data set may contain at most $\sqrt{n}$ clusters. Taking this into account, for a $G \times C$ microarray, the values of the maximum number of gene clusters and the maximum number of condition clusters are $\lceil \sqrt{G} \rceil$ and $\lceil \sqrt{C} \rceil$, respectively. Therefore value of $M$ may vary from 2 to $\lceil \sqrt{G} \rceil$ and value of $N$ may vary from 2 to $\lceil \sqrt{C} \rceil$. Hence the length of the strings will vary from 4 to $\lceil \sqrt{G} \rceil + \lceil \sqrt{C} \rceil$. The first $M$ positions can have values in the range $\{1, 2, \ldots, G\}$ and the next $N$ positions can have values in the range $\{1, 2, \ldots, C\}$. Hence the gene and condition cluster medoids are represented by indices of the genes and conditions, respectively.

A string that encodes $A$ gene clusters and $B$ condition clusters, represents a set of $A \times B$ biclusters, taking each pair of gene and condition clusters. Each pair $<gc_i, cc_j>$, $i = 1 \ldots M$, $j = 1 \ldots N$, represents a bicluster that consists of all genes of the gene cluster centered at gene $gc_i$, and all conditions of the condition cluster centered at condition $cc_j$.

B. Initial Population

The individuals of the initial population are generated randomly, and each gene or condition has the equal probability to become the cluster medoid for a gene cluster or a condition cluster, respectively. The population size is kept fixed across all the generations.
C. Fitness Computation

Given a valid string (i.e., the string contains no repetition of gene or condition indices, and \( 2 \leq M \leq \sqrt{G} \), \( 2 \leq N \leq \sqrt{G} \)), first all the gene clusters and the condition clusters encoded in it are extracted. Thereafter, the fuzzy membership \( \mu_{ik} \) for each gene \( g_k \) to \( i \)th gene cluster \( gc_i \) is computed as follows:

\[
\mu_{ik} = \frac{1}{\sum_{j=1}^{M} D(g_k, gc_i)} \text{, for } 1 \leq i \leq M; \ 1 \leq k \leq G. 
\]  
(12)

Here \( D(., .) \) is a distance function. In this article, Euclidean distance measure has been adopted. Subsequently, the gene index \( gc_i \) representing the \( i \)th gene cluster is replaced by \( q_i \), where,

\[
q_i = \arg \min_{1 \leq j \leq k} \sum_{k=1}^{G} \mu_{ik} D(g_j, g_k). 
\]  
(13)

Also the membership matrix for genes is recomputed.

The fuzzy membership \( \eta_{jk} \) for each condition \( c_k \) to \( i \)th condition cluster \( cc_i \) is computed as:

\[
\eta_{jk} = \frac{1}{\sum_{j=1}^{N} D(cc_i, cc_j)} \text{, for } 1 \leq i \leq N; \ 1 \leq k \leq C. 
\]  
(14)

Next, the condition index \( cc_i \) representing the \( i \)th condition cluster is replaced by \( r_i \), where,

\[
r_i = \arg \min_{1 \leq j \leq k} \sum_{k=1}^{C} \eta_{jk} D(c_j, c_k). 
\]  
(15)

Finally, the membership matrix for conditions is recomputed.

Three properties of a bicluster have been optimized simultaneously: fuzzy volume, fuzzy mean squared residue and fuzzy expression profile variance. The volume of bicluster \( B(I, J) \) corresponding to \( < gc_x, cc_y > \) pair is defined as:

\[
\text{fvol}(I, J) = \sum_{i=1}^{G} \sum_{j=1}^{C} \mu_{ij}^m \eta_{ij}^m. 
\]  
(16)

Here \( I \) is a fuzzy set corresponding to fuzzy gene cluster centered at \( gc_x \). It consists of all genes \( g_x \) with membership degree \( \mu_{ij} \), \( 1 \leq i \leq G \). Similarly, \( J \) is a fuzzy set corresponding to fuzzy condition cluster centered at \( cc_y \). It consists of all conditions \( c_y \) with membership degree \( \eta_{ij} \), \( 1 \leq j \leq C \).

Residue of an element \( a_{ij} \) of the fuzzy bicluster \( B(I, J) \) is defined as:

\[
f_{r_{ij}} = a_{ij} - a_{i\cdot} - a_{\cdot j} + a_{\cdot \cdot} 
\]  
(17)

where

\[
a_{i\cdot} = \frac{\sum_{j=1}^{C} \eta_{ij} \mu_{ij}}{\sum_{j=1}^{C} \eta_{ij}^m}, \ 
\]  
and

\[
a_{\cdot j} = \frac{\sum_{i=1}^{G} \mu_{ij} \eta_{ij}}{\sum_{i=1}^{G} \mu_{ij}^m}, \ 
\]  
and

\[
a_{ij} = \frac{\sum_{i=1}^{G} \sum_{j=1}^{C} \mu_{ij}^m \eta_{ij}^m a_{ij}}{\sum_{i=1}^{G} \sum_{j=1}^{C} \mu_{ij}^m \eta_{ij}^m}, \ 
\]

The fuzzy mean squared residue (\( \text{FMSR}(I, J) \)) of the fuzzy bicluster \( B = (I, J) \) is defined as:

\[
\text{FMSR}(I, J) = \frac{1}{\text{fvol}(I, J)} \sum_{i=1}^{G} \sum_{j=1}^{C} \mu_{ij}^m \eta_{ij}^m f_{r_{ij}}^2. 
\]  
(18)

The third objective, i.e. fuzzy expression profile variance of \( B(I, J) \) is computed as:

\[
f_{\text{var}}(I, J) = \frac{1}{\text{fvol}(I, J)} \sum_{i=1}^{G} \sum_{j=1}^{C} \mu_{ij}^m \eta_{ij}^m (a_{ij} - a_{\cdot \cdot})^2. 
\]  
(19)

All the fuzzy biclusters represented by each gene-cluster condition-cluster pair \( < gc_x, cc_y > \), \( 1 \leq x \leq M \), \( 1 \leq y \leq N \) are extracted and the three objective functions \( \text{fvol}, \text{FMSR} \) and \( f_{\text{var}} \) are computed for each of them as mentioned above. Note that, \( \text{FMSR} \) is to be minimized, whereas, \( \text{fvol} \) and \( f_{\text{var}} \) are to be maximized to have good quality large and non-trivial biclusters. As the proposed multiobjective technique is posed as a minimization algorithm, hence the three objective functions \((f_1, f_2, f_3)\) are taken as follows:

\[
f_1(I, J) = \text{FMSR}(I, J), 
\]  
(20)

\[
f_2(I, J) = \frac{1}{\text{fvol}(I, J)}, 
\]  
(21)

and

\[
f_3(I, J) = \frac{1}{1 + f_{\text{var}}(I, J)}. 
\]  
(22)

The denominator of \( f_3 \) is chosen such way to avoid accidental divide by zero condition when expression profile variance becomes 0. Note that all of \( f_1, f_2 \) and \( f_3 \) are to be minimized to obtain highly coherent, large yet non-trivial biclusters. For each encoded fuzzy bicluster, the fitness vector \( \overline{f} = \{f_1, f_2, f_3\} \) is computed. The fitness vector of a string is then the average of the fitness vectors of all fuzzy biclusters encoded in it.

Note that due to randomness of the genetic operators, invalid strings with repeated gene and/or condition indices may arise at any point of the algorithm. The invalid strings are given fitness vector \( \overline{f} = \{X, X, X\} \), where \( X \) is an arbitrary large number. Thus the invalid strings will be automatically out of the competition in subsequent generations.

D. Selection

The selection operation used here is the crowded binary tournament selection method. This is the original selection operator that is used in NSGA-II to evolve the mating pool. The reader may refer to [14] for the details of the selection procedure.
E. Crossover

Single point crossover is used depending on the crossover probability p_c. Each part (gene part and condition part) of the string undergoes crossover separately. Suppose M_i and N_i are the lengths of the gene part and condition part of the parent P_i, respectively. Let parent chromosomes P_1 and P_2 encode M_1 and M_2 gene clusters, respectively. The crossover point in P_1, λ_1 is generated as a random integer in the range 2 ≤ λ_1 ≤ M_1. Let λ_2 be the crossover point in P_2, and it may vary in between [λ_2; λ_2′], where λ_2 and λ_2′ indicate the lower and upper bounds of the range of λ_2, respectively. λ_1 and λ_2 are given by

\[
\lambda_1 = \min(2; \max(0; 2 - (M_1 - \lambda_1))], \quad \text{and} \quad (23)
\]

\[
\lambda_2 = [M_2 - \max(0; (2 - \lambda_1))]. \quad (24)
\]

If λ_2 ≥ λ_1, then λ_2 is generated randomly as an integer between λ_2 and λ_2′. Otherwise, λ_2 = 0. Simple calculations show that if crossover points λ_1 and λ_2 are chosen as per above rule, then number of gene clusters in each of the offsprings will not fall below 2. Similarly the condition parts of the parent chromosomes undergo crossover to generate two children. Fig. 1 illustrates the crossover operation.

F. Mutation

The mutation operation works as follows. A random position is chosen from the first M positions and its value is replaced by an index randomly chosen from the range \{1, 2, ..., G\}, where G is the total number of genes. Similarly, to mutate the condition portion of the string, a random position is selected from the next N positions and its value is substituted using a randomly selected index from the range \{1, 2, ..., C\}, where C is the total number of conditions. A string undergoes mutation depending on a mutation probability p_m. For an example, consider the following string:

\[
\text{Parent} : 1\ 10\ 25\ 23\ 8\ 2|3\ 10\ 2\ 4.
\]

The mutated positions are shown in bold faces.

G. Elitism

Elitism means propagation of non-dominated solutions in the parent and child populations to the next generation. Elitism is used to track the non-dominated Pareto-optimal solutions found so far.

H. Final Solutions

The proposed algorithm has been executed for a fixed number of generations with a fixed population size. When the final generation completes, the fuzzy biclusters are extracted from the non-dominated strings. For a gene-cluster condition-cluster pair \( g_{cx}, c_{gy} \), the fuzzy bicluster \( B(I, J) \) can be interpreted as a regular bicluster \( B'(I', J') \), where

\[
I' = \{ i\mid p_{xi} > = \alpha_p \}, \quad 1 \leq i \leq G, \quad (25)
\]

and

\[
J' = \{ j\mid y_{yj} > = \alpha_c \}, \quad 1 \leq j \leq C. \quad (26)
\]

Here \( \alpha_p \) and \( \alpha_c \) denote two threshold parameters on membership degrees. The threshold \( \alpha_p \) is taken as \( 1/M \) and \( \alpha_c \) is taken as \( 1/N \), where \( M \) and \( N \) are the number of gene clusters and condition clusters encoded in the string, respectively.

After extracting the regular biclusters from the fuzzy ones, all the \( \delta \)-biclusters are returned by the proposed method.

VI. COMPETITIVE BICLUSTERING ALGORITHMS

In recent years, several biclustering strategies have been developed by researchers. In this article, two well known biclustering algorithms, viz., Cheng and Church’s algorithm (CC) [7] and Random Walk Biclustering algorithm (RWB) [17], have been considered for the purpose of comparison. This section briefly describes above two biclustering techniques.

A. Cheng and Church’s Algorithm (CC)

In this algorithm [7], in order to identify the largest \( \delta \)-bicluster in a data set, a two-phase approach has been adopted. The first phase consists of removing the rows and columns of the data set on the basis of a greedy search until the \( \delta \)-constraint is met. The subsequent phase tries to add the previously removed rows and columns as long as the mean squared residue score is below the \( \delta \) value. To find more \( k \) biclusters, the process is iterated \( k \) times, each time masking the previously found biclusters using random values.
B. Random Walk Biclustering Algorithm (RWB)

In this algorithm [17], a greedy search technique enriched by a local search strategy to escape local optima has been presented. The algorithm begins with initial random solution and searches for a locally optimal solution by successive transformations (including random moves depending on some probability) to improve a gain function defined as a combination of mean squared residue, expression profile variance and the volume of the biclusters. The algorithm iterates \( k \) times to generate \( k \) biclusters.

VII. EXPERIMENTS AND RESULTS

The performance of the proposed multiobjective algorithm has been compared with that of its single objective version that minimizes \( f_1 \times f_2 \times f_3 \), Cheng and Church’s algorithm (CC) [7]), Random Walk Biclustering algorithm (RWB) [17]

Two real life data sets, i.e., Yeast and Human are used for this purpose. Data sets are described here.

1) Yeast Cell Cycle Data: This data set [7] contains 2884 genes and 17 conditions. Each element has a value in the range \([0, 1, \ldots, 600]\). There are two rows with missing values denoted by -1. These rows are omitted from the data set to form a data matrix of size \(2882 \times 17\). This is done to avoid random interference (putting random values for missing values) [24]. The data set is then normalized so that each row has mean 0 and variance 1. Note that although the normalized data matrix is used for distance calculation, to compute the FMSR and \( f_{vol} \) and \( f_{var} \), the original data matrix is used. Normalization helps the clustering process to work better [25], [26].

2) Human Large B-cell Lymphoma Data: There are 4026 genes and 96 conditions in this data set [7]. This data matrix has elements with expression values ranging from -750 to 650. There are 47639 missing values (12.3% of the matrix elements) represented by 999. The rows with missing values have been removed to reduce the data matrix to a smaller size of \(854 \times 96\), as in [24]. Subsequently, the data set is normalized so that each row of the matrix has mean 0 and variance 1.

Both the real data sets are publicly available at http://arep.med.harvard.edu/biclustering.

A. Parameter Settings

The proposed algorithm has been executed for 50 generations with population size 50. The crossover and mutation probabilities are set to 0.8 and 0.1, respectively, while the value of the fuzzy exponent, \( m \), is set to 2.0. These values are chosen after several experiments. For comparison, the \( \delta \) values (i.e. the maximum allowable mean squared residue) for Yeast and Human data sets are 300 and 1200, respectively, as selected in [7]. The other algorithms are executed with parameters suggested in respective articles.

B. Performance Metrics

Following three performance metrics have been utilized for performance comparison.

1) Bicluster Index: This measure is an estimation of goodness of biclusters in terms of \( MSR \), volume and expression profile variance. For a set of \( n \) biclusters, we define the Bicluster Index (\( BI \)) as follows:

\[
BI = \frac{1}{n} \sum_{i=1}^{n} \frac{MSR_i}{\sqrt{\delta}} \times (1 + \text{var}_i). \tag{27}
\]

Here \( G \) and \( C \) denote the number of genes and the number of conditions in the data matrix, respectively. Note that lower value of \( BI \) denotes better set of biclusters.

2) Coverage: Coverage is defined as the percentage of cells covered by \( n \) biclusters in the microarray matrix. Larger coverage indicates biclusters have covered larger area of the data matrix.

3) Statistical Difference from Background: This is a new index proposed here to find whether a set of \( n \) biclusters are statistically different from the background data matrix. Let \( MSR^{(g,c)}_i \) denotes the \( MSR \) of the \( i^{th} \) bicluster having \( g \) genes and \( c \) condition. \( RMSR^{(g,c)}_i \) denotes the \( MSR \) of the \( j^{th} \) bicluster that is built by randomly choosing \( g \) genes and \( c \) conditions from the data matrix. Now we define the Statistical Difference from Background (\( SDB \)) as follows:

\[
SDB = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{\sum_{j=1}^{G \times C} MSR^{(g,c)}_i - MSR^{(g,c)}_j}{\sqrt{\delta}} \right) \tag{28}
\]

Note that the numerator of Eqn. 28 is the \( MSR \) of the \( i^{th} \) bicluster which has lower value when the bicluster has higher coherence. In the denominator, the \( MSR \) of the \( i^{th} \) bicluster is subtracted from the average \( MSR \) of \( r \) random biclusters of same size. Thus the denominator signifies how the \( i^{th} \) bicluster is statistically different from the random background. Larger the values of the denominator, better is the difference between the bicluster and background. Hence, on the whole, the lower value of \( SDB \) is desirable.

C. Results on Yeast Data

In Figure 2, 8 biclusters among all biclusters found in a typical run of the proposed algorithm. It is evident from the figure that the proposed algorithm is able to discover non-trivial biclusters having high expression profile variance. In each of the biclusters, all the genes in the set have similar expression profile (i.e., each bicluster has low \( MSR \)).

To aid the comparative study, Table I summarizes the biclustering results of 100 best biclusters (in terms of \( BI \)) obtained by different algorithms over 10 consecutive runs. The table reports the average values for \( BI \), coverage and \( SDB \) of the biclusters in 10 consecutive runs along with the corresponding standard deviations in brackets. As evident from the table, The proposed multiobjective algorithm provides better \( BI \), coverage and \( SDB \) values compared to the other algorithms used in this study. Also the low standard deviations for the proposed algorithm confirms its stable behavior compared to the single objective version and RWB.
TABLE I
COMPARISON OF BICLUSTERS FOR Yeast DATA: AVERAGE VALUES OF EACH PERFORMANCE METRIC ARE REPORTED OVER 10 CONSECUTIVE RUNS FOR EACH ALGORITHM. BRACKETED VALUES REPRESENT THE CORRESPONDING STANDARD DEVIATIONS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Bk</th>
<th>Coverage</th>
<th>SDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiobjective</td>
<td>0.1279 (0.09)</td>
<td>71.09% (3.24)</td>
<td>0.3259 (0.05)</td>
</tr>
<tr>
<td>Single objective</td>
<td>0.3358 (1.12)</td>
<td>62.54% (4.02)</td>
<td>0.5741 (0.23)</td>
</tr>
<tr>
<td>CC</td>
<td>1.3326 (0.10)</td>
<td>61.78% (2.83)</td>
<td>0.4985 (0.05)</td>
</tr>
<tr>
<td>RWB</td>
<td>0.6013 (1.73)</td>
<td>50.35% (8.92)</td>
<td>0.5077 (0.34)</td>
</tr>
</tbody>
</table>

TABLE II
COMPARISON OF BICLUSTERS FOR Human DATA: AVERAGE VALUES OF EACH PERFORMANCE METRIC ARE REPORTED OVER 10 CONSECUTIVE RUNS FOR EACH ALGORITHM. BRACKETED VALUES REPRESENT THE CORRESPONDING STANDARD DEVIATIONS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Bk</th>
<th>Coverage</th>
<th>SDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiobjective</td>
<td>0.0558 (0.04)</td>
<td>55.07% (3.16)</td>
<td>0.3128 (0.03)</td>
</tr>
<tr>
<td>Single objective</td>
<td>0.1337 (0.17)</td>
<td>51.23% (3.22)</td>
<td>0.3302 (0.11)</td>
</tr>
<tr>
<td>CC</td>
<td>0.3886 (0.04)</td>
<td>46.93% (3.03)</td>
<td>0.3688 (0.04)</td>
</tr>
<tr>
<td>RWB</td>
<td>0.1801 (0.57)</td>
<td>31.18% (6.83)</td>
<td>0.3364 (0.28)</td>
</tr>
</tbody>
</table>

VIII. CONCLUSIONS

In this article, a NSGA-II based multiobjective fuzzy biclustering methodology has been proposed that optimizes mean squared residue, volume and expression profile variance simultaneously. In this regard, a variable string length novel encoding scheme has been proposed. A new performance measure that reflects how the biclusters are statistically distinguished from the background data matrix is also proposed. The performance of the proposed technique has been demonstrated on two benchmark real life microarray data sets extensively used in literature. Also a comparative study with single objective version and other two well known biclustering methods has been made to establish its effectiveness. Although the proposed GA based method has greater time complexity compared to Cheng and Church’s algorithm, it has the advantage that it generates a set of biclusters in a single run. The proposed algorithm has been shown to outperform other algorithms in finding coherent and non-trivial biclusters that are statistically different from the background data matrix.

ACKNOWLEDGMENT

This work is partially supported by NSFC grants 60575014 and 30625012 and the 863 project (2006AA02Z325) of...
China. Michael Zhang acknowledges Changjiang Guest Professorship Award of China and GM74688.

REFERENCES


[17] F. Angiulli, E. Cesarini, and C. Pizzuti, “Gene expression biclustering using random walk strategies,” in *Proc. 7th Int. Conf. on Data Warehousing and Knowledge Discovery (DAWAK’03),* Copenhagen, Denmark, 2005.


