Abstract—Biclustering in microarray data is used to discover a set of genes expressed similarly in a subset of conditions. Biclustering algorithms require to identify coherent and non-trivial biclusters, i.e., the biclusters should have low mean squared residue and high row variance. This article presents a genetic algorithm based biclustering technique that optimizes a combination of these objectives. A novel encoding strategy is proposed. The performance of the proposed algorithm has been evaluated on two benchmark real life gene expression data sets and compared with some other well-known biclustering techniques.

Index Terms—Biclustering, mean squared residue, row variance, genetic algorithm.

I. INTRODUCTION

Biclustering [1], [2] is an important microarray analysis tool that identifies a set of genes that have similar expression pattern in a subset of experimental conditions. Clustering algorithms [3], [4], [5], which aim to find the clusters of genes over all experimental conditions, may fail to find the genes having similar expression pattern over a subset of conditions. Hence the biclustering algorithms provide better reflection of the biological reality [2], [6].

The biclusters discovered by any algorithm should be highly coherent i.e., low mean squared residue and non-trivial i.e., high row variance. Taking this into account, in this article, a genetic algorithm (GAB) [7] based biclustering technique (GABI) has been developed. In this regard a novel encoding technique is proposed. Moreover, a new quantitative measure to evaluate the goodness of the biclusters is given.

In recent years, several studies have been made by the researchers in the context of biclustering of microarray data. Biclustering was first introduced in [8] in the name of direct clustering. The goal was to discover a set of sub-matrices with zero variance, i.e., with constant values. One of the prior works on biclustering can be found in [1], 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 [1] was based on a greedy search technique guided by a heuristic. Some other recent approaches to biclustering are Flexible Overlapped biclustering (FLOC) [9] and Random Walk biclustering (RWB) [10]. In [11], a coupled two-way clustering (CTWC) method has been proposed that uses hierarchical clustering in both dimensions and combines them to obtain the biclusters. This approach is partially followed in the proposed technique in this article. However, unlike CTWC, the proposed GABI algorithm clusters both the dimensions simultaneously. A bipartite graph based model called Statistical-algorithmic Method for Bicluster Analysis (SAMBA) has been proposed for biclustering in [12]. In [13], a genetic algorithm (GA) based biclustering algorithm has been presented. The main focus of that was to evolve high volume biclusters, and thus the algorithm may fail to discover highly coherent “interesting” biclusters that may have low volume, but high row variance. Moreover, the encoding strategy in the algorithm results in strings having length of summation of total number of genes and conditions. Thus the search space for this technique is too large and searching requires huge computational power. In this article, more focus has been given to discover good quality biclusters that have low mean squared residue and high row variance.

The performance of the proposed algorithm has been demonstrated on two real life benchmark microarray gene expression data sets, viz., Yeast cell cycle and Human large B-cell lymphoma. The results of the developed biclustering technique have also been compared with that of Cheng and Church’s algorithm (CC) [1] and RWB algorithm [10].

II. IMPORTANT TERMS

A microarray data set is considered as a $G \times C$ matrix $M$ representing the expression levels 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., $\text{vol}(I, J) = |I| \times |J|$. The mean squared residue (MSR($I, J$)) of a bicluster $B = (I, J)$ is defined as:

$$\text{MSR}(I, J) = \frac{1}{\text{vol}(I, J)} \sum_{i \in I, j \in J} (a_{ij} - \bar{a}_i - \bar{a}_j + a_{ij})^2, \quad (1)$$

where $a_{ij} = \frac{1}{|I|} \sum_{j \in J} a_{ij}$, $a_{iJ} = \frac{1}{|J|} \sum_{i \in I} a_{ij}$ and $a_{iJ} = \frac{1}{|I| \times |J|} \sum_{i \in I, j \in J} a_{ij}$, i.e., $a_{iJ}$, $a_{iJ}$ and $a_{iJ}$ denote the $i$th
row’s mean, j the column’s mean and the mean of the elements in the bicluster, respectively. The mean squared residue score of a bicluster represents the level of coherence among the elements of the bicluster. Lower residue score means larger coherence and thus better quality of the bicluster. For a given threshold value $\delta \geq 0$, a bicluster $B(I, J)$ is called a $\delta$-bicluster if $MSR(I, J) < \delta$. The row variance $var(I, J)$ of a bicluster $B = (I, J)$ is defined as:

$$var(I, J) = \frac{1}{vol(I, J)} \sum_{i \in I, j \in J} (a_{ij} - \bar{a}_{iJ})^2. \quad (2)$$

The objective is to determine the biclusters with sufficiently high row variance with mean squared residue scores below a certain threshold $\delta$. This is required to escape from the trivial bicluster with almost all elements having constant values.

III. GA based Biclustering

The proposed GA based biclustering (GABI) technique is discussed in detail here.

A. String Representation

Each string has two parts: one for clustering the genes, and another for clustering the conditions. If $M$ and $N$ denote the maximum number of gene clusters and maximum number of condition clusters, respectively, then the length of each string is $M + N$. 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 center of a set of genes, and each $cc_i$, $j = 1 \ldots N$, represents the index of a condition that acts as a cluster center 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, the values of the maximum number of gene clusters ($M$) and the maximum number of condition clusters ($N$) are used as $\lceil \sqrt{G} \rceil$ and $\lceil \sqrt{C} \rceil$, respectively. Here $G$ and $C$ denote the number of genes and the number of conditions in the data set, respectively. The first $M$ positions can have values in the range $\{0, 1, 2, \ldots, G\}$ and the next $N$ positions can have values in the range $\{0, 1, 2, \ldots, C\}$. Hence the gene and condition cluster centers are represented by indices of the genes and conditions, respectively, while a 0 value at any position means absence of any cluster center. Figure 1 illustrates the encoding scheme.

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 $cc_j$. For example, as in Figure 1, a total of $8 \times 5 = 40$ biclusters are formed. Thus each string has the same length, however each of them may encode different number of biclusters.

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 center 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), first all the gene clusters and the condition clusters encoded in it are extracted. Thereafter, the genes and the conditions in the data set are assigned to the respective least distant cluster centers. Subsequently each cluster center (both genes and conditions) are updated by selecting the most centrally located point, from which the summation of distances of other points of that cluster is minimum. Accordingly, the strings are updated. In this article, Euclidean distance measure has been adopted.

The next step is to find all the $\delta$-biclusters (biclusters having mean squared residue at most $\delta$), denoted by some $<gene\ cluster,\ condition\ cluster>$ pair, encoded in the updated string. The fitness function of a bicluster is defined as follows:

$$F(I, J) = \frac{MSR(I, J)}{\delta(1 + var(I, J))}. \quad (3)$$

Here $(I, J)$ denotes a bicluster having set of genes $I$ and set of conditions $J$. The denominator of $F$ is chosen such way to avoid accidental divide-by-zero condition when now variance ($var(I, J)$) becomes 0. Note that $F$ has to be minimized to obtain highly coherent yet “interesting” biclusters. For each encoded $\delta$-bicluster, the fitness function $F$ is computed. The fitness function of a string is then the mean of the fitness values of all encoded $\delta$-biclusters in it.

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

From the final population, all the $\delta$-biclusters are extracted from each string to produce the final biclusters.
D. Genetic Operators

Conventional roulette wheel selection and uniform crossover operation are used in GABI. 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 $\{0,1,2,\ldots,G\}$, 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 $\{0,1,2,\ldots,C\}$, where $C$ is the total number of conditions. Elitism is used to track the best string found so far.

IV. Experiments and Results

Two real-life benchmark microarray data sets are used for experiments. These are described below.

A. Data Sets and Data Preprocessing

1) Yeast Cell Cycle Data: This data set [1] contains 2884 genes and 17 conditions. The rows with missing values are omitted to form a data matrix of size $2882 \times 17$. The data set is publicly available at http://arep.med.harvard.edu/biclustering.

2) Human Large B-cell Lymphoma Data: There are 4026 genes and 96 conditions in this data set [1]. The rows with missing values have been removed to reduce it matrix to a size of $854 \times 96$. This data set is also publicly available at http://arep.med.harvard.edu/biclustering.

B. Input Parameters

The proposed GABI algorithm has been executed for 100 generations with population size 50. The crossover and mutation probabilities are set to 0.8 and 0.1, respectively. For Yeast and Human data sets the $\delta$ values are 300 and 1200, respectively, as selected in [1]. The other algorithms are executed with parameters suggested in respective articles. For each algorithm, 100 best biclusters have been selected based on $BI$ index (described later) from each data set.

C. Validation Index

We have defined a biclustering Index ($BI$) to measure the goodness of the biclusters. Suppose a bicluster has mean squared residue $H$, and row variance $R$. The biclustering index $BI$ for that bicluster is then defined as:

$$BI = \frac{H}{(1 + R)}. \quad (4)$$

As the objective is to minimize $H$ and maximize $R$, hence, lower value of the $BI$ index implies highly coherent and non-trivial biclusters.

D. Results on Yeast Data

For Yeast data, maximum number of gene clusters and maximum number of condition clusters are chosen to be 54 and 5, respectively. Thus the length of string is 59. In Figure 2, 6 biclusters among all biclusters found in a typical run of GABI are shown. Visual inspection from the figure reveals that GABI is able to discover “interesting” biclusters having high row variance. From the figure it is evident that in each of the 6 biclusters, the all genes in the set have similar expression profile (i.e., each bicluster has low MSR). Table I reports the details of these 6 biclusters. Note that each of the biclusters have MSR below 270 and $var$ over 500, thus producing very low values for $BI$ index. Bicluster (c) has the highest row variance (1087.8367), whereas bicluster (d) has the lowest MSR (125.9566). In terms of $BI$ index, bicluster (a) is the most “interesting” bicluster. Although, the biclusters don’t have high volume, they are non-trivial and highly coherent. This is because GABI tries to find highly row-variant biclusters instead of trivial biclusters with large volume.

![Fig. 2. 6 biclusters of Yeast data using GABI](image)

TABLE I

<table>
<thead>
<tr>
<th>bicluster</th>
<th>genes</th>
<th>conditions</th>
<th>$MSR$</th>
<th>$var$</th>
<th>$BI$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>71</td>
<td>7</td>
<td>165.6892</td>
<td>766.8410</td>
<td>0.2158</td>
</tr>
<tr>
<td>(b)</td>
<td>11</td>
<td>8</td>
<td>223.3569</td>
<td>996.4375</td>
<td>0.2239</td>
</tr>
<tr>
<td>(c)</td>
<td>21</td>
<td>8</td>
<td>125.9566</td>
<td>506.4115</td>
<td>0.2482</td>
</tr>
<tr>
<td>(d)</td>
<td>8</td>
<td>7</td>
<td>222.3437</td>
<td>669.3858</td>
<td>0.2492</td>
</tr>
<tr>
<td>(e)</td>
<td>36</td>
<td>10</td>
<td>222.7392</td>
<td>892.7143</td>
<td>0.3317</td>
</tr>
<tr>
<td>(f)</td>
<td>27</td>
<td>8</td>
<td>269.7345</td>
<td>645.2176</td>
<td>0.4174</td>
</tr>
</tbody>
</table>

For the purpose of comparison, Figure 3 shows the plot of average $BI$ index values of 100 best biclusters (in terms of $BI$) over 10 consecutive runs of the algorithms CC, RWB and GABI. It is evident from the figure that GABI has the most stable behaviour; as for most of the biclusters, it provides lower $BI$ index values compared to the other two algorithms.

Table II summarizes the average biclustering results of 100 best biclusters (in terms of $BI$) obtained by different algorithms over 10 consecutive runs. The table reports the average MSR, average row variance, average volume and average $BI$ index values of 100 best biclusters over 10 consecutive runs of each algorithm. The average standard deviation of each measure is reported in brackets. As evident from the table, GABI provides better average MSR and row variance compared to the other methods. However average volume of GABI is lower than that for CC and RWB. This is because GABI concentrates in discovering the “interesting” biclusters, thus sacrificing the volume. For example, GABI provides highest row variance of 3823.46 (not shown in the
Fig. 3. Plot of average BL index values of 100 best biclusters of Yeast data over 10 consecutive runs of algorithms CC, RWB and GABI

table), which is greater than the highest row variance provided by CC (3726.22) and RWB (1044.37). Also, the lowest MSR score for GABI is 77.19, which is better than that of CC (180.67) and RWB (269.75). Moreover, for each measure, the standard deviation of GABI is much lower than that of CC and RWB. This implies that GABI is more stable than its competitors.

E. Results on Human Data

For Human data, the maximum number of gene clusters is chosen as 30 and the maximum number of column clusters is chosen as 10. Hence, string length is 40. For the purpose of illustration, Figure 4 shows 6 biclusters among all biclusters generated in a typical run of GABI for Human data. It is evident from the figure that all of the 6 biclusters are highly coherent and have high value of row variance. The information about these 6 biclusters are given in Table III. Note that each of the biclusters has high row variance (greater than 1800). For example, bicluster (a) has the highest row variance (2315.0660). This bicluster is also the best among the 6 biclusters in terms of BL index score (0.4304).

Fig. 4. 6 biclusters of Human data using GABI

Table IV reports the average MSR, average row variance, average volume and average BL index values for the clustering results of 100 best biclusters (in terms of BL) obtained by all the algorithms over 10 consecutive runs. As evident from the table, GABI provides better average row variance and average MSR compared to the other methods. GABI provides the highest row variance of 5377.51 (not shown in the table), which is greater than the highest row variance provided by CC (5295.42) and RWB (3575.40). Also, the lowest MSR score for GABI is 407.05, which is better than that of CC (612.67) and RWB (1134.26). Average BL index score is also lower for GABI compared to that of CC and RWB. Note that RWB has lower standard deviation for MSR compared to that of GABI. This is probably due to the fact that an iteration of RWB terminates as soon as the MSR becomes lower than \( \delta \). Hence there is not much variation among the MSR values for this algorithm. However, average MSR for RWB is much poorer compared to that of GABI.

Fig. 5. Plot of average BL index values of 100 best biclusters of Human data using over 10 consecutive runs of algorithms CC, RWB and GABI

Table III

<table>
<thead>
<tr>
<th>bicluster</th>
<th>genes</th>
<th>conditions</th>
<th>MSR</th>
<th>var</th>
<th>BL</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>19</td>
<td>59</td>
<td>996.5422</td>
<td>2315.0660</td>
<td>0.4304</td>
</tr>
<tr>
<td>(b)</td>
<td>8</td>
<td>50</td>
<td>1005.4557</td>
<td>1976.0596</td>
<td>0.5086</td>
</tr>
<tr>
<td>(c)</td>
<td>12</td>
<td>50</td>
<td>995.8271</td>
<td>1865.1578</td>
<td>0.5336</td>
</tr>
<tr>
<td>(d)</td>
<td>10</td>
<td>59</td>
<td>1006.7489</td>
<td>1825.6937</td>
<td>0.5511</td>
</tr>
<tr>
<td>(e)</td>
<td>10</td>
<td>35</td>
<td>1054.5312</td>
<td>1895.1600</td>
<td>0.5561</td>
</tr>
<tr>
<td>(f)</td>
<td>23</td>
<td>36</td>
<td>1194.3213</td>
<td>1917.9915</td>
<td>0.6224</td>
</tr>
</tbody>
</table>

10 consecutive runs. Here the biclusters of each algorithm are sorted in ascending order of their BL index scores. From the figure, it is evident that for most of the biclusters, GABI provides smaller BL index scores compared to the other two algorithms.
F. Biological Significance Test

The biological relevance of a bicluster can be verified based on the statistically significant GO annotation database (http://db.yeastgenome.org/cgi-bin/GO/goTermFinder) for associated biological processes, molecular functions and biological components. The degree of functional enrichment ($p$-values) is computed using a cumulative hypergeometric distribution. This measures the probability of finding the number of genes involved in a given GO term (i.e., function, process, component) within a bicluster. From a given GO category, the probability $p$ for getting $k$ or more genes within a cluster of size $n$, can be defined as [14]:

$$p = 1 - \sum_{i=0}^{k-1} \left( \binom{n}{i} \left( \frac{g-f}{g} \right)^i \left( \frac{f}{g} \right)^{n-i} \right),$$

where $f$ and $g$ denote the total number of genes within a category and within the genome, respectively. Statistical significance is evaluated for the genes in a bicluster by computing $p$-values for each GO category. If the majority of genes in a bicluster have the same biological function, then it is unlikely that this takes place by chance and the $p$-value of the category will be close to 0. The biological relevance of an example bicluster of size $31 \times 8$ (shown in Figure 6), obtained using GABI for the Yeast data has been verified at 1% significance level. In terms of different gray levels. The figure shows the branching of a generalized molecular function into sub-functions such as structural constituent of ribosome, nucleic acid binding, etc., which are then grouped gene-wise to produce the ultimate result. The figure suggests that out of 22 significant genes, 19 of them have shared the functional term structural constituent of ribosome.

Table V reports the some of the significant GO terms along with their $p$-values for each GO category. If biological process is considered, the most significant process is macromolecule biosynthetic process ($p$-value = 1.77e-13), and in this process, total 23 genes (FBAl, GPM1, RPL10, RPL15A, RPL23B, RPL38, RPL40B, RPL8B, RPP0, RPS0A, RPS0B, RPS12, RPS18B, RPS1A, RPS1B, RPS21A, RPS25B, RPS29A, RPS31, RPS5, TMA19, YEF3) are involved. The other significant processes are translation, biosynthetic process, cellular protein metabolic process, protein metabolic process and cellular macromolecule metabolic process. Similarly most significant molecular function and cellular component are structural constituent of ribosome ($p$-value = 3.94e-20, 19 genes) and cystolic ribosome ($p$-value = 4.50e-20, 20 genes), respectively. Note that for each GO category, the significant GO terms have very low $p$-values, which indicates that the example bicluster is biologically significant. This signifies that GABI is able to discover biclusters of high biological significance.

V. Conclusions

This article presents a GA based biclustering method that optimizes a combination of mean squared residue and row variance in order to identify interesting biclusters from a gene expression data. In this regard, a novel encoding scheme has been proposed. The performance of the proposed technique (GABI) has been demonstrated on two real life benchmark microarray data sets. Also a comparative study with two other well known biclustering methods has been made to establish its effectiveness. Finally, a biological significance test based on Gene Ontology has been carried out for an example bicluster, to show that the proposed method is able to identify biologically significant biclusters.
Fig. 7. Example bicluster of Yeast data using GABI with corresponding significant GO terms in molecular functions category.

### TABLE V

<table>
<thead>
<tr>
<th>GO Category</th>
<th>Gene Ontology term</th>
<th>Cluster frequency</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>macromolecule biosynthetic process</td>
<td>23 out of 31 genes, 74.2%</td>
<td>1.77e-13</td>
</tr>
<tr>
<td></td>
<td>translation</td>
<td>21 out of 31 genes, 67.7%</td>
<td>4.10e-13</td>
</tr>
<tr>
<td></td>
<td>biosynthetic process</td>
<td>23 out of 31 genes, 74.2%</td>
<td>2.97e-10</td>
</tr>
<tr>
<td></td>
<td>cellular protein metabolic process</td>
<td>21 out of 31 genes, 67.7%</td>
<td>1.32e-06</td>
</tr>
<tr>
<td></td>
<td>protein metabolic process</td>
<td>21 out of 31 genes, 67.7%</td>
<td>2.51e-06</td>
</tr>
<tr>
<td></td>
<td>cellular macromolecule metabolic process</td>
<td>21 out of 31 genes, 67.7%</td>
<td>2.51e-06</td>
</tr>
<tr>
<td>Function</td>
<td>structural constituent of ribosome</td>
<td>19 out of 31 genes, 61.3%</td>
<td>3.94e-20</td>
</tr>
<tr>
<td></td>
<td>structural molecular activity</td>
<td>19 out of 31 genes, 61.3%</td>
<td>1.79e-16</td>
</tr>
<tr>
<td></td>
<td>cytosolic ribosome (sensu Eukaryota)</td>
<td>20 out of 31 genes, 64.5%</td>
<td>4.50e-24</td>
</tr>
<tr>
<td></td>
<td>cytosolic part</td>
<td>20 out of 31 genes, 64.5%</td>
<td>4.40e-24</td>
</tr>
<tr>
<td></td>
<td>ribosome</td>
<td>22 out of 31 genes, 71.0%</td>
<td>4.77e-21</td>
</tr>
<tr>
<td></td>
<td>cytosol</td>
<td>23 out of 31 genes, 74.2%</td>
<td>3.71e-17</td>
</tr>
<tr>
<td></td>
<td>ribonucleoprotein complex</td>
<td>22 out of 31 genes, 71.0%</td>
<td>8.24e-16</td>
</tr>
<tr>
<td></td>
<td>eukaryotic 48S initiation complex</td>
<td>11 out of 31 genes, 35.5%</td>
<td>3.5e-14</td>
</tr>
</tbody>
</table>

### REFERENCES


[10] F. Angiulli, E. Cesario, and C. Pizzuti, “Gene expression biclustering using random walk strategies,” in *Proc. 7th Int. Conf. on Data Warehousing and Knowledge Discovery (DAWAK’05)*, Copenhagen, Denmark, 2005.


