Pattern Recognition 39 (2006) 2464–2477

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

Multi-objective evolutionary biclustering of gene expression data

Sushmita Mitra, Haider Banka*

Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700 108, India

Received 3 November 2005; received in revised form 3 February 2006; accepted 1 March 2006

Abstract

Biclustering or simultaneous clustering of both genes and conditions have generated considerable interest over the past few decades, particularly related to the analysis of high-dimensional gene expression data in information retrieval, knowledge discovery, and data mining. The objective is to find sub-matrices, i.e., maximal subgroups of genes and subgroups of conditions where the genes exhibit highly correlated activities over a range of conditions. Since these two objectives are mutually conflicting, they become suitable candidates for multi-objective modeling. In this study, a novel multi-objective evolutionary biclustering framework is introduced by incorporating local search strategies. A new quantitative measure to evaluate the goodness of the biclusters is developed. The experimental results on benchmark datasets demonstrate better performance as compared to existing algorithms available in literature.

Keywords: Multi-objective optimization; Microarray; Genetic algorithms; Knowledge discovery; Clustering

1. Introduction

Microarray experiments produce gene expression patterns that offer enormous information about cell function. This is useful while investigating complex interactions within the cell [1]. Microarrays are used in the medical domain to produce molecular profiles of diseased and normal tissues of patients. Such profiles are useful for understanding various diseases, and aid in more accurate diagnosis, prognosis, treatment planning, as well as drug discovery. Being typically high-dimensional, gene expression data requires appropriate mining strategies like feature selection and clustering [2] for further analysis.

Biological networks relate genes, gene products or their groups (like protein complexes or protein families) to each other in the form of a graph. Clustering of gene expression patterns are being used to generate gene regulatory networks [3]. A major cause of coexpression of genes is their sharing of the regulation mechanism (coregulation) at the sequence level. Clustering of coexpressed genes, into biologically meaningful groups, helps in inferring the biological role of an unknown gene that is coexpressed with a known gene(s).

A cluster is a collection of data objects which are similar to one another within the same cluster but dissimilar to the objects in other clusters [4]. The problem is to group N patterns into n_c possible clusters with high *intra-class* similarity and low *inter-class* similarity by optimizing an objective function. In objective function-based clustering algorithms, the goal is to find a partition for a given value of n_c. Clustering in gene expression data includes partitional, hierarchical, grid-based and density-based approaches to clustering [5] to name a few. Here the genes are typically partitioned into disjoint or overlapped groups according to the similarity of their expression patterns over *all* conditions.

It is often observed that a subset of genes are coregulated and coexpressed under a subset of conditions, but behave almost independently under other conditions. Here the term "conditions" can imply environmental conditions as well as time points corresponding to one or more such environmental conditions. Biclustering attempts to discover such local structure inherent in the gene expression matrix. It refers to the simultaneous clustering of both genes and conditions in the process of knowledge discovery about local patterns from microarray data [6]. This also allows detection of overlapped groupings among the biclusters, thereby providing a better representation of the biological reality involving genes with multiple functions or those regulated by many factors. For example, a single gene may participate in multiple pathways that may or may not be coactive under all conditions.

It may be noted that clustering approaches compute global models, while biclustering techniques focus on local models. Some of the existing nomenclature for biclustering, particularly in other application fields, are bidimensional clustering, subspace clustering and coclustering.

1.1. Related work

There has been a lot of research in biclustering [7–9] involving statistical and graph-theoretic techniques. The pioneering work by Cheng and Church [6] employs a set of heuristic algorithms to find one or more biclusters in gene expression data, based on a uniformity criteria. One bicluster is identified at a time, iteratively. There are iterations of masking null values and discovered biclusters (replacing relevant cells with random numbers), coarse and fine node deletion, node addition, and the inclusion of inverted data. The computational complexity for discovering k biclusters is of the order of $O(mn \times (m+n) \times k)$, where m and n are the number of genes and conditions, respectively.

Sometimes the masking procedure may result in a phenomenon of *random interference*, thereby adversely affecting the subsequent discovery of high quality biclusters. In order to circumvent this problem, a two-phase probabilistic algorithm termed flexible overlapped clusters (FLOC) [10] is designed to simultaneously discover a set of possibly overlapping biclusters. Initial biclusters (or seeds) are chosen randomly from the original data matrix. Iterative gene and/or condition additions and/or deletions are performed with a goal of achieving the best potential residue reduction. The time complexity of FLOC is lower for p iterations ($p \ll n+m$); i.e., $O((n+m)^2 \times k \times p)$.

The Plaid model [11] tries to capture the approximate uniformity in a submatrix of the gene expression data, while discovering one bicluster at a time in an iterative process. The input matrix is described as a linear function of variables corresponding to its biclusters, and an iterative maximization process is pursued for estimating the function. It searches for patterns where the genes differ in their expression levels by a constant factor.

Bipartite graphs are employed in Ref. [12], with a bicluster being defined as a subset of genes that jointly respond across a subset of conditions. The objective is to identify the maximum-weighted subgraph. Here a gene is considered to be responding under a condition if its expression level changes significantly, under that condition over the connecting edge, with respect to its normal level. This involves an exhaustive enumeration, with a restriction on the number of genes that can appear in the bicluster. A simultaneous discovery of all biclusters is made at the same time. It may be noted that in all these methods it is possible to generate overlapped gene clusters.

A coupled two-way iterative method [13] has been devised to iteratively generate a set of biclusters, at a time, in cancer datasets. In the process it repeatedly performs one-way hierarchical clustering on the rows and columns of the data matrix, while using stable clusters of rows as attributes for column clustering and vice versa. The Euclidean distance is used as the similarity measure, after normalization of the data.

Gene ontology (GO) information, involving hierarchical functional relationships like “part of”, “overlapping”, has been incorporated into the clustering process called smart hierarchical tendency preserving algorithm (SHTP) [14]. A fast approximate pattern matching technique has been employed [15] to determine maximum sized biclusters with a number of conditions greater than a specified minimum. The worst case complexity of the procedure is claimed to be $O(m^2n)$. Rich probabilistic models have been used [16] for discovering relations between expressions, regulatory motifs and gene annotations. The outcome is a collection of disjoint biclusters, generated in a supervised manner.

Efficient techniques have been successfully amalgamated in the deterministic biclustering with frequent pattern mining algorithm (DBF) [17] to generate a set of good quality biclusters. Here the changing tendency between two conditions is modeled as an item, with the genes corresponding to transactions. A frequent itemset with the supporting genes forms a bicluster. In the second phase, these are iteratively refined by adding more genes and/or conditions.

A good survey on biclustering is available in literature [18], with a categorization of the different heuristic approaches made as follows:

- Iterative row and column clustering combination [13]: apply clustering algorithms to the rows and columns of the data matrix, separately, and then combine the results using some iterative procedure.
- Divide and conquer [7]: break the problem into smaller sub-problems, solve them recursively, and combine the solutions to solve the original problem.
- Greedy iterative search [6,10]: make a locally optimal choice, in the hope that this will lead to a globally good solution.
- Exhaustive biclustering enumeration: the best biclusters are identified, using an exhaustive enumeration of all possible biclusters existent in the data, in exponential time [12].
- Distribution parameter identification [11]: identify best-fitting parameters by minimizing a criterion through an iterative approach.

There exist a number of investigations dealing with time-series data [19,20]. However, in this study, we will not be concerned with differentiating between time-course and condition-based gene expression data.

A greedy local search heuristic for biclustering has been reported in literature [6]. Here similarity is computed as a measure of the coherence of the genes and conditions in the bicluster. Although the greedy local search methods are by themselves fast, but they often yield suboptimal solutions.

1.2. Role of genetic algorithms

The quality of a biclustering is often considered to be more important than the computation time required to generate it. Hence genetic algorithms (GAs) [21] provide an alternative efficient search technique in a large solution space, based on the theory of evolution. GAs involve a set of evolutionary operators, like selection, crossover and mutation. A population of chromosomes is made to evolve over generations by optimizing a fitness function, which provides a quantitative measure of the fitness of individuals in the pool. Single-objective GA, with local search, has been employed for identifying biclusters in gene expression data [22].

A simulated annealing (SA) based biclustering algorithm [23] is found to provide improved performance over that of Ref. [6], and is also able to escape from local minima. Unlike the classical optimization techniques like GA, that appreciate only improvements in the chosen fitness functions, SA also allows a probabilistic acceptance of temporary disimprovement in fitness scores. However, the results are often data dependent.

When there are two or more conflicting characteristics to be optimized, the single-objective GA requires an appropriate formulation of the single fitness function in terms of an aggregation of the different criteria involved. In such situations *multi-objective evolutionary algorithms* (MOEAs) [24] provide an alternative, more efficient, approach to searching for optimal solutions. They have found successful application in feature selection and classification of microarray gene expression patterns [25].

In this paper we investigate the use of MOEA, in conjunction with local search heuristics, while generating and iteratively refining an optimal set of biclusters. Here the objective is to find one or more biclusters that are optimal with respect to their homogeneity and size. Since these two criteria are usually conflicting, this lead us to formulate the biclustering problem in a multi-objective framework. The fitness functions are formulated as a pair, consisting of the mean squared residue score [6] and the size of the bicluster.

The remaining part of this article is organized as follows. Section 2 introduces the preliminaries of gene expression data, biclustering and MOEA. The proposed multi-objective GA is presented in Section 3. A new quantitative measure for evaluating the goodness of the biclusters is proposed in Section 4. Comparative results, along with statistical significance for biological relevance, are provided in Section 5 on benchmark gene expression datasets. Section 6 concludes the article.

2. Preliminaries

In this section we briefly discuss the basic concepts of microarray gene expression data, biclustering and MOEA.

2.1. Microarray and gene expression data

Reverse transcribed mRNA or cDNA microarrays (gene arrays or gene chips) [1] usually consist of thin glass or nylon substrates containing specific DNA gene samples spotted in an array by a robotic printing device. This measures the relative mRNA abundance between two samples, which are labeled with different fluorescent dyes viz. red and green. The mRNA binds (hybridizes) with cDNA probes on the array. The relative abundance of a spot or gene is measured as the logarithmic ratio between the intensities of the dyes, and constitutes the gene expression data.

Gene expression levels can be determined for samples taken (i) at multiple time instants of a biological process (different phases of cell division) or (ii) under various conditions (e.g., tumor samples with different histopathological diagnosis). Each gene corresponds to a high-dimensional vector of its expression profile. The data contain a high level of noise due to experimental procedures. Moreover, the expression values of single genes demonstrate large biological variance within tissue samples from the same class. Fig. 1 provides a schematic diagram of a microarray, depicting the gene expression matrix. Here the rows correspond to the gene expression levels of each sample.

2.2. Biclustering

A bicluster is defined as a pair (g, c), where $g \subseteq \{1, \ldots, m\}$ is a subset of genes and $c \subseteq \{1, \ldots, n\}$ is a subset of conditions. The optimization task [6] is to find the largest bicluster that does not exceed a certain homogeneity constraint stated below. The size (or volume) $f(g, c)$ of a bicluster is defined as the number of cells in the gene expression matrix E (with values e_{ij}) that are covered by it. The homogeneity $\mathscr{G}(g, c)$ is expressed as a mean squared residue score. We maximize

$$f(g, c) = |g| \times |c|, \tag{1}$$

subject to a low $\mathscr{G}(g, c) \leqslant \delta$ for $(g, c) \in X$, with $X = 2^{\{1,\ldots,m\}} \times 2^{\{1,\ldots,n\}}$ being the set of all biclusters, where

$$\mathscr{G}(g, c) = \frac{1}{|g| \times |c|} \sum_{i \in g, j \in c} (e_{ij} - e_{ic} - e_{gj} + e_{gc})^2. \tag{2}$$

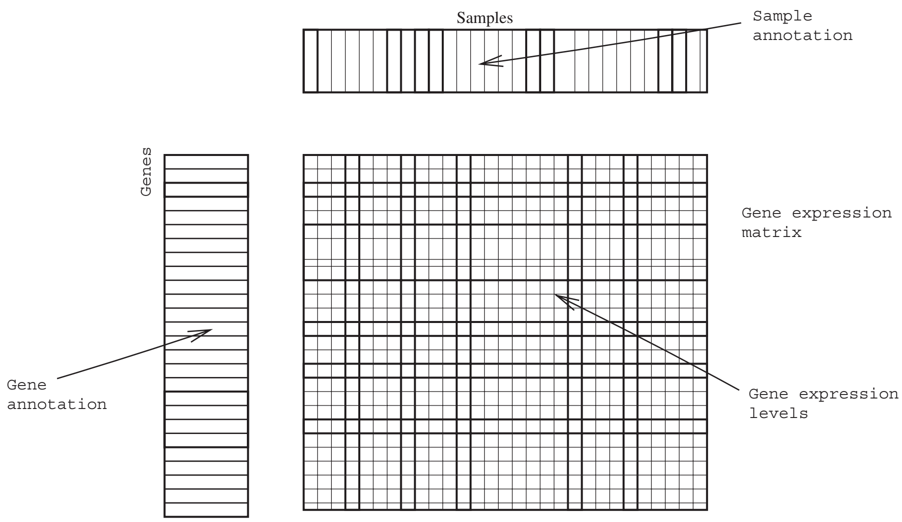

Fig. 1. Schematic diagram of a microarray.

Here

$$e_{ic} = \frac{1}{|c|} \sum_{j \in c} e_{ij}, \tag{3}$$

$$e_{gj} = \frac{1}{|g|} \sum_{i \in g} e_{ij} \tag{4}$$

are the mean row and column expression values for (g, c) and

$$e_{gc} = \frac{1}{|g| \times |c|} \sum_{i \in g, j \in c} e_{ij} \tag{5}$$

is the mean expression value over all cells contained in the bicluster (g, c). A user-defined threshold δ represents the maximum allowable dissimilarity within the bicluster. In other words, the residue quantifies the difference between the actual value of an element e_{ij} and its expected value as predicted from the corresponding row mean, column mean, and bicluster mean. A set of genes whose expression levels change in accordance to each other over a set of conditions can thus form a perfect bicluster even if the actual values lie far apart. For a good bicluster, we have $\mathscr{G}(g, c) < \delta$ for some $\delta \geqslant 0$.

The optimization task of finding one or more biclusters by maintaining the two competing constraints, *viz.*, homogeneity and size, is reported to be NP-complete [26]. The high complexity of this problem has motivated researchers to apply various approximation techniques to generate near optimal solutions.

2.3. Multi-objective EAs

Most real-world search and optimization problems typically involve multiple objectives. A solution that is better with respect to one objective requires a compromise in other objectives. In problems with more than one conflicting objective there exists no single optimum solution. Rather, there exists a set of solutions which are all optimal involving trade-offs between conflicting objectives.

Unlike single-objective optimization problems, the MOEA tries to optimize two or more conflicting characteristics represented by fitness functions. Modeling this situation with single-objective GA would amount to heuristic determination of a number of parameters involved in expressing such a scalar-combination-type fitness function. MOEA, on the other hand, generates a set of *Pareto-optimal* solutions [24] which simultaneously optimize the conflicting requirements of the multiple fitness functions.

Among the different multi-objective algorithms, it is observed that non-dominated sorting genetic algorithm (NSGA-II) [27] possesses all the features required for a good MOEA. It has been shown that this can converge to the global Pareto front, while simultaneously maintaining the diversity of population. We describe here the characteristics of NSGA-II, like non-domination, crowding distance and crowding selection operator. This is followed by the actual algorithm.

2.3.1. Non-domination

The concept of optimality, behind the multi-objective optimization, deals with a set of solutions. The conditions for a solution to be *dominated* with respect to the other solutions are given below.

Definition 1. If there are M objective functions, a solution $x^{(1)}$ is said to dominate another solution $x^{(2)}$, if both conditions (a) and (b) are true:

(a) The solution $x^{(1)}$ is no worse than $x^{(2)}$ in all the M objective functions.

(b) The solution $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one of the M objective functions.

Otherwise the two solutions are *non-dominating* to each other. When a solution i dominates solution j, then **rank** $r_i < r_j$.

The major steps for finding the non-dominated set in a population P of size $|P|$ are outlined below.

(i) Set solution counter $i = 1$ and create an empty non-dominated set P'.
(ii) **For** a solution $j \in P$ $(j \neq i)$, check if solution j dominates solution i.
If yes **then go to** Step 4.
(iii) **If** more solutions are left in P, increment j by one and **go to** Step 2.
Else set $P' = P' \cup \{i\}$.
(iv) Increment i by one.
If $i \leqslant |P|$ **then go to** Step 2 **else** declare P' as the non-dominated set.

After all the solutions of P are checked, the members of P' constitute the non-dominated set at the first level (front with rank $= 1$). In order to generate solutions for the next higher level (dominated by the first level), the above procedure is repeated on the reduced population $P = P - P'$. This is iteratively continued until $P = \emptyset$. The complexity of non-dominated sorting at each iteration is $O(M \times P^2)$.

2.3.2. Crowding distance

In order to maintain diversity in the population, a measure called *crowding distance* is used. This assigns the highest value to the boundary solutions and the average distance of two solutions [$(i+1)$th and $(i-1)$th] on either side of solution i along each of the objectives. The following algorithm computes the crowding distance d_i of each point in the front $\mathscr{F}$.

(i) Let the number of solutions in $\mathscr{F}$ be $l = |\mathscr{F}|$ and assign $d_i = 0$ for $i = 1, 2, \ldots, l$.
(ii) **For** each objective function $f_k, k = 1, 2, \ldots, M$, sort the set in its worse order.
(iii) Set $d_1 = d_l = \infty$.
(iv) **For** $j = 2$ to $(l-1)$ increment d_j by $f_{k_{j+1}} - f_{k_{j-1}}$.

The complexity incurred for calculating the crowding distance is $O(M \times P \log P)$.

2.3.3. Crowding selection operator

Crowded tournament selection operator is defined as follows. A solution i wins tournament with another solution j if any one of the following is true:

- Solution i has better rank, i.e., $r_i < r_j$.
- Both the solutions are in the same front, i.e., $r_i = r_j$, but solution i is less densely located in the search space, i.e., $d_i > d_j$.

2.3.4. NSGA-II

The multi-objective algorithm NSGA-II is characterized by the use of the above-mentioned three characteristics while generating the optimal solution. Let us now outline the main steps of NSGA-II [27].

(i) Initialize the population randomly.
(ii) Calculate the multi-objective fitness function.
(iii) Rank the population using the dominance criteria of Section 2.3.1.
(iv) Calculate the crowding distance based on Section 2.3.2.
(v) Do selection using crowding selection operator of Section 2.3.3.
(vi) Do crossover and mutation (as in conventional GA) to generate offspring population.
(vii) Combine parent and children population.
(viii) Replace the parent population by the best members of the combined population. Initially, members of lower fronts replace the parent population. When it is not possible to accommodate all the members of a particular front, then that front is sorted according to the crowding distance. Selection of individuals is done on the basis of higher crowding distance. The number selected is that required to make the new parent population size the same as the size of the old one.

The overall complexity of the above algorithm is $O(M \times P^2)$, and is mainly governed by the non-dominated sorting part.

3. Multi-objective biclustering

MOEA is a global search heuristic, primarily used for optimization tasks. In this section we present the general framework and implementation details of MOEA for biclustering. Local search heuristics are employed to speed up convergence by refining the chromosomes.

3.1. Representation

Each bicluster is represented by a fixed sized binary string called chromosome or individual, with a bit string for genes appended by another bit string for conditions. The chromosome corresponds to a solution for this optimal bicluster generation problem. A bit is set to one if the corresponding gene and/or condition is present in the bicluster, and reset to zero otherwise. Fig. 2 depicts such an encoding of genes and conditions in a chromosome.

The initial population is generated randomly. Uniform single-point crossover, single-bit mutation, and crowded

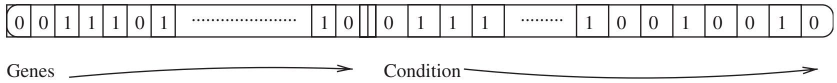

Fig. 2. An encoded chromosome representing a bicluster.

tournament selection (of Section 2.3.3) are employed in the multi-objective framework. Both parent and offspring population, in each generation, are combined to select the best members as the new parent population. Diversity is maintained within the biclusters by using the crowding distance operator (of Section 2.3.2).

3.2. Multi-objective framework

We observe here that one needs to concentrate on generating maximal set of genes and conditions while maintaining the "homogeneity" of the biclusters. These two characteristics of biclusters, being conflicting to each other, are well suited for multi-objective modeling. In order to optimize this pair of conflicting requirements, the fitness function f_1 is always maximized while function f_2 is maximized as long as the residue does not exceed the threshold δ. They are formulated as

$$f_1 = \frac{g \times c}{|G| \times |C|}, \tag{6}$$

$$f_2 = \begin{cases} \dfrac{\mathscr{G}(g,c)}{\delta} & \text{if } \mathscr{G}(g,c) \leqslant \delta, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where g and c are the number of ones in the genes and conditions within the bicluster, $\mathscr{G}(g,c)$ is its mean squared residue score as defined by Eqs. (2)–(5), δ is the user-defined threshold for the *maximum acceptable* dissimilarity or mean squared residue score of the bicluster, and G and C are the total number of genes and conditions of the original gene expression array.

Note that f_1 is maximum for $g = G$ and $c = C$, i.e., when the submatrix (g, c) is equal to the whole input dataset. Now as the size of the bicluster increases, so does the mean squared residue. Thereby f_2 is allowed to increase as long as it does not exceed the homogeneity constraint δ. Beyond this we assign a lower fitness value of zero to f_2, which is ultimately removed during non-dominated front selection of MOEA.

3.3. Local search

Since the initial biclusters are generated randomly, it may happen that some irrelevant genes and/or conditions get included in spite of their expression values lying far apart in the feature space. An analogous situation may also arise during crossover and mutation in each generation. These genes and conditions, with dissimilar values, need to be eliminated deterministically. Furthermore, for good biclustering, some genes and/or conditions having similar expression values need to be incorporated as well. In such situations, local search strategies [6] can be employed to add or remove multiple genes and/or conditions. It was observed that, in the absence of local search, stand-alone single-objective or MOEAs could not generate satisfactory solutions. The algorithm starts with a given bicluster and an initial gene expression array (G, C). The irrelevant genes or conditions having mean squared residue above (or below) a certain threshold are now selectively eliminated (or added) using the following conditions. A "node" refers to a gene or a condition in the sequel.

(i) Multiple nodes deletion.
  (a) Compute e_{ic}, e_{gj}, e_{gc} and $\mathscr{G}(g,c)$ of the bicluster by Eqs. (2)–(5).
  (b) Remove all genes $i \in g$ satisfying

$$\frac{1}{|c|}\sum_{i\in g}(e_{ij}-e_{ic}-e_{gj}+e_{gc})^2 > \alpha \times \mathscr{G}(g,c). \tag{8}$$

  (d) Recompute e_{ic}, e_{gj}, e_{gc} and $\mathscr{G}(g,c)$.
  Remove all conditions $j \in c$ satisfying

$$\frac{1}{|g|}\sum_{j\in c}(e_{ij}-e_{ic}-e_{gj}+e_{gc})^2 > \alpha \times \mathscr{G}(g,c). \tag{9}$$

(ii) Single node deletion, corresponding to a refinement of Step (i).
  (a) Recompute e_{ic}, e_{gj}, e_{gc} and $\mathscr{G}(g,c)$ of the modified bicluster by Step (i).
  (b) Remove the node with largest mean squared residue (done for both gene and condition), one at a time, until the mean squared residue drops below δ.

(iii) Multiple nodes addition.
  (a) Recompute e_{ic}, e_{gj}, e_{gc} and $\mathscr{G}(g,c)$ of the modified bicluster of Step (ii).
  (b) Add all genes $i \neq g$ satisfying

$$\frac{1}{|c|}\sum_{i\in g}(e_{ij}-e_{ic}-e_{gj}+e_{gc})^2 \leqslant \mathscr{G}(g,c). \tag{10}$$

  (c) Recompute e_{ic}, e_{gj}, e_{gc} and $\mathscr{G}(g,c)$.
  (d) Add all conditions $j \notin c$ satisfying

$$\frac{1}{|g|}\sum_{j\in c}(e_{ij}-e_{ic}-e_{gj}+e_{gc})^2 \leqslant \mathscr{G}(g,c). \tag{11}$$

It is proven that node deletion decreases the mean squared residue score of the bicluster [6]. Here the parameter α

determines the rate of node deletion. Usually a higher value of α implies a decrease in multiple node deletion, such that the resulting bicluster size increases. This leads to an increase in execution time, during fine tuning in Step (ii) of the algorithm.

3.4. Evolutionary algorithm

The NSGA-II of Section 2.3.4, in combination with the local search procedure of Section 3.3, is used for generating the set of biclusters. The main steps of the proposed algorithm, repeated over a specified number of generations, are outlined as follows.

(i) Generate a random population of size *P*.
(ii) Delete or add multiple nodes (genes and conditions) from each individual of the population, as discussed in Section 3.3.
(iii) Calculate the multi-objective fitness functions f_1 and f_2, using Eqs. (6)–(7).
(iv) Rank the population using the dominance criteria of Section 2.3.1.
(v) Calculate crowding distance as in Section 2.3.2.
(vi) Perform selection using crowding tournament selection of Section 2.3.3.
(vii) Perform crossover and mutation (as in conventional GA) to generate offspring population of size *P*.
(viii) Combine parent and offspring population.
(ix) Rank the mixed population using dominance criteria and crowding distance, as above.
(x) Replace the parent population by the best $|P|$ members of the combined population, as mentioned in Section 2.3.4.

4. Quantitative evaluation

The bicluster should satisfy two requirements simultaneously. On one hand, the expression levels of each gene within the bicluster should be similar over the range of conditions, i.e., it should have a low mean squared residue score. On the other hand, the bicluster should simultaneously be larger in size. Note that the mean squared residue represents the variance of the selected genes and conditions with respect to the coherence (homogeneity) of the bicluster.

In order to quantify how well the biclusters satisfy these two requirements, we introduce *Coherence Index CI* as a measure of evaluating their goodness. Here *CI* is defined as the ratio of mean squared residue score to the size of the formed bicluster. Let there be *P* biclusters of size $|g_k| \times |c_k|$, $\forall k \in P$ in Eq. (1), with mean squared residue score $\mathscr{G}_k(g_k, c_k)$ from Eqs. (2)–(5). We define

$$\mathscr{G}_k(g_k, c_k) = \frac{1}{|g_k| \times |c_k|} \sum_{i \in g_k, j \in c_k} (e_{ij} - e_{ic_k} - e_{g_k j} + e_{g_k c_k})^2, \quad (12)$$

$$f_k(g_k, c_k) = |g_k| \times |c_k|, \quad (13)$$

$$CI = \min_{k \in P} \frac{\mathscr{G}_k(g_k, c_k)}{f_k(g_k, c_k)}. \quad (14)$$

The *k*th bicluster for $k \in P$ is considered to be good, if it has minimum CI_k among all $j \in P$ and $j \neq k$. A small mean square residue indicates that the corresponding gene set has consistent value over the samples. Note that an increase in bicluster size also leads to a decrease in the value of *CI*.

5. Results

We have implemented the proposed multi-objective biclustering algorithm on microarray data consisting of two benchmark gene expression datasets, viz., *Yeast* and *Human B-cell Lymphoma*. Availability of literature on the performance of related algorithms on these datasets, prompted their selection in this study. As the problem suggests, the size of an extracted bicluster should be as large as possible while satisfying a homogeneity criterion. The threshold δ was selected as 300 for *Yeast* data in Refs. [6,10,15,17,22,28] and 1200 for *Human B-cell Lymphoma* data in Refs. [6,28]. There are no definite guidelines available in literature for the choice of this parameter δ. With a view to providing a fair comparison with existing methods, we have often used the same parameter settings for δ and α; i.e., $\delta = 300$ and 1200 for *Yeast* and *Human B-cell Lymphoma* data, respectively, with $\alpha = 1.2$. We have also made a detailed study on the variation of these parameters.

The crossover and mutation probabilities of 0.75 and 0.03 were selected after several experiments with random seeds. However it was noticed that the crossover and mutation parameters had insignificant effect on the results, as compared to that of δ and α. Due to lack of space, we restrict the population size to 50. Additionally, we investigated the effect of lower δ values in order to demonstrate the biological relevance of the extracted smaller biclusters.

Table 1
Best biclusters for *Yeast* data after 50 generations with $\delta = 300$

α	Bicluster size	No. of ganes	No. of conditions	Mean squared residue	*CI*
1.1	6447	921	7	206.77	0.032
1.2	8832	1104	8	249.61	0.028
1.3	9846	1094	9	263.48	0.027
1.4	11754	1306	9	298.54	0.025
1.5	12483	1387	9	299.88	0.024
1.6	12870	1287	10	299.85	0.023
1.7	12970	1297	10	299.87	0.023
1.8	14828	1348	11	286.27	0.019
1.9	13783	1253	11	299.95	0.022

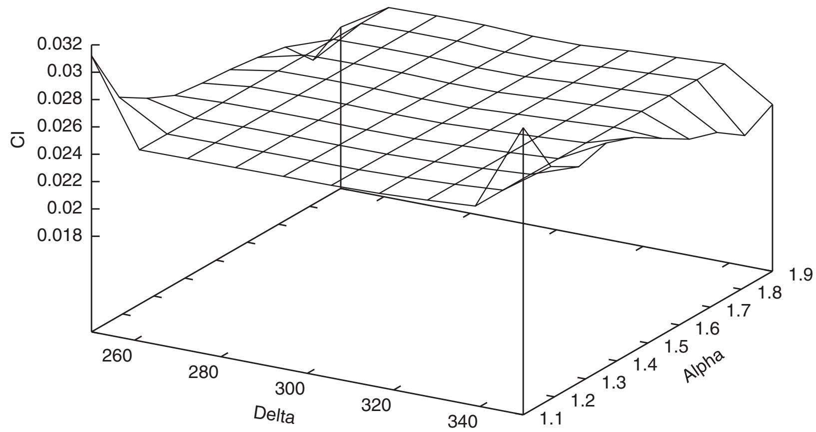

Fig. 3. Plot of *CI* for different choices of α and δ on *Yeast* data.

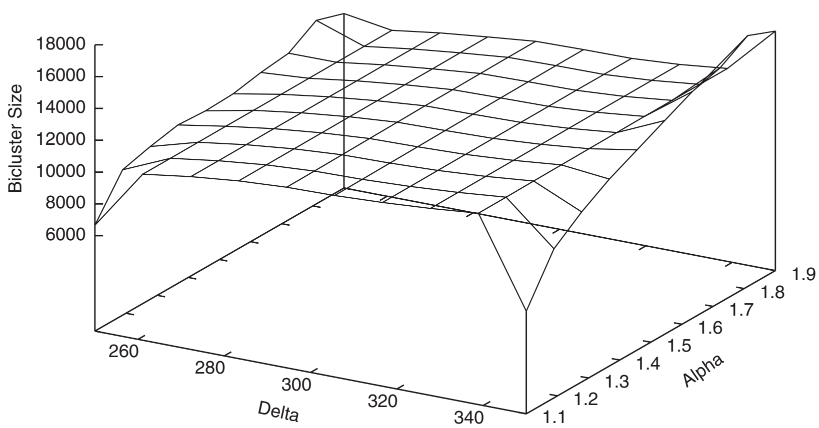

Fig. 4. Plot of bicluster size for different choices of α and δ on *Yeast* data.

5.1. Yeast data

Yeast data[1] is a collection of 2884 genes (attributes) under 17 conditions (time points), having 34 null entries with −1 indicating the missing values. All entries are integers lying in the range of 0–600. The missing values are replaced by random number between 0 and 800, as in [6].

Table 1 summarizes the best biclusters for *Yeast* data after 50 generations, with $\delta = 300$, for different values of α. The population size is chosen to be 50. The largest sized bicluster is found at $\alpha = 1.8$ for each δ, with coherence index *CI* being minimal and indicating the goodness of the discovered partitions. The minimum value of *CI* is 0.019 when $\delta = 300$ and $\alpha = 1.8$, with a corresponding size of 14,828 being the best in the table. As mentioned earlier, a low mean squared residue indicates a high coherence of the discovered biclusters. It may also include some trivial biclusters containing insignificant fluctuations in their expression values, and are not of interest to our study. Hence δ is used as an upper limit of allowable dissimilarity among genes and conditions. However, a higher δ is indicative of diminishing homogeneity.

Figs. 3 and 4 depict the 3D plots of *CI* and bicluster size, against the variations of parameters δ and α. It is observed that with increasing α and δ, the bicluster size also increases while *CI* proportionately decreases.

5.2. Human B-cell Lymphoma data

Human B-cell expression data[2] contain 4026 genes and 96 conditions, with 12.3% missing values, lying in the range of integers −750–650. Here the missing values are replaced by random numbers between −800 and 800, as in Ref. [6] .

Table 2 indicates the best biclusters for *Human B-cell Lymphoma* data, with population size 40 after 50 generations, with $\delta = 1200$, for different α values. The largest bicluster, in this table, is of size 37,560. This is greater than any other method reported in existing literature. The

[1] http://arep.med.harvard.edu

[2] http://arep.med.harvard.edu

Table 2
Best biclusters for *Human B-Cell Lymphoma* data after 50 generations with $\delta = 1200$

α	Bicluster size	No. of genes	No. of conditions	Mean squared residue	*CI*
1.1	25420	820	31	1199.94	0.047
1.2	27200	800	34	1199.33	0.044
1.3	28971	999	29	1199.69	0.041
1.4	31992	1032	31	1199.80	0.037
1.5	33000	1000	33	1199.91	0.036
1.6	33915	969	35	1199.94	0.035
1.7	33896	892	38	1199.31	0.035
1.8	33934	893	38	1199.88	0.035
1.9	37560	939	40	1199.98	0.032

corresponding *CI* value is 0.031. Figs. 5 and 6 demonstrate the 3D variation of *CI* and bicluster size with δ and α. The size of a bicluster increases with α and δ while *CI* proportionately decreases.

Fig. 7 depicts the gene expression profile of this largest bicluster, corresponding to $\delta = 1200$ and $\alpha = 1.9$. The gene expression values in the range −100–100 indicate the highly dense profiles of the coregulated genes having little or no fluctuations under the selected conditions of the bicluster. However, there also exist a few genes (about 3–5) having large expression values. This is perhaps because of the presence of a large number of missing values (about 12.3% of the total) that are replaced by random numbers between −800 and 800, some of which remain in the biclusters without violating the homogeneity constraint. Sometimes this can also occur when a few genes having large variation in their expression values get included while continuing to satisfy the homogeneity constraint δ of the bicluster. Although it is possible to deterministically eliminate these highly fluctuating genes (i.e., those with expression values above 100 or below −100), thereby generating a smaller sized bicluster, yet we choose to retain them as their inclusion does not violate the total allowable dissimilarity δ.

5.3. Comparative study

Biclusters of smaller size were discovered in Ref. [28] for *Yeast* and *Human B-cell Lymphoma* data, using the same values of δ as mentioned earlier. Those extracted from *Yeast* are of sizes (124 × 8), (124 × 9), (19 × 8), (19 × 9), (63 × 9), (23 × 9), (20 × 8) and (20 × 9), with the two entries within the parentheses corresponding to the numbers of genes and conditions, respectively. Note that we generate biclusters of comparable sizes with δ as small as 10–20 in case of *Yeast* data. The authors in Ref. [28] removed genes with missing entries, from *Human B-cell Lymphoma*, to start with a reduced set of 854 genes. Some of the biclusters are of sizes (4 × 15), (5 × 15), (5 × 81), (15 × 81), (4 × 83), (5 × 83), (7 × 83), (26 × 83), (21 × 83), (25 × 83), (72 × 13) and (106 × 13). Sample biclusters, discovered in Ref. [6] for *Human B-cell Lymphoma* data, are of sizes (103 × 25), (127 × 13), (158 × 17), (59 × 18), for $\delta = 1200$. Tables 1 and 2 depict some of the bicluster sizes generated by our proposed algorithm, for these two datasets, corresponding to different α–δ combinations. In all cases our results indicate a better performance in terms of larger bicluster size, while satisfying the homogeneity criterion in terms of δ. The best entry is marked in bold, in both cases.

Table 3 lists a comparative summarization of results on *Yeast* data, involving performance of related algorithms with a threshold $\delta = 300$. The deterministic DBF [17] discovers 100 biclusters, with half of these lying in the size range 2000–3000, and a maximum size of 4000. FLOC [10] uses a probabilistic approach to find biclusters of limited size, that is again dependent on the initial choice of random seeds. FLOC is able to locate large biclusters. However DBF generates a lower mean squared residue, which is indicative of increased similarity between genes in the biclusters. Both these methods report an improvement over the pioneering algorithm of Ref. [6], considering mean squared residue as well as bicluster size. A largest size of 4485 is discovered [6] with $\delta = 300$. Comparing with Table 1, we observe that the bicluster size is always larger in our proposed method.

Single-objective (classical) GA, along with the local search strategy of Section 3.3, was investigated for different sizes of bicluster populations. The fitness function

$$F_t = \gamma_1 f_1 + \gamma_2 f_2 \tag{15}$$

was used, in terms of Eqs. (6)–(7). Additionally, we worked with $0 < \gamma_1, \gamma_2 < 1$ for $0.1 \leqslant \gamma_1 \leqslant 0.9$ subject to $\gamma_1 = 1 - \gamma_2$, and found no significant change in results. Comparison is provided for the same set of parameter initializations, involving multiple (13–15) runs over the same number of generations. The average result is included in Table 3 for *Yeast* data. It is observed that a population size of 50 leads to the generation of a largest bicluster of size 1408. This is less than the bicluster size generated by the proposed multi-objective approach.

Single-objective GA has also been used with local search [22], to generate considerably overlapped biclusters. An initial deterministic selection of biclusters, having similar size, is made for uniform distribution of chromosomes in the population. Thereafter GA is used with minimization of a fitness function, defined as

$$F(g,c) = \begin{cases} \dfrac{1}{f(g,c)} & \text{if } \mathscr{G}(g,c) \leqslant \delta, \\ \dfrac{\mathscr{G}(g,c)}{\delta} & \text{otherwise.} \end{cases} \tag{16}$$

The best bicluster generated from *Yeast* data is 12,350, with an average size of 8600.

The SA based algorithm [23] is able to find significant biclusters of size 18,460 with $\delta = 300$ for *Yeast* data, but it suffers from the "random interference". The results are also data dependent.

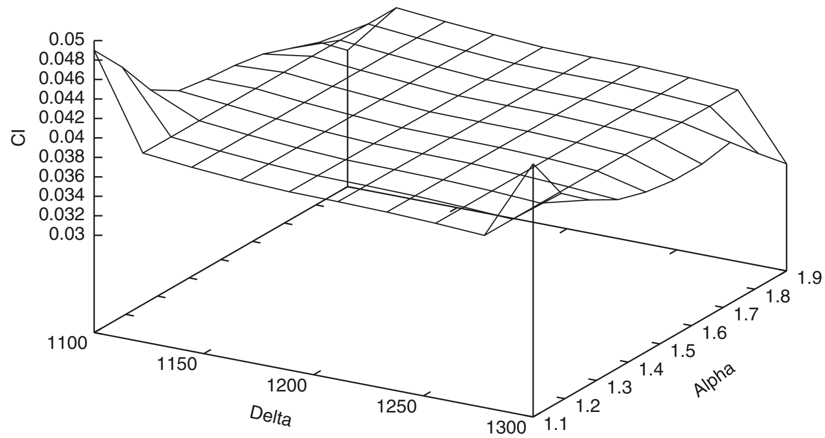

Fig. 5. Plot of *CI* for different choices of α and δ on *Human B-cell Lymphoma* data.

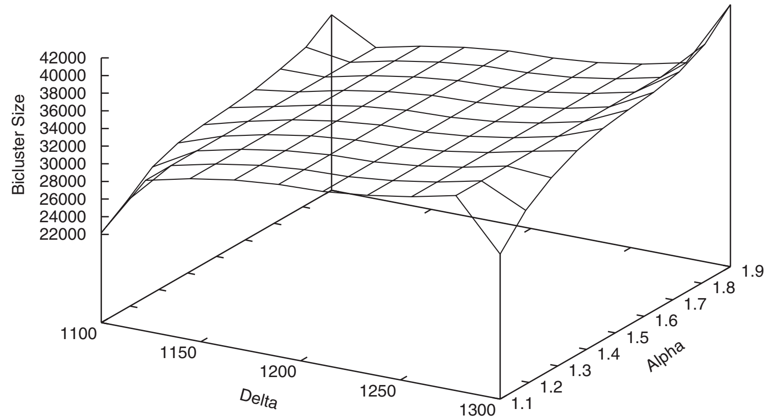

Fig. 6. Plot of bicluster size for different choices of α and δ on *Human B-cell Lymphoma* data.

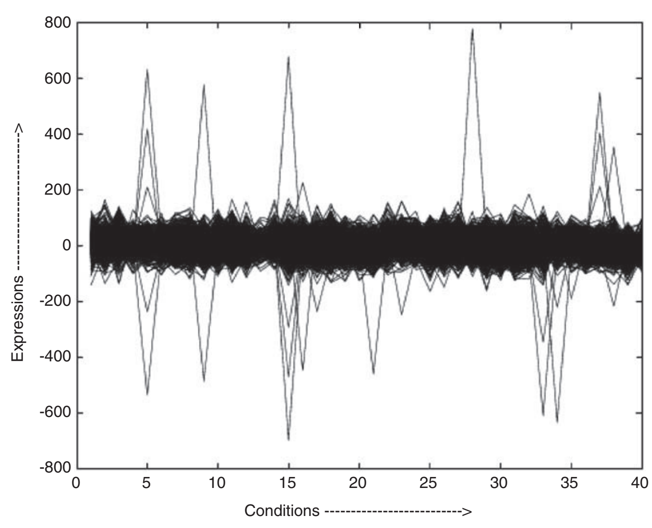

Fig. 7. Gene expression profile of a large bicluster on *Human B-cell Lymphoma* data, with 939 genes and 40 conditions.

Fig. 8 depicts sample gene expression profiles for small biclusters, generated with $\delta = 20$, for *Human B-cell Lymphoma* data. Note that similar sizes of biclusters required a higher δ of 1200, as reported in literature. This implies that MOEA can generate good quality biclusters with comparatively smaller δ values. We also investigated the significance

Table 3
Comparative study on *Yeast* data

Method	Average residue	Average bicluster size	Average no. of genes	Average no. of conditions	Largest bicluster size
FLOC [10]	187.54	1825.78	195	12.8	2000
DBF [17]	114.7	1627.2	188	11	4000
Cheng–Church [6]	204.29	1576.98	167	12	4485
Single-objective GA	52.87	570.86	191.12	5.13	1408
Proposed MOEA	234.87	10301.71	1095.43	9.29	14,828

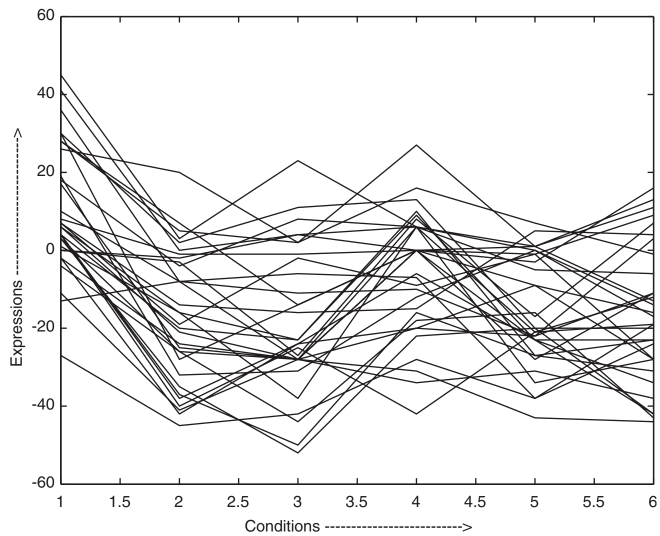

Fig. 8. Small biclusters, of size 32×6, for *Human B-cell Lymphoma* data with $\delta = 20$.

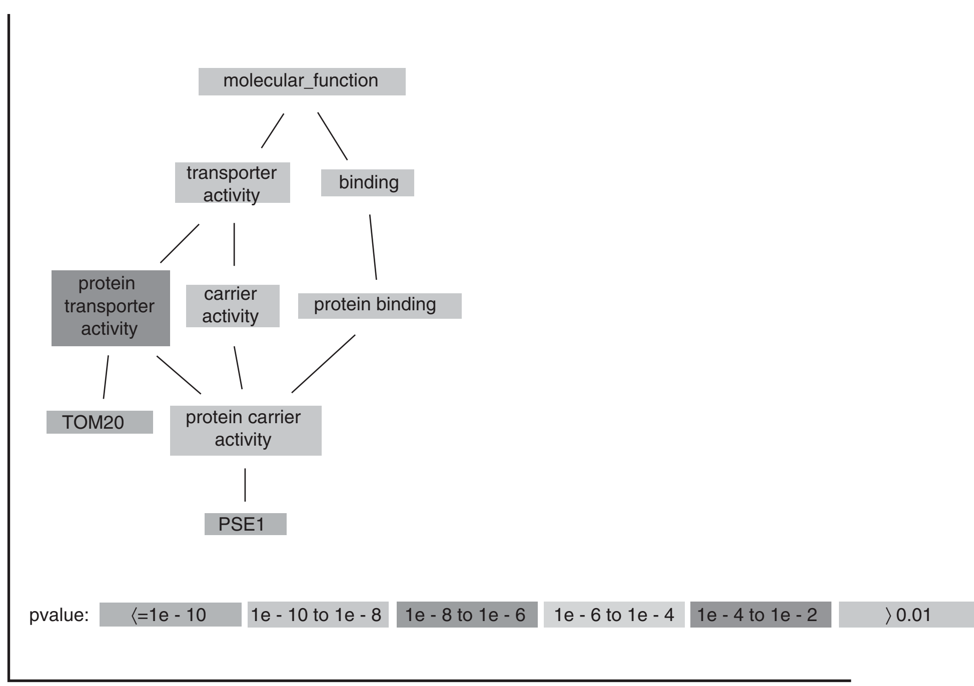

Fig. 9. Sample of 18 genes for *Lymphoma* data, with corresponding GO terms and their parents, for function ontology.

Table 4
Significant shared GO terms (process, function, component) of the selected 12, 18 genes for *Yeast* data

No. of genes	Process	Function	Component
12	tRNA methaylation (2, 0.00024), RNA methaylation (2, 0.00027), biopolymer methaylation (2, 0.0014), tRNA modification (2, 0.0015), cellular process (12, 0.0052), intracellular transport (4, 0.006), establishment of cellular localization (4, 0.0072)	tRNA (guanine) methyltransferase activity (6.07e-05), tRNA methyl-transferase activity (2, 0.00027), RNA methyl transferase activity (2, 0.0006), methyltransferase activity (2, 0.007)	Cytosolic small ribosomal subunit (sensu Eukaryota) (2, 0.0046), Eukaryotic 48S initiation complex (2, 0.004), Eukaryotic 43S preinitiation complex (2, 0.0062), small ribosomal subunit (2, 0.010)
18	Cell organization and biogenesis (9, 0.0048), rRNA processing (3, 0.008), primary transcript processing (2, 0.0120), membrane lipid biosynthesis (2, 0.0130)	Protein transporter activity (2, 0.0067)	Nucleus (4, 0.0053), small nucleolar ribonucleo protein complex (2, 0.0089)

of annotated *Yeast cell-cycle* genes, in case of smaller biclusters generated with $\delta = 20$. This is described in Section 5.4.

Another measure of comparative study is coverage. This signifies the total number of cells in the gene expression array that are *covered* by the biclusters. MOEA covers an average of 51.34% cells in *Yeast* data, with a population size of 50, while an average coverage of 67.30% [6] and 50.99% [22] cells are reported in literature with a population of 100 biclusters. In *Human B-cell Lymphoma* data the MOEA covers an average of 20.96% cells with population size of 40, whereas an average of 36.81% cells are covered in Ref. [6]. It is observed that the process of masking the discovered biclusters by random numbers [6] prohibits the already discovered genes and conditions from future pattern discovery, and leads to discovery of smaller subsets of new genes and conditions. This sort of random interference leads to a higher coverage [6] as compared to MOEA, while simultaneously inhibiting the discovery of larger biclusters.

5.4. Statistical significance

We determined the biological relevance of smaller biclusters for the *Yeast cell-cycle* data, with $\delta = 20$, in terms of the statistically significant GO annotation database.[3] Here genes are assigned to three structured, controlled vocabularies (ontologies) that describe gene products in terms of associated biological processes, components and molecular functions in a species-independent manner.

[3] http://db.yeastgenome.org/cgi-bin/GO/goTermFinder

We have measured the degree of enrichment i.e., *p*-values[4] using a cumulative hypergeometric distribution, that involves the probability of observing the number of genes from a particular GO category (i.e., function, process, component) within each bicluster. The probability p for finding at least k genes, from a particular category within a cluster of size n, is expressed as

$$p = 1 - \sum_{i=0}^{k-1} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}}, \tag{17}$$

where f is the total number of genes within a category and g is the total number of genes within the genome [3]. The p-values are calculated for each functional category in each cluster. Statistical significance is evaluated for the genes in each bicluster by computing p-values, which signify how well they match with the different GO categories. Note that a smaller p-value, close to zero, is indicative of a better match.

Fig. 9 depicts the significant GO terms (or parents of GO terms) for a set of 18 genes along with their p-values, with the significance being indicated in terms of the grayness displayed. It shows the branching of a generalized molecular function into sub-functions like carrier activity, protein binding, protein transporter activity, etc., which are then clustered gene-wise to produce the final result. In other words, it displays the annotated genes in a sample bicluster that is enriched for GO categories.

Table 4 shows the significant shared GO terms (or parent of GO terms) used to describe the set of genes (12 and 18)

[4] The p-value of a statistical significance test represents the probability of obtaining values of the test statistic that are equal to or greater in magnitude than the observed test statistic

in each bicluster, for the process, function and component ontologies. The common terms with increasing order of p-value (i.e., decreasing order of significance) are displayed.

For the first bicluster, the genes (TRM82, TRM112) are particularly involved in the process of tRNA and RNA methaylation, tRNA modification; while genes (GBP2, AGE1, TOM20, VPS21, PRS4, SSK22, NHP10, SOK1, etc.) are involved in cellular process, intracellular transport, etc. The values within parentheses after each GO term in columns 2–4 of the table, such as (2, 0.00024) in the first row, indicate that out of 12 genes in the first cluster two belong to this process and their statistical significance is provided by a p-value of 0.00024. Note that the genes in the cluster share other GO terms also, but with a lower significance (i.e., have higher p-value). From the table we note that each extracted bicluster is distinct along each category. For example, the most significant process in the second bicluster are cell organization and biogenesis (genes LSM2, RRP7, NHP10, TOM20, ECM9, EMG1, SEC65), rRNA processing (genes LSM2, RRP7, EMG1) and membrane lipid biosynthesis (genes VRA7, FEN1).

Looking at the function category of each bicluster, we discover that the most significant terms for the first row are tRNA (guanine) methyltransferage activity (genes TRM82, TRM112), while for the second row it is protein tansporter activity (genes TOM20, PSE1). Finally, the extracted biclusters also differ in terms of their cellular component. The genes (RPS22A, RPS16A) of the first bicluster belong to small ribosomal subunit, while those of the second bicluster (LSM2, RRP7, EMG1, RPC19) belong to the nucleus. This validates the claim that the proposed method is capable of detecting potentially biologically significant biclusters.

6. Conclusions

In this article we have introduced a general multi-objective framework for biclustering gene expression data, while incorporating local search for finer tuning. A qualitative measurement of the formed biclusters, along with a comparative assessment of results, is provided on two benchmark gene expression datasets to demonstrate the effectiveness of the proposed method. Biological validation of the selected genes within the biclusters have been provided by publicly available GO consortium.

Gene expressions provide a fundamental link between genotypes and phenotypes, and play a major role in biological processes and systems including gene regulation, evolution, development and disease mechanism. Biclustering has been mainly applied to gene expressions involving cancerous data, particularly for identification of coregulated genes, gene functional annotation, and sample classification.

Biclustering is typically employed in situations involving (say) the (i) participation of a small set of genes in a cellular process of interest, (ii) study of an interesting cellular process that is active only over a subset of conditions, (iii) participation of a single gene in multiple pathways, which may or may not be coactive under all conditions. Robustness of the algorithms is also desirable, due to the complexity of the gene regulation processes as well as to intelligently handle the level of noise inherent in the actual experiments. Uncovering genetic pathways (or chains of genetic interactions) is equivalent to generating clusters of genes with expression levels that evolve coherently under subsets of conditions, i.e., discovering biclusters where a subset of genes are co-expressed under a subset of conditions. Such pathways can provide clues on (say) genes that contribute towards a disease. This emphasizes the possibilities and challenges posed by biclustering.

However, there also exist other application domains, including information retrieval, text mining, collaborative filtering, target marketing, market research, database research and data mining. The tuning and validation of biclustering methods, in comparison to known biological data, is certainly one of the important open issues for future research.

Acknowledgment

This work was supported by CSIR research Grant no. 22/0346/02/EMR-II.

References

[1] Special issue on bioinformatics, IEEE Comput. 35(7) (2002).
[2] R.B. Altman, S. Raychaudhuri, Whole-genome expression analysis: challenges beyond clustering, Curr. Opinion Struct. Biol. 11 (3) (2001) 340–347.
[3] S. Tavazoie, J.D. Hughes, M.J. Campbell, R.J. Cho, G.M. Church, Systematic determination of genetic network architecture, Nature Genet. 22 (1999) 281–285.
[4] J.T. Tou, R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, London, 1974.
[5] S. Mitra, T. Acharya, Data Mining: Multimedia, Soft Computing, and Bioinformatics, Wiley, New York, 2003.
[6] Y. Cheng, G.M. Church. Biclustering of gene expression data, in: Proceedings of ISMB 2000, 2000, pp. 93–103.
[7] J.A. Hartigan, Direct clustering of a data matrix, J. Am. Stat. Assoc. 67 (337) (1972) 123–129.
[8] S.Y. Kung, M.-W. Mak, I. Tagkopoulos, Multi-metric and multi-substructure biclustering analysis for gene expression data, Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference (CSB'05), 2005.
[9] H. Turner, T. Bailey, W. Krzanowski, Improved biclustering of microarray data demonstrated through systematic performance tests, Comput. Stat. Data Anal. 48 (2) (2005) 235–254.
[10] J. Yang, H. Wang, W. Wang, P. Yu, Enhanced biclustering on expression data, in: Proceedings of the Third IEEE Symposium on Bioinformatics and Bioengineering (BIBE'03), 2003, pp. 1–7.
[11] L. Lazzeroni, A. Owen, Plaid models for gene expression data, Stat. Sin. 12 (2002) 61–86.
[12] A. Tanay, R. Sharan, R. Shamir, Discovering statistically significant biclusters in gene expression data, Bioinformatics 18 (2002) S136–S144.

[13] G. Getz, H. Gal, I. Kela, D.A. Notterman, E. Domany, Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data, Bioinformatics 19 (2003) 1079–1089.
[14] J. Liu, W. Wang, J. Yang, Gene ontology friendly biclustering of expression profiles, in: Proceedings of the 2004 Computational Systems Bioinformatics Conference (CSB 2004), 2004, pp. 436–447.
[15] A H. Tewfik, A.B. Tchagang, Biclustering of DNA microarray data with early pruning, in: Proceedings of ICASSP 2005, 2005, pp. V773–V776.
[16] E. Segal, B. Taskar, A. Gasch, N. Friedman, D. Koller, Rich probabilistic models for gene expression, Bioinformatics 17 (2001) S243–S252.
[17] Z. Zhang, A. Teo, B.C. Ooi, K.-L. Tan, Mining deterministic biclusters in gene expression data, in: Proceedings of the Fourth IEEE Symposium on Bioinformatics and Bioengineering (BIBE'04), 2004, pp. 283–292.
[18] S.C. Madeira, A.L. Oliveira, Biclustering algorithms for biological data analysis: a survey, IEEE Trans. Comput. Biol. Bioinformatics 1 (2004) 24–45.
[19] J. Liu, J. Yang, W. Wang, Biclustering in gene expression data by tendency, in: Proceedings of the 2004 Computational Systems Bioinformatics Conference (CSB 2004), 2004, pp. 1–12.
[20] Y. Zhang, H. Zha, C.H. Chu, A time-series biclustering algorithm for revealing co-regulated genes, in: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05), 2005, pp. 1–6.
[21] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[22] S. Bleuler, A. Prelić, E. Zitzler, An EA framework for biclustering of gene expression data, in: Proceedings of Congress on Evolutionary Computation, 2004, pp. 166–173.
[23] K. Bryan, P. Cunningham, N. Bolshakova, Biclustering of expression data using simulated annealing, in: 18th IEEE Symposium on Computer-Based Medical Systems (CBMS 2005), 2005, pp. 383–388.
[24] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiley, London, 2001.
[25] M. Banerjee, S. Mitra, H. Banka, Evolutionary-rough feature selection in gene expression data, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev., 2006, to appear.
[26] R. Peeters, The maximum edge biclique problem is NP-complete, Discrete Appl. Math. 131 (2003) 651–654.
[27] K. Deb, S. Agarwal, A. Pratap, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2002) 182–197.
[28] H. Cho, I.S. Dhilon, Y. Guan, S. Sra, Minimum sum-squared residue co-clustering of gene expression data, in: Proceedings of Fourth SIAM International Conference on Data Mining, 2004.

About the author—SUSHMITA MITRA is a Professor at the Machine Intelligence Unit, Indian Statistical Institute, Kolkata. From 1992 to 1994 she was in the RWTH, Aachen, Germany, as a DAAD Fellow. She was a Visiting Professor in the Computer Science Departments of the University of Alberta, Edmonton, Canada, in 2004, Meiji University, Japan, in 1999, 2004, 2005, and Aalborg University Esbjerg, Denmark, in 2002, 2003. Dr. Mitra received the National Talent Search Scholarship (1978–1983) from NCERT, India, the IEEE TNN Outstanding Paper Award in 1994 for her pioneering work in neuro-fuzzy computing, and the CIMPA-INRIA-UNESCO Fellowship in 1996.
She is the author of the books "Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing" and "Data Mining: Multimedia, Soft Computing, and Bioinformatics" published by John Wiley. Dr. Mitra has guest edited special issues of journals, and is an Associate Editor of "Neurocomputing". She has more than 100 research publications in referred international journals. According to the science citation index (SCI), two of her papers have been ranked 3rd and 15th in the list of Top-cited papers in Engineering Science from India during 1992–2001. Dr. Mitra is a Senior Member of IEEE. She served in the capacity of Program Chair, Tutorial Chair, and as member of programme committees of many international conferences. Her current research interests include data mining, pattern recognition, soft computing, image processing, and bioinformatics.

About the author—HAIDER BANKA received his M.Sc. and M.Tech. degree in Computer Science from University of Calcutta, India, in 2001 and 2003, respectively. During 2003–2004, he was a lecturer in Engineering College, Durgapur, India. Since 2004 he is a Senior Research Fellow at Machine Intelligence Unit, Indian Statistical Institute, Kolkata. Mr. Banka serves as a reviewer of several international journals. His current research interests include data mining, pattern recognition, soft computing, combinatorial optimization, and bioinformatics.