A Multiobjective Genetic Algorithm for the Biclustering of Gene Expression Data

Jorge E. Luna-Taylor[1], Carlos A. Brizuela[2]
[1]*Departamento de Sistemas y Computación, Instituto Tecnológico de La Paz, Boulevard Forjadores No. 4720, La Paz, B.C.S., México; E-mail: jelt0024@hotmail.com, web: http://www.itlp.edu.mx*
[2]*Departamento de Ciencias de la Computación, Centro de Investigación Científica y de Educación Superior de Ensenada, Carretera Ensenada- Tijuana 3918, Ensenada, B.C.N., México, E-mail: jelt0024@hotmail.com, web: http://www.itlp.edu.mx; cbrizuel@gmail.com, web: http://www.cicese.mx*

Abstract

In recent years the development of new technologies for the design of DNA microarrays, has generated a large volume of biological data, which requires the development of parallel computational methods for their functional interpretation. On these sets of data, biclusters construction algorithms attempt to identify gene associations and experimental conditions, where genes exhibit a high correlation to each given condition. In this paper, we introduce a new multi-objective genetic algorithm, that unlike other evolutionary proposals, does not require a local search for the identification of optimum biclusters. The proposed algorithm is simpler and had better performance than the ones found in the current literature for two real gene expression data.

Keywords: biclustering, gene expression, multi-objective genetic algorithm, microarray DNA.

1. Introduction

The increased use of microarray technology has generated a large volume of biological data, which necessitates the development of parallel computational methods for their functional interpretation. To address this challenge it is necessary to apply data mining techniques. Among these techniques, in regards to database gene expression, clustering has become one of the most used approaches as a first step in the work of discovering new knowledge. However, the results of clustering methods applied to genes have been limited. This limitation causes difficulty in analyzing the expression of genes for a given set of experimental conditions. To overcome this limitation various algorithms have been proposed to cluster genes and conditions simultaneously. These algorithms are called bicluster algorithms and have the aim to identify groups of genes, given a set of experimental conditions, where the genes exhibit a high correlation across a set of given conditions.

The search for biclusters in gene expression arrays is a very attractive computational challenge. The work of Cheng and Church [1] is of significance since it introduced the concept of bicluster applied to the analysis of gene expression for the first time, and proposed an original algorithm for its construction. Despite some limitations presented in this algorithm (as discussed by Rodriguez et al. [2] and by Aguilar [3]), it has been used as a basis for evaluating and comparing the performance of a wide variety of more recent and elaborated algorithms.

Madeira and Oliveira [4] presented a classification of biclustering methods mainly based on two aspects: i) the type of biclusters that the algorithms are able to find, and ii) the computational technique used. There are algorithms that seek biclusters with constant values, e.g. the mClustering [5], based on the divide and conquer approach, and the DCC [6] based on a combination of clustering of rows and columns. Other methods identify biclusters with columns or rows with constant values, such as the CTWC [7], the δ-Patterns [8] based on a greedy approach, and Gibbs [9]. Some methods like δ- biclusters [1] and FLOC [10, 11], use greedy approaches, the pClusters [12] uses exhaustive search, Plaid Models [13] and PRM [14, 15], are based on the identification of the distribution parameters. There are also methods that seek biclusters with patterns of coherent evolution (OPSMs [16] and xMotifs [17]), using a greedy search, and SAMBA [18] and OP-Clusters [19], which perform exhaustive search.

Rodriguez et al. [2] add to this classification methods that use stochastic search. In this branch algorithms such as the SEBI [20] and Simulated Annealing [21] are included.

Despite the existence of a large number of biclustering algorithms, there are still many significant challenges to overcome:

- The scarce information available to define the type of specific biclusters to search.
- The amount of noise in the data matrices.
- The computation time due to the complex calculations often required.
- The absence of data in the input matrices.
- The existence of user parameters that strongly influence the final results.
- The lack of assessment methods for the generated results.
- The multi-objective nature of the problem, since the MSR and the bicluster size, must be optimized at the same time.

In this paper, we introduce an evolutionary algorithm for the biclustering problem. The algorithm considers biclusters themselves as individuals in a population to evolve. The objective is to minimize the MSR value of biclusters, while simultaneously maximizing its size. Experiments are performed on two reference sets data (Yeast Saccharomyces cerevisiae and Human Lymphoma B-cells). The problem to solve is formally defined next.

2. Biclustering analysis of gene expresion

Cheng and Church [1] introduced the concept of bicluster within the context of gene expression data analysis. A bicluster is a subset

of genes and a subset of conditions with a high level of similarity. The similarity is considered as a consistency measure between genes and conditions in the bicluster.

Within this context, we can define biclustering as the process of grouping genes and conditions simultaneously, searching for biclusters of maximum size and maximum similarity within a data matrix of gene expression.

Madeira and Oliveira [4] present a formal approach to the problem of biclustering. As input data having a matrix of n by m, where each element a_{ij} is usually a real value. In the case of gene expression arrays, a_{ij} represents the level of expression of gene i under condition j. More generally, one considers the data matrix A with a set X of rows and a set of columns Y, where the element aij corresponds to a value representing the relationship between the row i and column j.

The matrix A with n rows and m columns is defined by its set of rows, $X = \{x_1,...,x_n\}$ and its set of columns, $Y = \{y_1,...,y_m\}$. (X, Y) is used to denote the matrix A. If I $\subseteq$ X and J $\subseteq$ Y are subsets of rows and columns of A, respectively, then $A_{IJ} = (I, J)$ which denotes the submatrix A_{IJ} of A containing only the elements a_{ij} belonging to the submatrix with the set of rows I and the column set J.

Given the matrix A, a cluster of rows is a subset of rows that have a similar behavior through the set of all columns. This means that a cluster of rows $A_{IY} = (I, Y)$ is a subset of rows defined by the set of all columns Y, where $I = \{i_1,...,i_k\}$ is a subset of rows $I \subseteq X$ and $k \leq n$. A cluster of rows (I, Y), can thus be defined as a submatrix k by m of the data matrix A. Similarly, a cluster of columns is a subset of columns which have a similar behavior across the set of all rows. A cluster $A_{XJ} = (X, J)$ is a subset of columns defined on the set of all rows of X, where $J = \{j_1,...,j_s\}$ is a subset of columns ($J \subseteq Y$ and $s \leq m$). A cluster of columns $A_{XJ} = (X, J)$ can be defined as a submatrix of n by s of the data matrix A.

A bicluster is a subset of rows that have a similar behavior through a subset of columns, and vice versa. The bicluster $A_{IJ} = (I, J)$ is a subset of rows and a subset of columns of Y, where $I = \{i_1,...,i_k\}$ is a subset of rows ($I \subseteq X$ and $k \leq n$), and $J = \{j_1,...,j_s\}$ is a subset of columns ($J \subseteq Y$ and $s \leq m$). A bicluster (I, J) can be defined as a submatrix of k by s of the data matrix A.

The specific problem addressed by the biclustering algorithms is defined as: given a data matrix A it is required to identify a set of biclusters $B_k = (I_k, J_k)$ such that each bicluster B_k satisfies some property of homogeneity. The exact features of homogeneity of biclusters vary according to the statement of the problem.

Although the complexity of the biclustering problem depends on the exact formulation of the problem, and specifically the function used to evaluate the quality of a bicluster, most variants of this problem are NP-hard.

3. Related work

Recently there have been several algorithms based on a variety of techniques to find biclusters, for example, BBC [22], Reactive GRASP [23], RAP [24], GS Binary PSO [25] and TreeBic [26], among others.

In general, it is difficult to evaluate and compare biclustering methods, since the results obtained strongly depend on the scenario under consideration. Prelic et al. [27]

present an evaluation and comparison of five outstanding methods. The evaluated methods were: CC [1], Samba [18], OPSM [16], ISA [28, 29] and xMotif [17]. To evaluate the methods both artificial and real data sets were used. Tested with artificial data biclusters with constant and additive values. They also were tested with systematic increase in noise, and with increasing overlap between the created biclusters. As for the real data, biological information used GO annotations [30, 31], maps of metabolic pathways [31], and information on protein-protein interaction [32, 31]. In general, the methods ISA, Samba and OPSM perform well. While some methods perform better under certain scenarios, show lower performance in others.

Mitra and Banka [33] introduced a multiobjective evolutionary algorithm (MOEA) with the addition of local search. The objective is to find large size biclusters, with MSR values below a predefined threshold. Their method was evaluated using two sets of gene expression data referenced in the literature: yeast and Human B Cell Lymphoma. The yeast data they used, is a collection of 2884 genes under 17 conditions, with 34 null entries with value -1, indicating loss of data. Expression data of Human B cells [37], containing 4026 genes under 96 conditions, with 12, 3% of values lost. The results of this method were compared with FLOC [11], DBF [35] and CC [1], using as a criterion of comparison the MSR, and the size of the biclusters obtained by each method. In addition, algorithms determine the biological significance of the biclusters in connection with information on the yeast cell cycle. The biological relevance is determined based on the statistical significance determined by using the GO annotation database [36]. As for the comparison based on the MSR and the size of the biclusters obtained, MOEA obtained results far superior to the obtained by other methods.

Dharan and Nair [23] proposed Reactive GRASP method. Statistical significance of the found biclusters is assessed to see how well they correspond with the known gene annotation [33]. For this purpose the package *SGD GO gene ontology term finder* [36] were used. The performed tests show that the Reactive GRASP is able to find biclusters with higher statistical significance than the basic GRASP [23] and the CC [1].

Das and Idicula [25] proposed an algorithm based on greedy search combined with PSO. The tests were conducted on expression data of the cell cycle of the yeast Saccharomyces cerevisiae. The data used is based on [34], and consists of 2884 genes under 17 conditions. The results were compared with those of SEBI [20], CC [1], FLOC [11], DBF [35] and Modified Greedy [25]. The comparison was based on the MSR (also named as MSE) presented by [1], and the size of the biclusters. The GS Binary PSO outperformed the other methods, except to DBF, on the MSR, and showed competitive results in the size of the biclusters.

Caldas and Kaski [26] proposed a method based on a hierarchical model (TreeBic). The model assumed that the samples or conditions in a microarray are grouped in a tree structure, where nodes correspond to subsets of the hierarchy. Each node is associated with a subset of genes, for which, samples are highly homogeneous. The tests were conducted on a collection of 199 miRNAs profiled from 218 human tissues from healthy and tumor

cell lines. The results were compared with those obtained by Samba [18] Plaid [13], DC [1] and OPSM [16] methods. TreeBic performed better in general, both in terms of the proportion of biclusters enriched to at least one tissue or GO category, and in terms of the total number of tissues and GO categories enriched. Despite these results, the TreeBic method had the second lowest number of biclusters generated.

4. Proposed algorithm

We propose a multi-objective genetic algorithm, where each individual in the population is a bicluster. The objective is to minimize both the MSR and the bicluster size. Unlike the MOEA proposed in [33], this algorithm does not require of a local search to keep biclusters under the MSR threshold δ. Instead, consider the fact of being under or over the threshold, as a first condition when select the best biclusters. This represents two important advantages, first it avoids the use of the parameter α required in local search, which influences heavily the obtained results. The second advantage is that it reduces the execution time, allowing to use a larger number of individuals and generations.

4.1 Representation of biclusters

A bicluster is represented as a binary string where the first bits correspond to genes and the second part of bits to the conditions. If the binary string has a value one at position j of the first part it indicates that the gene j, the same applies for the condition part of the string. A bicluster consists of the values of selected gene expression under the selected conditions. Figure 1A shows an example of the binary representation of a bicluster. The bicluster corresponding to the binary string shown in Figure 1A is shown if Figure 1C, it is extracted from the expression matrix presented in Figure 1B.

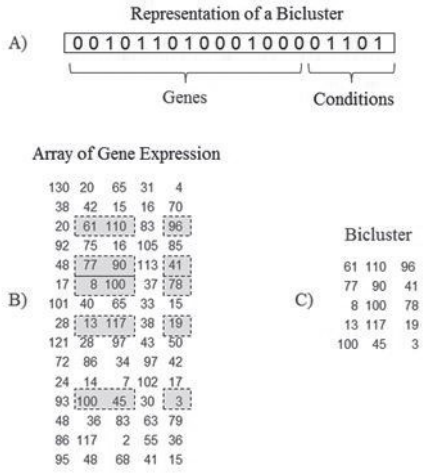

Figure 1. Representation of a bicluster. A) The binary string representing the bicluster. B) An array of gene expression data. C) Bicluster values comprising selected expression (shaded values) of the matrix in Figure B.

4.2 Multiobjective genetic algorithm

The Algorithm 1 starts by creating a population of n biclusters. Each bicluster is created by selecting at random two genes and two conditions of the matrix expression, so that the *MSR* do not exceed the threshold δ. If it is exceeded the selected pair are discarded, and the process repeated until the *MSR* becomes under the threshold.

The nondominated front of each bicluster is computed as it is done in [33]. The nondominated front is calculated based on the

concept of dominance. A bicluster *i* dominates bicluster *j*, if either of the following conditions hold:

1. The MSR of *i* (MSR_i) is less than or equal to MSR_j, the size of *i* ($size_i$) is larger than $size_j$.

2. $size_i$ is greater than or equal to $size_j$, and MSR_i is less than MSR_j.

For a bicluster to belong to a nodominated front, it should not be dominated by some other in the population. Once the biclusters are identified in the first front, they are discarded for the identification of the second front of biclusters. This process is repeated successively until there are no dominated biclusters.

Algorithm 1: Multi-Objective Genetic Biclustering
Input: matrix gene expression, threshold δ MSR
Output: A set of *n* optimized biclusters

1. generate an initial population of *n* biclusters with MSR below δ
2. calculates the nondominated fronts for each bicluster
3. calculates the crowdig distance of each bicluster
4. **repeat**
5. selects best biclusters
6. apply crossover
7. apply mutation
8. combines parent and children populations
9. calculates the nondominated fronts of the combined population
10. calculates the crowding distances of the combined population
11. sort the biclusters of the combined population
12. define new population of *n* biclusters
13. **until** the number of generations without improvement is *ng*
14. **return** the *n* biclusters of the last generation

Step 3 computes the crowding distance of each individual as it is done by Mitra and Banka [33]. This distance is a measure of the degree of saturation of the search space (in terms of size and MSR). The more similar the MSR and size of an individual is to the rest of the population, the lower the crowing distance. This distance is used to maintain diversity in the population.

Once the nondominated fronts and the crowding distance are computed, the selection of the best individuals is performed. The binary selection with crowding is applied. First randomly rearrange the individuals within the population, and two adjacent individuals to conduct the tournament. An inidividual i is chosen on an individual j if it meets any of the following conditions:

1. The *MSR* of *i* is below the threshold δ, and the *MSR* of *j* is not below the threshold.

2. Both *MSR* are on the same side of the threshold δ, and *i* is in a front with lower index than *j*.

3. Both *MSR* are on the same side of the threshold of δ, both belong to the same front, and the crowding distance of *i* is greater than the one corresponding to individual *j*.

Crossover is applied on the selected individuals. For this process individuals are taken in pairs (parents), and creates two new biclusters (children) per each pair of parents. For each child, two random crossover points are selected in the binary strings corresponding to both parents. The first crossover point is set to a bit position corresponding to a gene, and the second crossover point is set to a position corresponding to a condition. The child takes from one of the parents the genes found to the left side of the first crossover point, and from the other parent genes to the right. The same procedure applied to the conditions. The parent bicluster whose genes are taken

from the left side of the first crossing point is chosen randomly. Figure 2 shows an example of the crossover of two biclusters. The child one is created by taking genes from the left side of the first crossing point in parent one, and genes on the right side of the first crossing point of the parent two. Take the conditions on the left side of the second crossing point of parent two and conditions from the right side of the second crossing point of parent one. The child two is created in reverse order of child one.

Subsequently, the mutation process was applied to a percentage of biclusters in the children population. Mutation of a bicluster is done by selecting a random bit in the string, and changing its value. If the bit is a zero value is changed to one which represents a gene or condition that was not considered in the bicluster now is included.

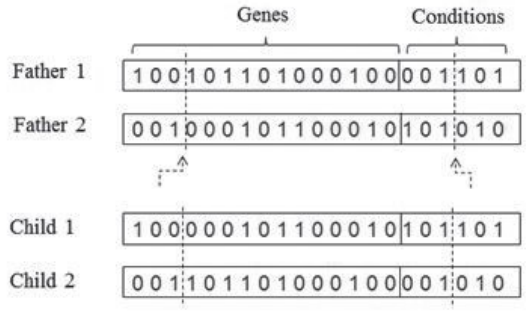

Figure 2. Example of a cross between two biclusters.

Figure 3 shows an example of a mutation in a bicluster. In this example the tenth bit was randomly selected, and modified. The bit corresponds to the position of a gene. The value of this bit was changed from zero to one, which represents the values of expression of the gene number 10 in the matrix expression for the selected conditions (values shaded), will be included in the bicluster.

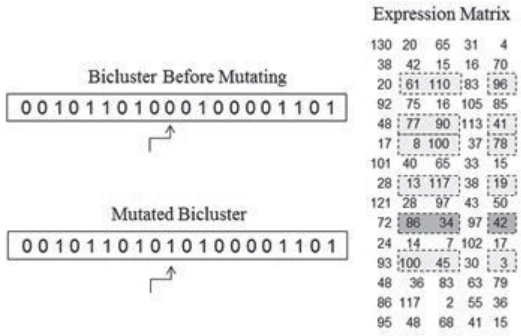

Figure 3. Example of mutation of a bicluster

After the mutation is performed a process which combines both populations (parent and children) is carried out. This process consists in considering only as a single population all biclusters from both populations. For this combined population nondominated fronts and crowding distances were recalculated. Subsequently, the biclusters are ordered for this combined population, according to the following criteria:

1. First fit the biclusters that are below the threshold δ.
2. Within biclusters which are at the same side of the threshold, first fit those having a lower front.
3. Among the biclusters on the same side of the threshold, and in the same front, fit first those with a larger crowding distance.

Once the individuals are arranged in a combined population the first n are selected, which will be considered the next generation of biclusters. This process stops after a number of generations ng without change in the size of the largest bicluster with MSR below

threshold is reached.

5. Experimental results

We apply the multi-objective genetic algorithm on two data sets used as test cases. The first experimental condition tested was the expression of 2884 genes under 17 conditions with the Yeast *Saccharomyces Cerevisiae*, containing 34 nulls. The second set corresponds to the expression of 4026 genes under 96 conditions of *Human* B *cells Lymphoma*, with 47,639 null values corresponding to 12.3% of the full set. Both sets of data were taken from the site http://arep.med.harvard.edu/ [37]. The experiments were performed using an MSR threshold of $\delta = 300$ for the yeast, and a threshold $\delta = 1200$ for the Lymphoma. Although there is no justification from the point of view of biology, these values have been used extensively to evaluate and compare a variety of biclustering methods. In the case of the yeast assembly, null values were replaced by random values in the range 0 and 800. In the case of Lymphoma null values were replaced by random values in a range of -800 to 800. Both threshold values selected for the MSR, as well as the strategy and range to replace the zero values were established in the discussed manner in order to make a more direct comparison with the results reported in other studies.

The experiments consisted of 30 runs with each data set, using populations of 50 individuals, and setting a value of 400 for the number of generations without improvement. A 90% of selection, while 100% and 50% for crossover and mutation, respectively. The method was coded and implemented in C Sharp, experiments were performed under the Windows using Visual Studio 7 Ultimate 2010 in a Laptop of 1.73 GHz speed and 1.00 GB of RAM. The algorithm receives as input a text file with the matrix of expression data to be processed. Returns as output another text file with the built biclusters, the values that were used to replace the null values in the array, and a descriptive information on the best biclusters built.

The average MSR value, the number of genes, number of conditions, and the average and maximum size of the discovered biclusters were used as assessment criteria. Table 1 shows a comparison of the results obtained from the Yeast dataset. For this comparison the FLOC algorithms [10], DBF [35], MOEA [33], and the one presented by Cheng and Church [1] are considered here. This is a representative group of algorithms for biclustering, which have been analyzed frequently in literature. The results reported for these algorithms were taken from the work of Mitra and Banka [33].

The proposed algorithm (called MOGA) significantly outperforms other algorithms in the size of the biclusters discovered under the defined threshold. The MOGA obtains larger biclusters, even larger than those of MOEA, which already exceeds the performance of other algorithms. A very important advantage MOGA with respect to MOEA is that it does not require a local search to keep the biclusters below the threshold, which avoids the handling of the parameter α (used in various methods), whose proper choice largely influences on the results.

Table 1. Comparative results of biclustering methods on data from the Yeast Saccharomyces Cerevisiae, using a threshold MSR δ = 300.

Method	Average MSR	Average bicluster size	Size of larger bicluster
FLOC [10]	187.54	1825.78	2000
DBF [35]	114.70	1627.20	4000
Cheng-Church [1]	204.29	1576.98	4485
MOEA [33]	234.87	10301.71	14828
MOGA	282.45	14112.60	16488

The results of the algorithm using the Lymphoma data were compared with the results reported by Mitra and Banka [33], which were the best results in the literature (see Table 2). This table shows that MOGA outperforms the best MOEA result, both in terms of the size of biclusters found, as in *CI* value. The *CI* (Consistency Index) introduced by Mitra and Banka, represents the relationship between the MSR of a bicluster and its size. This ratio indicates how well the two requirements of biclusters are met: i) the expression levels of genes are similar over a range of conditions, i.e., must have a low MSR, and ii) the size is as large as possible. A bicluster is considered better as its *CI* value is smaller.

Table 2. Best biclusters found on the data set of the Human B-Lymphoma cells, MSR using a threshold δ = 1200.

Method	MSR	Bicluster size	CI
MOEA [33]	1199.98	37560	0.032
MOGA	1199.38	43834	0.027

6. Conclusions

A new multi-objective genetic algorithm for the biclustering of gene expression data has been proposed. Experiments conducted on two sets of biological data, which have been used widely as test cases, have shown that the proposed algorithm performs better than others currently reported in the literature. An important feature of our algorithm is that it does not require a local search, contrary to some current algorithms which require maintaining the MSR below the threshold by means of this technique.

Experiments were focused on the discovery of large biclusters with MSR below predefined thresholds for both sets of data, which are widely accepted by the scientific community.

Future work will assess the biological significance of the generated biclusters, based on ontological annotations. The proposed evolutionary algorithm will be redesigned to work with parallel models of computation, which will allow us to deal with larger instances.

7. References

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

[2] D. S. Rodriguez, J. C. Riquelme, and J. S. Aguilar, "Analisis de datos de expresion genética mediante tecnicas de biclustering," tech. rep., Universidad de Sevilla, 2000.

[3] J. Aguilar, "Shifting and scaling patterns from gene expression data," Bioinformatics, vol. 21, 2005, pp. 3840–3845.

[4] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: a survey," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 1, no. 1, 2004, pp. 24–45.

[5] J. A. Hartigan, "Direct clustering of a data matrix," Journal of the American Statistical Association (JASA), vol. 67, no. 337, 1972, pp. 123–129.

[6] S. Busygin, G. Jacobsen, and E. Kramer, "Double conjugated clustering applied o leukemia microarray data," in Proceedings of the 2nd SIAM International Conference on Data Mining, Workshop on Clustering High Dimensional Data, 2002.

[7] G. Getz, E. Levine, and E. Domany, "Coupled two-way clustering analysis of gene microarray data," Proceedings of the Natural Academy of Sciences USA, 2000, pp. 12079–12084.

[8] A. Califano, G. Stolovitzky, and Y. Tu, "Analysis of gene expression microarays for phenotype classification," in Proceedings of the International Conference on Computacional Molecular Biology, 2000, pp. 75–85.

[9] Q. Sheng, Y. Moreau, and B. D. Moor, "Biclustering microarray data by gibbs sampling," Bioinformatics, vol. 19, no. 2, 2003, pp. ii196–ii205.

[10] J. Yang, W. Wang, H. Wang, and P. Yu, "δ-clusters: Capturing subspace correlation in a large data set" Proceedings of the 18th IEEE International Conference on Data Engineering, 2002, pp. 517–528.

[11] J. Yang, W. Wang, H. Wang, and P. Yu, "Enhanced biclustering on expression data," Proceedings of the 3rd IEEE Conference on Bioinformatics and Bioengineering, 2003, pp. 321–327.

[12] H. Wang, W. Wang, J. Yang, and P. S. Yu, "Clustering by pattern similarity in large data sets," Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, 2002, pp. 394–405.

[13] L. Lazzeroni and A. Owen, "Plaid models for gene expression data," Statistica Sinica, vol. 12, 2002, pp. 61–86.

[14] E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller, "Rich probabilistic models for gene expression," Bioinformatics, vol. 17, no. Suppl. 1, 2001, pp. S243–S252.

[15] E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller, "Decomposing gene expression into cellular processes," Proceedings of the Pacific Symposium on Biocomputing, vol. 8, 2003, pp. 89–100.

[16] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini, "Discovering local structure in gene expression data: The order-preserving submatrix problem," in Proceedings of the 6th International Conference on Computacional Biology (RECOMB'02), 2002, pp. 49–57.

[17] T. M. Murali and S. Kasif, "Extracting conserved gene expression motifs from gene expression data," Proceedings of the Pacific Symposium on Biocomputing, vol. 8, 2003, pp. 77–88.

[18] A. Tanay, oded Sharan, and R. Shamir, "Discovering statistically significant biclusters in gene expression data," Bioinformatics, vol. 18, no. Suppl. 1, 2002, pp. S136–S144.

[19] J. Liu and W. Wang, "Op-cluster: Clustering by tendency in high dimensional space," Proceedings of the 3rd IEEE International Conference on Data

Mining, 2003, pp. 187–194.

[20] F. Divina and J. S. Aguilar, "Biclustering of expression data with evolutionary computation," IEEE Transactions on Knowledge and Data Engineering, vol. 18, 2006, pp. 590–602.

[21] K. Bryan, P. Cunningham, and N. Bolshakova, "Biclustering of expression data using simulated annealing," 18th IEEE Symposium on Computer Based Medical Systems (CBMS'05), 2005, pp. 383-388.

[22] J. Gu and J. S. Liu, "Bayesian biclustering of gene expression data," BMC Genomics, vol. 9, no. Suppl. 1, 2008, p. S4.

[23] S. Dharan and A. S. Nair, "Biclustering of gene expression data using reactive greedy randomized adaptive search procedure," BMC Bioinformatics, vol. 10, no. Suppl. 1, 2009, p. S27.

[24] G. Pandey, G. Atluri, M. Steinbach, C. L. Myers, and V. Kumar, "An association analysis approach to biclustering," ACM SIGKDD, ACM New York, NY, USA, 2009, pp. 677-686

[25] S. Das and S. M. Idicula, "Greedy search-binary pso hybrid for biclustering gene expression data," International Journal of Computer Applications, vol. 2, no. 3, 2010, pp. 0975–8887.

[26]J. Caldas and S. Kaski, "Hierarchical generative biclustering for microrna expression analysis," RECOMB, 2010, pp. 65-79.

[27] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Buhlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler, "A systematic comparison and evaluation of biclustering methods for gene expression data," Bioinformatics, vol. 22, 2006, pp. 1122–1129, 2006.

[28] J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai, "Revealing modular organization in the yeast transcriptional network," Nature Genetics, vol. 31,2002, pp. 370–377.

[29] J. Ihmels, S. Bergmann, and N. Barkai, "Defining transcription modules using large-scale gene expression data," Bioinformatics, vol. 20, 2004, pp. 1993–2003.

[30] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene ontology: tool for the unification of biology," Nature Genetics, vol. 25, no. 1, 2000, pp. 25–29.

[31] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown, "Genomic expression programs in the response of yeast cells to environmental changes," Molecular Biology of the Cell, vol. 11, 2000, pp. 4241–4257.

[32] A. Wille, P. Zimmermann, E. Vranova, A. Furholz, O. Laule, S. Bleuler, L. Hennig, A. Prelic, P. von Rohr, L. Thiele, E. Zitzler, W. Gruissem, and P. Buhlmann, "Sparse graphical gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana," Genome Biology, vol. 5, 2004, p. R92.

[33] S. Mitra and H. Banka, "Multi-objective evolutionary biclustering of gene expression data," Journal of the Pattern Recognition Society, vol. 39, 2006, pp. 2464–2477.

[34] S. Tavazoie, J. Hughes, M. Campbell, R. Cho, and G.M.Church, "Systematic determination of genetic network architecture," Nat. Genet., vol. 22, no. 3, 1999, pp. 281–285.