A Binary Particle Swarm Optimization Approach for Gene Expression Biclustering Problem

Bilal Taher*, Muhammad.H Fares† and Saeed Jalili‡

* School of ECE, University of Tehran, Islamic Republic of Iran.
† Advanced Robotics and Intelligent Systems Laboratory, School of ECE, University of Tehran, Iran.
‡ EDST, Lebanese University; Hadath, Lebanon.
§ School of ECE, Tarbiat Modares University, Tehran/Iran.

Abstract
Microarray techniques are widely used in Gene expression analysis. These techniques are based on discovering submatrices of genes that share similar expression patterns across a set of experimental conditions with coherence constraint. Actually, these submatrices are called biclusters and the extraction process is called biclustering. In this paper we present a novel binary particle swarm optimization model for the gene expression biclustering problem. Hence, we apply the binary particle swarm optimization algorithm with a proposed measure, called Discretized Column-based Measure (DCM) as a novel cost function for evaluating biclusters where biological relevance, MSR and bicluster’s size are considered as evaluation metrics for our results.

Results are compared to the existing algorithms and they show the validity of our proposed approach.

Keywords: Gene expression, bicluster, binary particle swarm optimization, mean square residue, biological relevance.

1. Introduction
Gene Expression level is measured using Microarray techniques under different experimental conditions, producing a huge amount of data. Microarray data is widely used in genomic research, and it is usually organized in matrices. For example, in such matrices, the rows are the genes and the columns represent the experimental conditions. Thus, an element of such
expression matrix stands for the expression level of a given gene under a specific experimental condition.

Many approaches have been proposed to extract information from expression matrices. Among these, clustering has been widely used for finding groups of genes that present a similar behavior of the expression level under all the experimental conditions [2]. However, relevant genes are not necessarily related to every condition; in other words, genes might be relevant only for a subset of experimental conditions [3]. Thus, clustering should be performed on two dimensions (genes and conditions) simultaneously instead of one (genes).

Biclustering techniques are introduced to handle such problem [4]; due to the ability of grouping both genes and conditions, they became more popular. The first applied approach for microarray analysis was proposed by Cheng and Church [1]. As in the traditional clustering, the biclustering has the same principle, but for for two main aspects instead of one. For instance, for a microarray of gene expression data, a typical clustering technique would build a set of clusters, where each gene belongs exactly to one single cluster. Nevertheless, many genes may be grouped into diverse clusters (or none of them), depending on their participation in different biological processes within the cell [5]. Another difference is that the biclustering aims to identify genes that are co-expressed under subsets of conditions. This is essential for numerous biological problems, such as the analysis of genes that contribut to certain diseases [3], assigning biological functionalities to genes, or when the conditions of a microarray are diverse.

Biclustering of gene expression is one solution of the NP-hard problems [6]; where simple clustering techniques [7] can’t find significant biclusters. Consequently, many of the proposed techniques are based on optimization procedures as the search heuristic [8]. The development of a suitable heuristic is a critical factor for discovering interesting biclusters in an expression matrix. In order to guide a search heuristic, it is essential to define a measure or cost function for establishing the quality of biclusters. The use of a suitable measure is a key factor, as it determines the effectiveness of the heuristic. Moreover, such a measure can be used for comparing the performances of different search strategies. However, most approaches do not consider the biological relevance of the extracted biclusters in the modeling of the problems.

In this paper, we apply the binary particle swarm optimization algorithm with a proposed measure, called Discretized Column-based Measure (DCM) as a novel cost function for
evaluating biclusters where biological relevance, MSR and bicluster size are considered as evaluation metrics for our results.

The paper is organized as follows: Section 2 presents the basic concepts, and then in Section 3 our approach for solving the biclustering problem is detailed. Experimental results are discussed in Section 4. Finally, Section 5 concludes the paper.

2. Basic concepts

2.1. Binary PSO

PSO was originally developed for continuous-valued search spaces. Kennedy and Eberhard developed the first binary PSO to operate on binary search spaces [9] [10]. For the binary PSO, particles represent positions in binary space. Each element of a particle’s position vector can take the binary value 0 or 1. Changes in a particle’s position take place by flipping a bit from one value to the other. A particle may then be seen to move to near and far corners of the search space by flipping bits. A natural normalization of velocities is obtained by using the sigmoid function (equation 2).

\[
v_{ij}(t) = w \cdot v_{ij}(t-1) + c_1 r_1 (x_{gbest} - x_{ij}(t-1)) + c_2 r_2 (x_{pbest} - x_{ij}(t-1)) \tag{1}
\]

\[
\text{Sig} (v_{ij} (t)) = \frac{1}{1 + e^{-v_{ij}(t)/\beta}} \tag{2}
\]

The position update as follows

\[
x_{ij} (t+1) = \begin{cases} 
1 & \text{if } r(t) < \text{Sig}(v_{ij}(t+1)) \\
0 & \text{otherwise}
\end{cases} \tag{3}
\]

The velocity vector \(v_{ij}(t)\) is real-valued, calculated as given in equation 1, where \(i\) is the index of a particle in the swarm \((i = 1, \ldots, n)\), \(j\) is the index of position in the particle \((j = 1, \ldots, m)\), \(t\) represents the iteration number. The parameter \(w\) represents the inertia weight, \(r_1, r_2\) and \(r (t) \sim U (0, 1)\) uniform distribution where \(c_1\) and \(c_2\) are two constants used to bias the influence between particle’s local bests and global bests. The \(x_{ij}\) is the position of the particle. The binary PSO algorithm is summarized in Algorithm 1. As an initialization step the algorithm
create the particles, spread them randomly in the search space, and assign for each particle its local best. Then the swarm enters a loop and starts moving and selecting its global best and local best, after that it updates its velocity and computes the sigmoid function to update its binary position till the stopping condition comes true.

2.2 Bicluster

Biclusters are represented in different ways [4], where genes can represent either the rows or columns, and different names refer the same expression submatrix. The gene expression dataset is represented by a matrix $M$ with $m$ rows and $n$ columns. $M$ contains the expression level of $m$ genes $G = \{g_1, g_2, \ldots, g_m\}$ over a set of $n$ conditions $C = \{c_1, c_2, \ldots, c_n\}$ during a biological development. Each element $m_{ij}$ represents the expression level of the $i^{th}$ gene at the $j^{th}$ condition. A submatrix $B_{(I,J)}$ of expression matrix $M$ denotes a bicluster, where $I$ and $J$ are subsets of genes set $G$ and conditions set $C$ respectively, i.e., $(I \subseteq G$ and $J \subseteq C)$. Then $B$ can be represented as follows:

$$B = \begin{bmatrix}
    b_{11} & b_{12} & \cdots & b_{1j} \\
    b_{21} & b_{22} & \cdots & b_{2j} \\
    \vdots & \vdots & \ddots & \vdots \\
    b_{i1} & b_{i2} & \cdots & b_{ij}
\end{bmatrix}$$

(4)

3. Proposed Method

This section explains the proposed algorithm that improves the identification of significant biclusters.

3.1. Bicluster encoding

Each bicluster represents a particle of a fixed size binary string composed to genes and conditions. If a gene or condition is included in a bicluster, the corresponding bit is set to 1, otherwise 0. Fig.1 shows an encoding bicluster as described above.
3.2 Mean square residue (MSR)

MSR is a metric that measures the numerical coherence among the genes in a bicluster [1]; it is defined by (5):

$$MSR(B) = \frac{1}{I \times J} \sum_{i=1}^{I} \sum_{j=1}^{J} (b_{ij} - b_{ij} - b_{ij} + b_{ij})^2$$  

Where $b_{ij}$, $b_{ij}$, $b_{ij}$, and $b_{ij}$ represent the element in the $i^{th}$ row (condition) and $j^{th}$ column (gene), the column and row means, and the mean of the bicluster B, respectively. MSR was the first quality metric defined for biclusters of expression data, and it has been included in many approaches from different authors [1], [6], [7], [11-15].

Cheng and Church defined $\delta$-bicluster as follow “$\delta$-bicluster is a bicluster that have an MSR value that is less than a given threshold $\delta$”. Thus their column/row addition/deletion mechanism from a bicluster was based on their definition. In the inequality (6), the left part represent the residue of the row i, and on the right side $\beta \geq 1$ is a penalty factor multiplied by the Bicluster’s MSR. This inequality is used to remove rows satisfying this inequality from the biclusters. Concerning the columns, column with maximum residue value (as calculated in (7)) is removed.

$$\frac{1}{J} \sum_{j=1}^{J} (b_{ij} - b_{ij} - b_{ij} + b_{ij})^2 > \beta MSR(B)$$  

$$\frac{1}{I} \sum_{i=1}^{I} (b_{ij} - b_{ij} - b_{ij} + b_{ij})^2$$  

3.3 Discretized Column-based Measure

In this section we present a measure for establishing the quality of biclusters. The main idea behind the Discretized Column-based Measure is to discretize gene expression data for reducing the infinite set of real gene expression values to an adequate range of discrete values and then apply DCM function to obtain the quality of biclusters. Existing discretization techniques replace each absolute expression value by a symbol from a given alphabet set of two or three symbols.
\{D, U\} and \{D, N, U\}, respectively, where D means down-regulation, N is no-regulation and U means up-regulation. We consider three discretizing techniques as defined in [16].

- **Expression mean and standard deviation** uses an alphabet of three symbols \{D, N, U\} and parameter \(\lambda\) defined by the researcher. Symbol D is used to replace all expression values below the difference between the mean value and the product of \(\lambda\) and the standard deviation. U is used for expression values higher than the sum of the mean value and the product of \(\lambda\) and the standard deviation. N is used for the remaining expression values.

- **Simple threshold technique** discretizes expression values in a binary alphabet \{D, U\} such that if an expression value is higher than a threshold defined by the researchers, it is replaced with U otherwise D.

- **Transitional state discrimination** uses a binary alphabet \{D, U\}. The element \(M'_{i,j}\) of the normalized matrix \(M'\) is set to U if the difference between \(M_{i,j}\) and the \(M'_{i,j}\) exceeds 0, otherwise, it set to the symbol D.

After choosing the Expression mean and standard deviation as discretizing technique where it best fits our problem, we have a discretized expression matrix \(M'\). Here, we define the DCM function to determine the quality of biclusters. Given a bicluster \((I, J)\) of expression matrix \(M\) (G, C), where \(I \subseteq G\) and \(J \subseteq C\). The DCM value of \(B\) is computed by (8).

\[
DCM_B(I,J) = \sum_{j \in J}(1 - \frac{(1+\alpha) \times f_j}{|I|})
\]

(8)

Where \(\alpha\) is a penalty factor \((0<\alpha<1)\), \(|I|\) is the number of genes in bicluster \(B\) and \(f_j\) is computed for every column \(j\) of the bicluster as follows; It counts the frequency of each discrete symbol \{D, U\} (in case of two-letters alphabets). If a symbol has the majority (means has more than \(|I|/2\) occurrences), then \(f_j\) is the number of discretized symbols in column \(j\) that are unequal to the majority symbol. Otherwise, if none of the symbols have majority, \(f_j\) is set to \(|I|\). Also, if the majority symbol is N (in case of three-letters alphabet), \(f_j\) is set to \(|I|/2\). Considering (Eq. 1), we see that in the best case when all discrete symbols of column \(j\) have same value from \{D, U\}, \(f_j\) is 0 and as a result \((I, J)\) is equal to \(|J|\). On the other hand, in the worst case, if no discrete symbols have majority, \(DCMB(I,J)\) is equal to \(-\alpha\times|J|\). So, \(-\alpha\times|J| < DCMB(I,J) \leq |J|\).

### 3.4 The architecture of the proposed method
The proposed method consists of three steps: 1. Preprocessing and Discretization of input dataset, 2. running the binary PSO algorithm with DCM as fitness function and a local search to help in converging toward the best solution. In the final step, (4) is applied on biclusters till their Mean square residue value becomes less than a given threshold $\delta$, note that a bicluster is said to be $\delta$-bicluster if it’s mean square residue is less than a given threshold $\delta$. and as a final output we will get $\delta$-biclusters. Fig.2 illustrates the global view of the proposed algorithm.

![Diagram of the proposed method](image)

**Figure 2.** The architecture of the proposed method

### 3.5 Local Search

The local search function aims to optimize the position of each particle in every iteration by finding a better neighborhood which is 5% different from the main particle.

### 3.6 DCM-$\delta$ Algorithm

The proposed Algorithm (Algorithm 2) takes as an input the discretized expression matrix, the threshold $\delta$, and the swarm list $S$ which contain $k$ particle. Each particle is represented as shown in figure 2 where it is initialized using a random function. At the end of the BPSO we get a list of biclusters.
Our proposed DCM-δ algorithm is shown in Fig.3:

| Input: Discretized Expression Matrix DEM; threshold δ; k particle; |
| Output: List of δ-biclusters S |
| 1. Create the initial population S with k particles; |
| 2. Initialize the memory of each particle: |
| i. Initialize the velocity matrices to 0; |
| ii. Initialize the position of each particle using Eq.3; |
| iii. Assign particles local best and global best; |
| 3. Repeat till stable state is reached or till maximum iteration: |
| i. Evaluate each particle using Eq.5 |
| ii. Compute the velocity matrices using Eq.1 |
| iii. Compute the new position for the particles using Eq.3 |
| iv. Local search |
| v. Actualize the memory of each particle |
| vi. Assign the local best |
| vii. Assign the global best |
| viii. Update list S |

/* Fulfillment of δ-biclusters based on Eq.4 */
4. For each particle i in S do
   i. \( B = S(i) \)
   ii. \( B'_{δ} = \text{node deletion phase}(B, δ) \)
   iii. \( B''_{δ} = \text{node addition phase}(B'_{δ}, δ) \)
   iv. \( S(i) = B''_{δ} \)
5. End for
6. Return S

Figure 3: The proposed algorithm for DCM-δ

4. Evaluation of the proposed method

DCM-δ based BPSO algorithm is implemented in JAVA where we used jMetal framework; an object-oriented Java-based framework for multi-objective optimization with metaheuristic techniques. The jMetal project is available on http://jmetal.sourceforge.net/. We run our
algorithm on a PC with 3.4 GHz core i7, 16 GB memory, and Windows 7 ultimate 64 bit as an operating system.

**4.1 Dataset**

For our experiment we use 3 well known datasets ,the first one is the Yeast cell cycle expression dataset [1] with 2884 rows and 17 columns .The second dataset is the Saccharomyces cerevisiae dataset [17] with 2993 rows and 173 columns, and the third one is the Human dataset [1] with 4026 rows and 96 columns. The values of δ used in DCM- δ for the human and the yeast cell cycle datasets are taken from [1], while for the other dataset, it is established using the procedure suggested in [1]. The values of δ of each dataset is shown in table 1.

| Dataset                                      | δ for DCM-δ |
|----------------------------------------------|-------------|
| Yeast Cell Cycle [4]                         | 300         |
| Human B-cell [4]                             | 1200        |
| Saccharomyces cerevisiae [13]                | 0.03        |

**4.2 Evaluation metrics**

In this section we study the performance of DCM-δ algorithm and compare it with different methods: CC [1], BicFinder [12], MODPSFLB [13], CMOPSOB [14], HMOBI [17], OPSM [18] and NSGA2B [19]. Concerning these methods we just report their published results. Mainly the MSR value and the average biclusters size are used to be evaluation metric for such problem (biclustering of gene expression), where researchers aim to resolve the existing tradeoff between these two metrics by finding δ-biclusters of large size. On the other hand many methods do not take into consideration the biological relevance of the extracted biclusters as an evaluation metric, where its presence is important to find the rate of genes that share significant GO terms in each bicluster using the gene ontology annotation, and makes sense in finding out similarities between genes included in each bicluster from the biological point of view.

**4.3 Experimental results**

Tables 2 and 3 present the quality of the extracted biclusters from Yeast Cell Cycle dataset and Human B-cell expression dataset respectively. We reported the average size of the founded biclusters, and average mean squared residue. After experimental tests we restricted the
population to 45 particles (bicluster), with maximum 140 iteration or when the termination criterion is satisfied i.e. the solution converged to a stable state.

As we can notice in tables 2 and 3, all the methods have the $\delta$-biclusters constrain fulfilled, on the other hand the larger the average bicluster size is, the better the solution is. For example the CMOPSOB [14] algorithm is of an average MSR (208.86) and average size (11085.4) which is a good result however the MODPSFLB [13] algorithm is better, although it have a worst MSR (212.8) than the CMOPSOB [14] algorithm yet it fulfilled the $\delta$-biclusters condition (which means its biclusters are of good quality) and at the same time its biclusters are larger than that of the CMOPSOB [14] algorithm. Our results on the yeast cycle dataset are better than the others since our average size of biclusters, 14724 is the largest among the existing ones and on the other hand our average MSR is 251.12 (i.e. biclusters are $\delta$-biclusters) which is a significant result with respect to the obtained average biclusters size. Similarly for the human B-cell dataset the obtained biclusters satisfy the constrain of being smaller than threshold $\delta$ and at the same time the average size is the largest among the different algorithms.

Table 2. the comparison of the proposed method and other methods on yeast cell cycle dataset where $\delta=300$

| Algorithm            | Avg. MSR | Avg. size  |
|----------------------|----------|------------|
| CMOPSOB [14]         | 208.86   | 11085.4    |
| MODPSFLB [13]        | 212.8    | 11220.7    |
| NSGA2B [19]          | 234.87   | 10301.7    |
| HMOBI [17]           | 299.6    | 7665       |
| DCM-$\delta$         | 251.12   | 14724      |
Table 3. The comparison of the proposed method and other methods on human B-cells dataset where $\delta=1200$

| Algorithm      | Avg. MSR | Avg. size |
|----------------|----------|-----------|
| CMOPSOB [14]   | 921.7    | 36401     |
| MODPSFLB [13]  | **904.9**| 35602     |
| NSGA2B [19]    | 987.6    | 33464     |
| HMOBI [17]     | 1199.9   | 47442     |
| DCM-\(\delta\) | 1191     | **128368**|

4.4 Biological relevance

GOTermFinder[^1] and FuncAssociate [20] are web tools that can be used over Yeast Cell Cycle and Saccharomyces cerevisiae datasets to extract the corresponding GO terms of the genes of the above datasets, so we use these tools to compute the biological relevance of the extracted biclusters. Here we introduce some related definitions:

- **Response to stimulus**, any process that results in a change in state or activity of a cell or an organism (in terms of movement, secretion, enzyme production, gene expression, etc.) as a result of a stimulus. The process begins with detection of the stimulus and ends with a change in state or activity or the cell or organism. *Source: GOC: ai, GOC: bf http://amigo.geneontology.org/amigo/term/GO:0050896*

- **Catalytic activity**, catalysis of a biochemical reaction at physiological temperatures. In biologically catalyzed reactions, the reactants are known as substrates, and the catalysts are naturally occurring macromolecular substances known as enzymes. Enzymes possess specific binding sites for substrates, and are usually composed wholly or largely of

[^1]: [http://go.princeton.edu/cgi-bin/GOTermFinder](http://go.princeton.edu/cgi-bin/GOTermFinder)
protein, but RNA that has catalytic activity (ribozyme) is often also regarded as enzymatic. 

Source: ISBN: 0198506732, GOC: vw http://amigo.geneontology.org/amigo/term/GO:00038

- **Ion binding**, charged atoms or groups of atoms interacting selectively and non-covalently with ions. 
  
  Source: GOC:jl http://amigo.geneontology.org/amigo/term/GO:0043167

- **Cellular process**, any process that is carried out at the cellular level, but not necessarily restricted to a single cell. For example, cell communication occurs among more than one cell, but occurs at the cellular level. 
  
  Source: GOC:go_curators, GOC:isa_complete http://amigo.geneontology.org/amigo/term/GO:0009987

- **Intracellular**, The living contents of a cell; the matter contained within (but not including) the plasma membrane, usually taken to exclude large vacuoles and masses of secretory or ingested material. In eukaryotes it includes the nucleus and cytoplasm. 
  
  Source: ISBN: 0198506732 http://amigo.geneontology.org/amigo/term/GO:0005622

- **Chromosomal part**, Any constituent part of a chromosome, a structure composed of a very long molecule of DNA and associated proteins (e.g. histones) that carries hereditary information. 
  
  Source: GOC: jl http://amigo.geneontology.org/amigo/term/GO:0044427

Table 4 with respect to the above six definitions presents the performance of the proposed algorithm in comparison with [17]. We run GoTermFinder on each bicluster and as a result GoTermFinder showed that our algorithm is able to find biological interrelated biclusters with high efficiency. As we can see in table 4 after calculating the average GO terms of all biclusters and calculating the variance between the results, 90.3% of genes presented in each bicluster with variance ± 0.235 were implicated in cellular process and the statistical significance is provided by a p-value 8.64e-09, a result that is more significant than the one represented in [17] which was 82.2% with p-value 8.39e-07. Also biclusters were highly related to various biological aspects like catalytic activity (51.2%, 8.32e-26), intracellular (94.5%, 2.42e-51), response to
stimulus (26%, 4.41e-05), ion binding (17.5%, 3.70e-06) and chromosomal part (9%, 1.80e-05) with higher percentages than algorithm [17].

By referring to the previous definitions we can justify the low percentages in the “response to stimulus”, “ion binding”, and “chromosomal part” by the absence of various factors such as time series, and physical interaction between gene products, that are implicated in the relationship between the genes of these processes i.e. microarray technology only treat works the expression of these genes which is not sufficient to predict all the interrelated groups in this gene set. The percentages here show only the sets of genes that are related according to the expression data. Experiments other than microarray are needed to detect the complete sets of gene.

Table 4 biological relevace (significant shared GO terms) of the proposed method in comparison with the HOMBI method [17] on the yeast cell cycle dataset

| Algorithm | Bic’s size | Process function | Component function |
|-----------|------------|------------------|--------------------|
| HOMBI[17] | 709 x 9    | Cellular process (82.2%, 8.39e-07) | Catalytic activity (41.1%, 0.00049) | Intracellular (84.5%, 8.57e-10) |
|           | 1746*13    | Response to stimulus (20.0%, 0.00024) | Ion binding (25.9%, 0.00137) | Chromosomal part (7%, 0.00172) |
| DCM-5     | 1310*11    | Cellular process (90.3% ± 0.235, 8.64e-09) | Catalytic activity (51.2% ± 0.966, 8.32e-26) | Intracellular (94.5% ± 0.1512, 4.5e-51) |
|           |            | Response to stimulus (25% ± 0.1068, 4.41e-05) | Ion binding (17.5% ± 0.4913, 7.0e-06) | Chromosomal part (9% ± 0.186, 1.80e-05) |

We examine the biological interest of our biclusters extracted from Saccharomyces cerevisiae dataset using the web tool FuncAssociate [20]. Actually, for a given genes list, FuncAssociate computes an adjusted significance score (p-value) which indicates how well the genes match with the different GO aspects. The results are compared to HMOBI[17], BicFinder [12], OPSM [18] and CC [1] heuristics.

Table 5 presents the average rows number, average columns number and the rate of significant biclusters with a p-value ≤0.1%. Results show that the proposed algorithm can extract significant biclusters that is at the same time larger than the HMOBI [17] and BicFinder [12] biclusters. OPSM [18] extracts small biclusters of a good quality, but for the CC [1] biclusters, relevance is very poor which can be explained by its lack of robustness against noise. In fact, after the extraction of a bicluster, CC algorithm replaces the elements of this bicluster in the original data.
matrix by random values. Hence, the next extracted biclusters may be influenced by the included random values.

Table 5. Biological relevance (Significant shared GO terms) of the proposed method in comparison with other methods applied on the Saccharomyces cerevisiae data

| Algorithm   | Avg. Genes | Avg. Conditions | Avg. size | Rate |
|-------------|------------|-----------------|-----------|------|
| HOMBI [17]  | 1714.64    | 109.305         | 187418    | 100% |
| BicFinder[12] | 421       | 173             | 21471     | 100% |
| OPSM[18]    | 10         | 51.69           | 516.9     | 87%  |
| CC[1]       | 511.6      | 59.9            | 30660     | 24%  |
| DCM-δ       | 1793.88    | 155.77          | 279432    | 100% |

5. Conclusion

In this paper we introduced a novel binary particle swarm optimization algorithm that solves the gene expression biclustering problem. The idea behind our algorithm is to discretize datasets to make use of the regulation state of the genes to find biclusters of good quality. Using the literature real datasets, the performance of DCM-δ based BPSO algorithm was proved to be significant assessed in terms of Gene Ontology (GO) and in the terms of mean square residue with respect to the average bicluster size. The comparative study with some other methods shows that DCM-δ based BPSO algorithm presents better results. And as a future work we can apply this procedure in a multi objective environment where lot existing fitness function can be adopted to work aside with our proposed method.

References

[1] Y. Cheng, G.M. Church, Biclustering of expression data, In: Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology, 2000, pp. 93–103.

[2] M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns. In: proceedings of the Natl. Acad. Sci. 1998 pp 14863–14868.
[3] H. Wang, W. Wang, J. Yang, P.S. Yu, Clustering by pattern similarity in large data sets, In: proceeding of ACM SIGMOD International Conference on Management of Data 2002 pp. 394–405.

[4] J. Hartigan, Direct clustering of a data matrix, J. Am. Stat. Assoc. 67 (337) : 1972 pp 123–129.

[5] Gasch, A.P. and Eisen, M.B. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. Genome biology 3(11): 2002, pp.1.

[6] A. Tanay, R. Sharan, R. Shamir, Discovering statistically significant biclusters in gene expression data, Bioinformatics 18:.2002 pp 136–144.

[7] F. Divina, J.S. Aguilar-Ruiz, Biclustering of expression data with evolutionary computation, IEEE Trans. Knowl. Data Eng. 18 (5) 2006 pp 590–602.

[8] Pontes, B., Giráldez, R. and Aguilar-Ruiz, J.S. Biclustering on expression data: A review. Journal of biomedical informatics, 57, 2015, pp.163-180

[9] Kennedy and R.C. Eberhard. A Discrete Binary Version of the Particle Swarm Algorithm. In Proceedings of the World Multi conference on Systemic, Cybernetics and Informatics: 1997 pp 4104–4109.

[10] J. Kennedy, R.C. Eberhart, and Y. Shi. Swarm Intelligence. Morgan Kaufmann, 2001

[11] B. Pontes, R. Giráldez, J.S. Aguilar-Ruiz, Quality measures for gene expression biclusters, PloS One 10 (3): 2015. pp. e0115497–e0115497.

[12] A. Wassim, E. Mourad, H. Jin-Kao, Bicfinder: a biclustering algorithm for microarray data analysis, Knowl. Inf. Syst. 30 (2) 2012 pp 341–358.

[13] J. Liu, Z. Li, X. Hu, Y. Chen, F. Liu, Multi-objective dynamic population shuffled frog-leaping biclustering of microarray data, BMC Genomics 13 (Suppl. 3) 2012 S6.

[14] J. Liu, Y. Chen, Dynamic biclustering of microarray data with MOPSO, In: proceeding in IEEE International Conference on Granular Computing, IEEE Computer Society, 2010, pp. 330–334.

[15] Khedidja Seridi, Laetitia Jourdan, El-Ghazali Talbi, Using multi-objective optimization for biclustering microarray data. Applied Soft Computing 33: 2015 pp 239–249

[16] Gonçalves, J.P., S.C. Madeira, and A.L. Oliveira, BiGGEsTS: integrated environment for biclustering analysis of time series gene expression data. BMC research notes, 2(1) 2009. pp. 124
[17] Seridi, K. Jourdan, and Talbi. Using multi-objective optimization for biclustering microarray data, Applied Soft Computing: 2015, pp.239-249.

[18] Ben-Dor, A., Chor, B., Karp, R. and Yakhini, Z.,. Discovering local structure in gene expression data: the order-preserving submatrix problem. Journal of computational biology, 10(3-4), 2003, pp.373-384.

[19] S. Mitra and H. Banka, “Multi-objective evolutionary biclustering of gene expression data,” Pattern Recognition, 39(12), 2006, pp. 2464-2477.

[20] G.F. Berriz, Characterizing gene sets with funcassociate, Bioinformatics 19 2003 pp 2502–2504.

[21] A. Ben-Dor, B. Chor, R.M. Karp, Z. Yakhini, Discovering local structure in gene expression data: the order-preserving submatrix problem, in: RECOMB, 2002, pp. 49–57