Virtual Error: A New Measure for Evolutionary Biclustering

Beatriz Pontes[1], Federico Divina[2], Raúl Giráldez[2], and Jesús S. Aguilar–Ruiz[2]

[1] Department of Computer Science, University of Seville
Avenida Reina Mercedes s/n, 41012 Sevilla, Spain
bepontes@lsi.us.es
[2] School of Engineering, Pablo de Olavide University
Ctra. de Utrera, km. 1, 41013, Sevilla, Spain
{fdivina,giraldez,aguilar}@upo.es

Abstract. Many heuristics used for finding biclusters in microarray data use the mean squared residue as a way of evaluating the quality of biclusters. This has led to the discovery of interesting biclusters. Recently it has been proven that the mean squared residue may fail to identify some interesting biclusters. This motivates us to introduce a new measure, called *Virtual Error*, for assessing the quality of biclusters in microarray data. In order to test the validity of the proposed measure, we include it within an evolutionary algorithm. Experimental results show that the use of this novel measure is effective for finding interesting biclusters, which could not have been discovered with the use of the mean squared residue.

1 Introduction

Nowadays, technological advances offer the possibility of completely sequentialize the genome of some living species. This constitutes a great source of information which needs to be analyzed. Microarray techniques allow us to study genomes on their own or also to combine some of them in order to extract relational knowledge [12].

Microarray data are usually transformed into a numerical matrix which could then be analyzed. There exist various techniques to extract relevant information from a microarray, depending on the specific application in study. These techniques include clustering methods [4], where the goal is to cluster together genes that have a similar behaviour under all the experimental conditions. This grouping is carried out by means of any specific algorithm or mathematical formula based on genes similarity over all conditions [13]. It may be interesting, however, to analyze whether several genes in a microarray show the same behaviour under a subset of the experimental conditions. This has motivated a recent line of research named biclustering. Biclustering techniques aim at individuating subsets of genes that present the same behaviour under a subset of experimental conditions. This problem has been proven to be even much more complex than clustering [8].

Biclustering was first applied to genomic data in [6], where the authors present a greedy search method for finding biclusters. In the same work, a measure for assessing the quality of biclusters, named *Mean Squared Residue* (MSR), is proposed. This measure has been used by many researches who have proposed different heuristics for biclustering biological data, e.g., [2,14]. Some other authors have established a search model to detect significant biclusters, without using a specific formula to optimize [11]. For a review of different biclustering techniques, we refer the reader to [9,10]. Among the used techniques, it is interesting to emphasize the application of evolutionary computation to the problem of finding biclusters in microarray data [5,8]. In these work the search was biased towards biclusters with low MSR.

The use of MSR to guide the search for biclusters in microarray data has led to the discovery of interesting biclusters. However, it has been proven that MSR may fail to recognize some interesting biclusters as quality biclusters [1]. This motivates us to introduce a new measure, called *Virtual Error* (VE), for assessing the quality of biclusters in microarray data. In order to evaluate the validity of the proposed measure, we include it within an evolutionary algorithm (EA). In a previous version of this EA, the search was guided mainly by the MSR. Experimental results show that the so modified EA is capable of finding interesting biclusters, which could not have been discovered with the use of MSR.

This paper is organized as follows: in Section 2 we present the motivations for this paper, we therefore describe the quality measure we propose in Section 3. Section 4 describes the used EA and how VE has been included into the algorithm, while some experimental results are shown in Section 5. Finally, Section 6 summarizes the main conclusions.

2 Motivation

One of the most used quality measures for biclusters is the *Mean Squared Residue*, MSR [6]. MSR tries to evaluate the coherence of the genes and conditions of a bicluster $\mathcal{B}$ consisting of I rows and J columns. MSR is defined as:

$$\mathtt{MSR}(\mathcal{B}) = \frac{1}{I \cdot J} \sum_{i=1}^{i=I} \sum_{j=1}^{j=J} (b_{ij} - b_{iJ} - b_{Ij} + b_{IJ})^2 \tag{1}$$

where b_{ij}, b_{iJ}, b_{Ij} and b_{IJ} represent the element in the ith row and jth column, the row and column means, and the mean of the submatrix, respectively. If the gene expression levels fluctuate in unison under the conditions contained in a bicluster $\mathcal{B}$, then $\mathtt{MSR}(\mathcal{B}) = 0$. In general, the lower the MSR, the stronger the coherence exhibited by the bicluster, hence the better the quality. It follows that a trivial or constant bicluster where there is no fluctuation is characterized by a very low value of MSR. In order to reject these kind of biclusters, most heuristics combine the MSR with some other measures, e.g., the row variance [8,6].

As demonstrated in [1], MSR may not be the optimal measure for assessing the quality of some kinds of biclusters. In this work, the author makes a further study on the main characteristics inherent to biclusters, extracting from them

two main principles, shifting patterns and scaling patterns. Genes in a bicluster might present either one of these patterns or both of them simultaneously. It is demonstrated that the MSR value is useful to recognize shifting behaviours in the biclusters, while it may fail to recognize a bicluster presenting scaling patterns.

Figure 1 shows two biclusters whose genes fluctuate in unison under the conditions contained in the bicluster. Each line in the graphs represents the expression levels of a gene under different conditions. This figure also presents the numerical values for each bicluster in a matrix, where columns correspond to genes and rows to experimental conditions. Despite the fact the the genes present the same behaviour under the experimental conditions, the MSR value for the two biclusters does not seem to indicate that they are equally good biclusters. The MSR for the two biclusters is 236.25 and 385, respectively. As it can be seen in Figure 1, the only difference between these two biclusters is represented by the value assumed by the genes under the third condition. Comparing these two biclusters graphically, we cannot conclude that the left one is better than the right one, as it would be unfair to claim that genes presenting lower values for a certain condition are preferable to higher values.

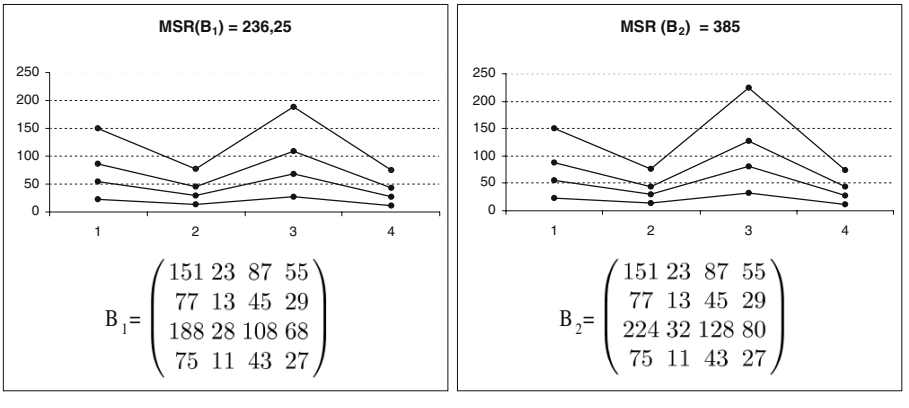

Fig. 1. Examples of similar biclusters with different MSR values

This motivates us to propose a novel measure for assessing the quality of bicluster in microarray data. This measure should avoid taking into account the numerical similarities in the submatrix. Instead, it should quantify the behaviour of the genes under all the conditions contained in the bicluster. We therefore propose a novel criterion, called *Virtual Error* (VE), based on the concept of behavioural patterns.

3 Virtual Error

The main idea behind VE is to create a pattern from each bicluster in order to represent the general trends within it. This pattern will try to capture the

overall behaviour of the genes over the conditions in the bicluster, checking if the expression levels of the genes vary in unison, with independence on the specific values and slopes. VE is based on the use of a tendency pattern for each bicluster. Therefore, this quality value will depend on the way in which the pattern is built.

Next, we formally define how the pattern is created, starting from a bicluster $\mathcal{B}$, consisting of I conditions (rows) and J genes (columns), and where each element in the bicluster is represented as b_{ij}, where $1 \leq i \leq I$ and $1 \leq j \leq J$.

Definition 1 (Tendency Pattern). *Given a bicluster $\mathcal{B}$ containing I conditions and J genes, we define the tendency pattern as a collection of I elements P_i, each of them given by: $P_i = \frac{\sum_{j \in J} b_{ij}}{J}, b_{ij} \in \mathcal{B}, 1 \leq i \leq I, 1 \leq j \leq J$.*

Thus, each of the points of the pattern will represent a significative value for all genes under a specific condition.

Once the pattern has been built, the aim is to examine to what extent the genes are similar to it. In this sense, we need a mechanism in order to do an appropriate comparison between each gene and the pattern. This mechanism would be responsible for smoothing every gene behaviour, since the most important issue is to characterize their conduct but not their numerical values. An example of this is represented by scaling patterns (see section 2), where two different genes may present the same behaviour under the same experimental conditions, but with different magnitude of expression values.

Definition 2 (Standardization). *Let $\mathcal{B}$ be a bicluster containing J genes and I conditions. Let b_{ij} denotes the elements of $\mathcal{B}$, for $1 \leq i \leq I$ and $1 \leq j \leq J$. We then define the standardization of $\mathcal{B}$ as the bicluster $\mathcal{B}'$, whose element b'_{ij} are $b'_{ij} = \frac{b_{ij} - b_{Ij}}{\sigma_{g_j}}, 1 \leq i \leq I, 1 \leq j \leq J$, where σ_{g_j} is the standard deviation of all the expression values of gene j.*

By means of the standardization, two distinct tasks are carried out. The first one is to shift all the genes to a similar range of values (near 0 in this case). The second one is to homogenize the expression values for each gene, modifying in this way their values under all the conditions, and smoothing their graphical representation.

It is important to notice that in order to fairly compare genes values to pattern values, all of them must be enclosed in the same range of values. Thus, the pattern must be also standardized, generating a so called virtual pattern. This is shown in equation 2, where P_i refers to the pattern value for condition i, and $\overline{P}$, σ_P refer to the mean and the deviation of all the values in the pattern, respectively.

$$P'_i = \frac{P_i - \overline{P}}{\sigma_P} \tag{2}$$

Definition 3 (Virtual Error). *Given a bicluster $\mathcal{B}$ containing I conditions and J genes, and a pattern P containing I values, we define VE as the mean of*

the numerical differences between each standardized gene and pattern values for each condition:

$$\texttt{VE}(\mathcal{B}) = \frac{1}{I \cdot J} \sum_{i=1}^{i=I} \sum_{j=1}^{j=J} (b'_{ij} - P'_i)$$

$\texttt{VE}(\mathcal{B})$ corresponds to our measure proposal, which states that biclusters with lower levels of VE are considered to be better than those with higher values. This is due to the fact that VE computes the differences between the standardized genes and the pattern, therefore, the more similar the genes are, the lower the value for VE. It then is important to note that shifting patterns do not increase the value of VE, since standardizing genes allows the VE to compare their behaviour within the same range of values. In the case of scaling patterns, it has a minimal effect on our measure, as the standardization decreases the numerical differences among genes. As an instance, biclusters shown in Figure 1 have VE values practically equal to zero ($\texttt{VE}(B_1) = 2,77 \times 10^{-17}$ and $\texttt{VE}(B_2) = -1,39 \times 10^{-17}$). These values indicate that VE considers both biclusters as equally good. VE owes its name to the fact that the error is not computed using the original genes and pattern, but with virtual ones, once the original data has been standardized.

In the whole, this new measure provides a value for each bicluster, quantifying the similarities among genes by means of comparing their behaviour to a pattern. This comparison is carried out in such a way that shifting and scaling trends are minimally penalized, while behavioural differences among genes notably increase the quality value.

4 Description of the Algorithm

In order to assess the effectiveness of the VE as a measure for establishing the quality of biclusters, we have incorporated it in the EA SEBI [8]. In order to use the VE within SEBI, we have modified the fitness function of SEBI, as explained next.

SEBI adopts a sequential covering strategy: an EA, called EBI (for Evolutionary BIclustering), is called n times, where n is an user-defined parameter. EBI takes as input the expression matrix and returns a bicluster, which is stored in a list called *Results*, and EBI is called again.

In order to avoid too much overlapping among the found biclusters, we associate a weight to each element of the expression matrix. After a bicluster is returned, these weights are adjusted. The weight of an element depends on the number of biclusters in *Results* containing the element. The more biclusters cover an element, the higher the weight of the element will be (see [8] for more details).

In [8], the fitness of an individual X was:

$$f(X) = \frac{\texttt{MSR}(X)}{\delta} + \frac{1}{row\_variance(X)} + +w\_d + penalty \qquad (3)$$

where $MSR(X)$ represents the mean squared residue of X, δ is a user supplied threshold, $row\_variance(X)$ is the row variance of X, w_d is used for penalizing

smaller biclusters and *penalty* is the sum of the weights assigned to the element of the expression matrix belonging to the bicluster X. Notice that the fitness has to be minimized. It follows that the aim of EBI was to find biclusters with mean squared residue lower than δ, with high volume, with a relatively high row variance, and minimizing the effect of overlapping among biclusters.

In the version of EBI we use in this paper, we have modified the fitness function defined in equation 3 in the following way:

$$f(X) = \texttt{VE}(X) + w\_d + penalty \tag{4}$$

where VE(X) is the virtual error of X, w_d and *penalty* are defined as in [8], but are scaled to adapt to VE. Also this fitness has to be minimized, hence we prefer biclusters characterized by a low VE, high volume and minimum overlapping with biclusters contained in *Results*. In this fitness function, we do not use the row variance, as it happened in equation 3. This is because with the use of VE we do not need this factor to reject trivial biclusters, as it happened when the MSR was used.

As in [8], the initial population consists of biclusters containing only one element of the expression matrix. Tournament selection is used for selecting parents. Selected pairs of parents are recombined with a crossover operator with a given probability p_c (default value 0.9), and the resulting offspring is mutated with a probability p_m (default value 0.1). Elitism is applied with a probability p_e (default value 0.75). At the end of the evolutionary process, EBI returns the best individual, according to the fitness.

Each individual of the population encodes one bicluster. Biclusters are encoded by means of binary strings of length $N + M$, where N and M are the number of rows (genes) and of columns (conditions) of the expression matrix, respectively. Each of the first N bits of the binary string is related to the rows, in the order in which the bits appear in the string. In the same way, the remaining M bits are related to the columns. If a bit is set to 1, it means that the relative row or column belongs to the encoded bicluster; otherwise it does not.

5 Experiments

In order to show the quality of our approach, we run SEBI on two well known datasets. The first one, the yeast *Saccharomyces cerevisiae* cell cycle expression dataset [7], is a microarray which contains 2884 genes and 17 conditions. The second dataset is the human B-cells expression data [3], that consists of 4026 genes and 96 conditions.

With regard to the EA parameters, we used the same parameter setting as in [8]. Thus, we can compare the results with those obtained in the previous EA version, where the MSR was used as main term of the fitness function. Specifically, we used a population of 200 individuals and a number of generations of 100. The crossover probability was of 0.85 and the mutation probability was 0.2. The number of biclusters was set to 100, that is, SEBI generated one hundred biclusters for each dataset.

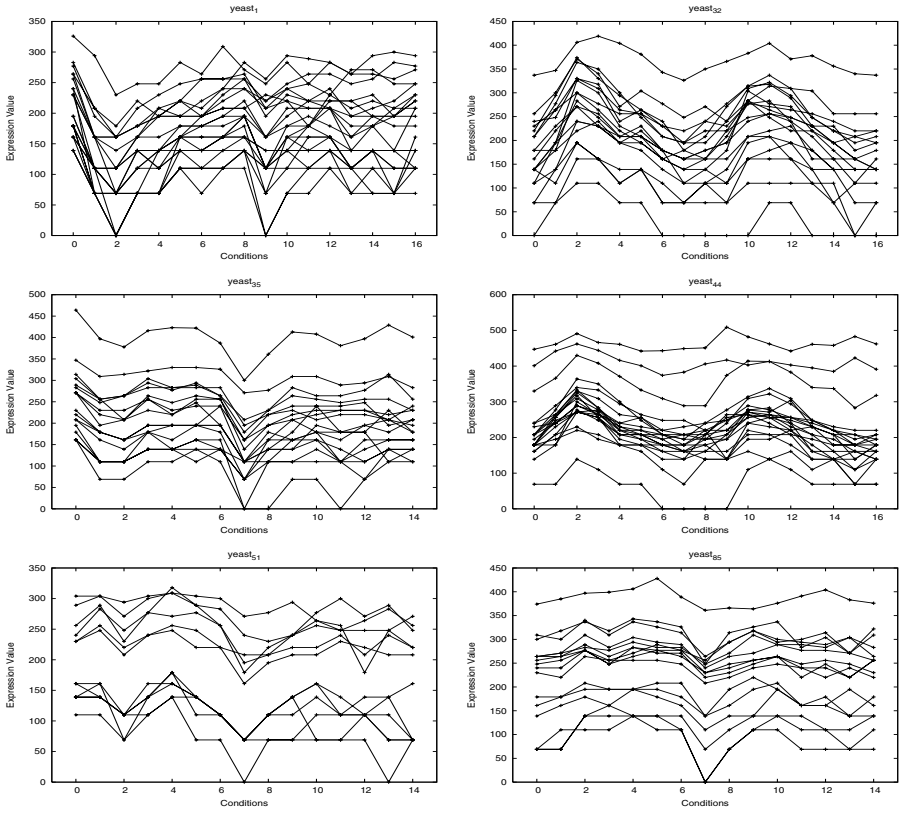

Fig. 2. Biclusters found on the Yeast dataset

Figure 2 shows six biclusters out of the one hundred found on the yeast dataset (see Table 1 for numerical results). The bicluster labelled $yeast_1$ is found with the fist call of `SEBI`. As we can see, this bicluster is visually interesting. Furthermore, it has low `VE` of 0.38, but high residue of 535.8. This is a remarkable result, since first biclusters found with `VE` were interesting, while this is not the case with `MSR`, as it can be seen in [8] and [6].

In general, we can notice from a visual inspection of all the biclusters that the genes present a similar behaviour under the set of selected conditions. All the `VE` of the biclusters are lower than 0.38. We find especially interesting the fact that, some genes in the bicluster are distant from the rest of it but they show a similar trend. For example, bicluster $yeast_{44}$ is interesting because it differentiates three genes at the top of the graph from the others, although all the genes seem to have the same behaviour. This points out that `VE` is not sensitive to the scale or magnitude difference in the expression values of the genes. Furthermore, this is a kind of bicluster difficult to find by using the `MSR` [1].

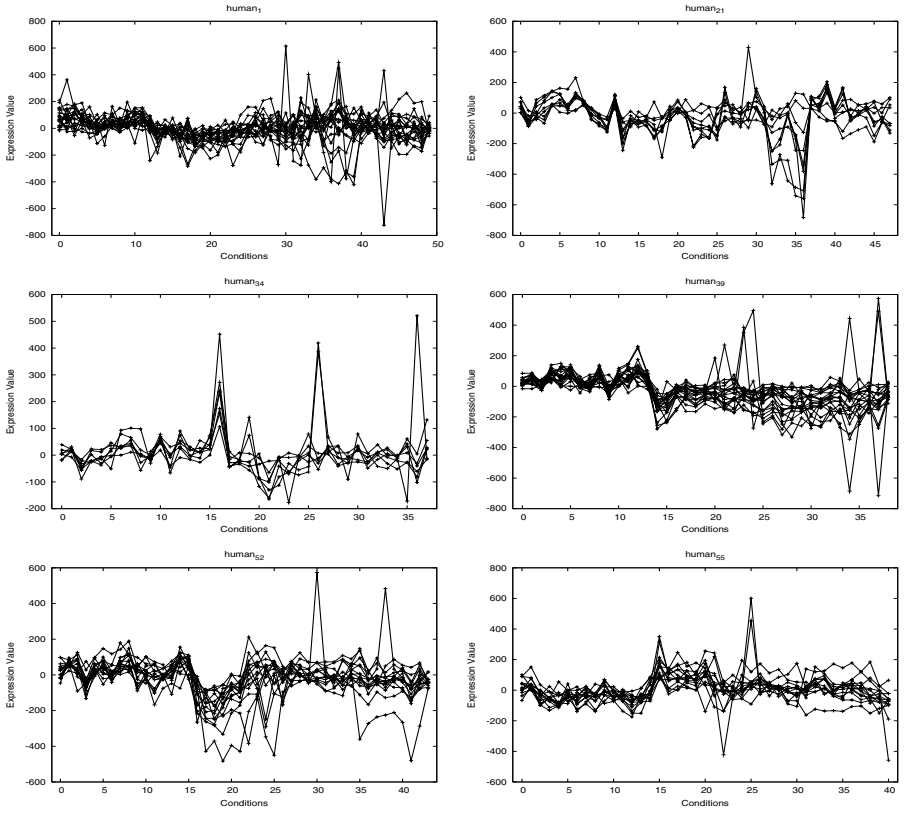

Fig. 3. Biclusters found for the human dataset

Concerning the size of the biclusters, many biclusters contain all the seventeen conditions. A similar result was also obtained in the previous version of SEBI [8], where MSR was used as main term in fitness function. However, the number of genes was higher in this case than those obtained in the aforementioned work. Thus, the use of VE allows SEBI to include more genes without damaging the bicluster quality.

The human dataset is larger and more complex than the yeast dataset. Therefore, it is also more complex to find good biclusters with low VE. Six out of one hundred biclusters found on such dataset are shown in Figure 3 (see Table 1 for numerical results).

The bicluster ($human_1$), that SEBI with VE finds in the first execution of EBI, is interesting. However, it does not happen the same with MSR. Although it has also low VE (0.57), the most remarkable aspect is that the residue of this bicluster is very high (7173.5). This strengthens our conclusion that SEBI can find interesting biclusters in the first iterations when VE is used as quality measure instead of MSR.

Table 1. Information about biclusters of Figures 2 and 3

Yeast Dataset				
Bicluster	VE	MSR	#Genes	#Cond.
$yeast_1$	0.38	535.8	23	17
$yeast_{32}$	0.29	408.9	19	17
$yeast_{35}$	0.28	380.6	18	15
$yeast_{44}$	0.30	583.5	21	17
$yeast_{51}$	0.34	346.7	12	15
$yeast_{85}$	0.36	232.1	16	15

Human Dataset				
Bicluster	VE	MSR	#Genes	#Cond.
$human_1$	0.57	7173.5	21	50
$human_{21}$	0.39	6405.4	9	48
$human_{34}$	0.43	3278.8	7	38
$human_{39}$	0.44	5786.1	21	39
$human_{52}$	0.42	5660.7	15	44
$human_{55}$	0.46	4069.5	14	41

Regarding the shape of the biclusters, all of them present a similar trend, although the genes are closer in this case than in yeast dataset case. Also in the human dataset, VE is not sensitive to the magnitude difference in the values of the genes. This aspect can be observed in the bicluster $human_{52}$, where there is a decrease of the expression level between 15 and 25 for all the genes but with different scale. Finally, the number of genes and conditions are similar to those produced in the previous version of SEBI with MSR.

Table 1 summarizes the numerical results for Figures 2 and 3. The left table corresponds to the yeast dataset, while the right one to the human dataset. For each table, the first column indicates the name of bicluster. The second column gives the VE value and the third ones the MSR measurement for each bicluster. The last two columns report the number of genes and conditions of the bicluster, respectively.

The most interesting result that can be extracted from these tables is that the biclusters present low VE but high MSR for both datasets. Taking into account that these bicluster are interesting, they could be rejected by the other approaches which use the MSR as main quality measure. For instance, the version of SEBI proposed in [8] used a threshold for rejecting biclusters with MSR higher than this threshold. For the yeast dataset this threshold was set to 300, and for the human dataset to 1200, as in [6]. Therefore, all the biclusters shown in Table 1, with the exception of $yeast_{85}$, would have been rejected.

6 Conclusions

In this work, we have proposed a novel measure for assessing the quality of biclusters in microarray data, called *Virtual Error* (VE). VE is based on the concept of tendency pattern. The majority of the existing biclustering methods are based on the well-known *Mean Squared Residue* (MSR) as the quality measure. However, MSR may fail to recognize some interesting biclusters.

In order to test the goodness of our proposal, we have used VE as main term in the fitness function of an EA. We have then applied the resulting EA to two well-known datasets. From experimental results we can draw the following three main conclusions.

By using VE, the EA found very interesting biclusters with very low VE values. The same biclusters would not have been considered as high-quality biclusters if evaluated with MSR. The row variance is not needed in order to reject trivial or constant biclusters as it happens when MSR is the main term in the fitness function. Another result is that VE is not sensitive to the scale or magnitude difference in the expression values of the genes, as long as they present the same behaviour. It has been proven that this kind of biclusters is difficult to find by using the MSR.

References

1. J. S. Aguilar-Ruiz. Shifting and scaling patterns from gene expression data. *Bioinformatics*, 21:3840–3845, 2005.
2. J. S. Aguilar-Ruiz, D. S. Rodriguez, and D. A. Simovici. Biclustering of gene expression data based on local nearness. In *1Proceedings of EGC 2006,*, pages 681–692, Lille, France, 2006.
3. A. A. Alizadeh, M. B. Eisen, R. E. Davis, and et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.
4. A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3-4):281–297, 1999.
5. S. Bleuler, A. Prelić, and E. Zitzler. An EA framework for biclustering of gene expression data. In *Congress on Evolutionary Computation (CEC-2004)*, pages 166–173, Piscataway, NJ, 2004. IEEE.
6. Y. Cheng and G. M. Church. Biclustering of expression data. In *In Proceedings of the 8th International Conference on Intellingent Systemns for Molecular Biology*, pages 93–103, La Jolla, CA, 2000.
7. R. Cho, M. Campbell, E. Winzeler, and et al. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2:65–73, 1998.
8. F. Divina and J. S. Aguilar-Ruiz. Biclustering of expression data with evolutionary computation. *IEEE Transactions on Knowledge & Data Engineering*, 18(5):590–602, 2006.
9. S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions on Computational Biology and Bioinformatics*, 1:24–25, 2004.
10. A. Prelić, S. Bleuler, P. Zimmermann, and et al. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22:1122–1129, 2006.
11. A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18:136–144, 2002.
12. C. Tilstone. Dna microarrays: Vital statistics. *Nature*, 424:610–612, 2003.
13. H. Wang, W. Wang, J. Yang., and P. S. Yu. Clustering by pattern similarity in large data sets. In *ACM SIGMOD International Conference on Management of Data*, page 394–405, Madison, WI, 2002.
14. J. Yang, H. Wang, W. Wang, and P. S. Yu. An improved biclustering method for analyzing gene expression profiles. *International Journal on Artificial Intelligence Tools*, 14:771–790, 2005.