A Multi-Objective Approach to Discover Biclusters in Microarray Data

Federico Divina
School of Engineering
Pablo de Olavide University
Seville, Spain
fdivina@upo.es

Jesus S. Aguilar–Ruiz
School of Engineering
Pablo de Olavide University
Seville, Spain
aguilar@upo.es

ABSTRACT

The main motivation for using a multi–objective evolutionary algorithm for finding biclusters in gene expression data is motivated by the fact that when looking for biclusters in gene expression matrix, several objectives have to be optimized simultaneously, and often these objectives are in conflict with each other. Moreover, the use of evolutionary computation is justified by the huge dimensionality of the search space, since it is known that evolutionary algorithms have great exploration power.

We focus our attention on finding biclusters of high quality with large variation. This is because, in expression data analysis, the most important goal may not be finding biclusters containing many genes and conditions, as it might be more interesting to find a set of genes showing similar behavior under a set of conditions. Experimental results confirm the validity of the proposed technique.

Categories and Subject Descriptors

I.5.3 [**Pattern Recognition**]: Clustering; J.3 [**Life and Medical Sciences**]; I.2 [**Artificial Intelligence**]

General Terms

Algorithms

Keywords

Biclustering, Gene Expression Data, Multi–Objective Evolutionary Computation

1. INTRODUCTION

Microarray techniques allow to measure the expression level of thousands of genes under different conditions in a single experiment, producing in this way a huge amount of data. Usually these data are organized in matrices, where rows represent genes and columns represent experimental conditions. Each element in the matrix refers to the expression level of a particular gene under a specific condition. These data have enormous potential in gene profiling, facilitating the prognosis and the discovering of subtypes of diseases.

Clustering is the most commonly applied technique for analyzing gene expression data, with the main goal of finding groups of genes that present a similar variation of expression level. However, relevant genes are not necessarily related to every condition. In other words, there are genes that can be relevant for a subset of conditions [18]. On the contrary, it is also possible to discriminate groups of conditions by using different groups of genes. From this point of view, clustering can not only be addressed horizontally (conditions) or vertically (genes), but also in the two dimensions simultaneously. This approach, named *biclustering*, identify groups of genes that show "similar" levels of expression under a specific subset of experimental conditions.

Biclustering was first introduced by [15], as a way to cluster simultaneously rows and columns of a matrix, and it was named "direct clustering". The goal was to find biclusters with minimum variance, which ideally provided biclusters of size 1, since they looked for constant biclusters (constant values within the sub–matrix).

In order to avoid this problem, k biclusters were searched for at a time. Cheng and Church [9] proposed the biclustering of gene expression data, introducing the *residue* of an element in the bicluster and the *mean squared residue* (MSR) of a sub–matrix. The *row variance* was used in order to reject trivial biclusters. Getz et al. [13] presented the coupled two–way clustering. It uses hierarchical clustering applied separately to each dimension and then they defined the process for combining both results. Lazzeroni and Owen [16] used "plaid models" in the same context, where the concept of "layers" is used to compute the values in the data matrix, which is described as a linear function of layers. Basically, each element is seen as a superposition of layers. Yang et al. [19] presented δ–clusters, and the same authors improved the Cheng and Church's approach in FLOC [20], paying attention to missing values. FLOC follows the same technique as Cheng and Church's algorithm, by adding/removing each row/column to a set of initial biclusters, improving its quality iteratively. Tanay et al. [1] identified biclusters by means of a bipartite graph–based model and using a greedy approach to add/remove vertices in order to find maximum weight subgraphs, which are related to its statistical significance. Evolutionary Algorithms (EAs) have been used in [3, 7, 12] for finding biclusters in gene expression data. In these works single–objective EAs were used. In particular, in [7], the EA adopted, incorpo-

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

rated some ad–hoc search techniques similar to those used in [9]. In [8] the order preserving submatrix method by Ben–Dor [6] was used inside an EA for biclustering in order to treat separately strongly related experiments such as time series. For an overview of biclustering techniques, we refer the reader to [17].

When searching for biclusters in microarray data, several objectives, e.g., the volume and the mean squared residue, are to be optimized at the same time. Often, these objectives are in conflict with each other. For example, a bicluster consisting of just one element has mean squared residue equal to zero, or, again, a constant bicluster have row variance equal to zero, but also mean squared residue equal to zero. It follows that the problem of finding biclusters can be straightforwardly seen as a multi–objective optimization problem. This represents our main motivation for introducing a Multi–Objective EA (MOEA) for finding biclusters in gene expression data. Besides, by using a MOEA it will not be necessary to combine all the objectives into a single fitness function, which can be complicated, expecially given the fact that some of the objectives are interdependent.

The MOEA we propose in this paper is similar to the approach proposed in [3, 12]. As the algorithm partially uses the *mean squared residue*, the results have been compared to those of Cheng and Church. In expression data analysis, the most important goal may not be finding the maximum bicluster or even finding a bicluster covering for the data matrix. It is more interesting to find a set of genes showing strikingly similar up–regulation and down–regulation under a set of conditions. A low mean squared residue score plus a large variation from the constant may be a good criterion for identifying these genes and conditions. Therefore, our goal is to find biclusters of maximum size, with mean squared residue lower than a given δ, with a relatively high row variance, and with a low level of overlapping among biclusters.

The paper is organized as follows: in Section 2 the definitions related to biclustering are presented; Section 3 describes in detail the algorithm, while experimental results are discussed in Section 4, comparing the quality to those generated by Cheng and Church's algorithm. Finally, some conclusions are summarized in Section 5.

2. THE MODEL OF BICLUSTER

We follow the biclustering model proposed in [9]. A bicluster is defined on a gene–expression matrix. Let $G = \{g_1, \ldots, g_N\}$ be a set of genes and $C = \{c_1, \ldots, c_M\}$ a set of conditions. The data can be viewed as an $N \times M$ expression matrix EM of real numbers, with possible null values. Each $e_{ij} \in EM$ corresponds to the logarithm of the relative abundance of the mRNA of a gene g_i under a specific condition c_j.

A bicluster essentially corresponds to a sub–matrix that exhibits some coherent tendency. Each bicluster can be identified by a unique set of genes and conditions, that determine the sub–matrix. Thus a bicluster is a matrix $I \times J$, denoted as (I, J), where I and J are set of genes (rows) and conditions (columns), respectively, and $|I| \leq |N|$ and $|J| \leq |M|$. We define the volume of a bicluster (I, J) as the number of elements e_{ij} such that $i \in I$ and $j \in J$, i.e., $|I| \times |J|$.

Definition 1 Let (I, J) be a bicluster, then we define the base of a gene g_i as $e_{iJ} = \frac{\sum_{j \in J} e_{ij}}{|J|}$. In the same way we define the base of a condition c_j as $e_{Ij} = \frac{\sum_{i \in I} e_{ij}}{|I|}$. The base of a bicluster is the mean of all the entries contained in (I, J),

$$e_{IJ} = \frac{\sum_{i \in I, j \in J} e_{ij}}{|I| \cdot |J|}$$

Definition 2 The mean squared residue r_{IJ} of a bicluster (I, J) is defined as

$$r_{IJ} = \frac{\sum_{i \in I, j \in J} (e_{ij} - e_{iJ} - e_{Ij} + e_{IJ})^2}{|I| \cdot |J|}$$

The mean squared residue is an indicator of the degree of coherence of an element with respect to the remaining ones in the bicluster, given the tendency of the relevant gene and the relevant condition. The lower the mean squared residue, the stronger the coherence, and the better the quality of the bicluster. If a bicluster has a mean squared residue lower than a given value δ, then we call the bicluster a δ-bicluster. In addition to the mean squared residue, we may prefer the row variance to be relatively large to reject trivial bicluster, where the row variance is

$$\frac{\sum_{i \in I, j \in J} (e_{ij} - e_{iJ})^2}{|I| \cdot |J|}$$

Biclusters characterized by high values of row variance contains genes that present large changes in their expression values under different conditions. It follows that row variance can be used to guarantee that the bicluster captures genes exhibiting fluctuating yet coherent trends under some set of conditions.

```
SMOB(δ)
load Expression Matrix EM
repeat
   MOEB(EM,δ)
   if (bicluster returned)
      store bicluster in Results
      adjust weights of EM
   else
      end_cond met
   if (max_iter is reached)
      end_cond met
until end_cond is met
return Results
```

(a) Sequential Covering

Figure 1: A general scheme of the sequential covering algorithm SMOB.

3. THE ALGORITHM

The algorithm we propose in this paper is called SMOB (for Sequential Multi–Objective Biclustering) and is outlined in Figure 1. As in [3, 12], SMOB adopts a sequential covering strategy. The δ–biclusters returned are stored in a list, until the evolutionary algorithm is called a maximum number of times. δ is an user supplied parameter, as in [9].

We can individuate two main reasons that justify the use of a MOEA for finding biclusters. First, the problem of

finding biclusters in an expression matrix can be straightforwardly seen as a multi–objective problem. Second, with a MOEA, it is not necessary to combine the objectives in a single weighted fitness function. Finding a way to combine the objectives to be optimized in a single function can be problematic, and may require more parameters to set [11].

The objectives considered for being optimized are: the mean squared residue, the volume and the row variance. We want to minimize the mean squared residue, while the volume and the row variance have to be maximized.

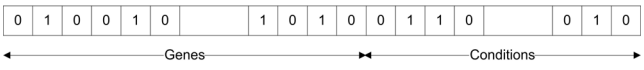

Figure 2: Encoding for the biclustering problem. In this example four genes and three conditions were chosen, so the potential bicluster has 12 elements.

The encoding of biclusters is the one proposed in [3, 12], where bit strings are used (see Figure 2). A bit is associated to each gene and each condition of the expression matrix. If a bit is set to one, it means that the relative gene/condition belongs to the bicluster, otherwise it does not.

Individuals are initialized in the following way. First the number of genes $|I|$ and of conditions $|J|$ contained in the biclusters are randomly determined. Then, $|I|$ bits corresponding to genes and $|J|$ bits corresponding to conditions are randomly selected. The selected bits are set to one, which means that the relative gene/condition is contained in the bicluster encoded by the individual. We perform this initialization instead of a pure random initialization of bit–strings because in that way the initial biclusters would contain all about the same number of genes and conditions.

The fitness $f(x)$ of an individual x is calculated on the basis of the Pareto dominance [14], namely $f(x)$ is based on the number of individuals x dominates. In order to establish if x dominates y, we use the mean squared residue, the volume and the row variance of x and y. However if x dominates n individuals, n is increased for each individual y dominated by x such that the mean squared residue of x is lower than y.

Moreover, in order to promote diversity in the population, two distance measures are used: one is calculated on the objective set and the other one on the decision set. The former is implemented by calculating the distance from the nearest neighbor in term of objectives, i.e., mean square residue, volume and row variance. The latter is the normalized average number of individuals covering the same elements of the expression matrix covered by the biclustering being evaluated. The inverses of these two distances are added to the fitness.

The fitness of an individual x is then given by

$$f(x) = \frac{1}{n} + \frac{1}{dist_{obj}} + \frac{1}{cov}$$

where n is computed as described above, $dist_{obj}$ is the distance considering the objectives and cov is the normalized average of individuals covering the elements of the expression matrix covered by x. Notice that the fitness has to be minimized. Biclusters with a mean squared residue higher than δ are penalized by adding to $f(x)$ the value $\frac{MSR(x)-\delta}{\delta}$, where $MSR(x)$ is the mean squared residue of x.

Individuals are selected with a tournament mechanism, with a tournament size of four. Three crossover operators are used with different probabilities: one-point crossover, two-point crossover and uniform crossover. The application of the uniform crossover is the one having the highest probability. Uniform crossover is preferred to one–point and two–point crossovers, as both would prohibit certain combinations of bits to be crossed over together [7].

Three mutation operators are used: a classical mutation operator, one that can add a row and one that can add a column. We consider columns and rows separately, because typically there are many more rows than columns, thus considering them together, would give more probability of mutation to rows than to columns.

Elitism is applied by letting the non–dominated individuals to survive to the next generation.

In order to avoid overlapping among biclusters, after each call of `MOEB`, we assign a weight to each element e_{ij} of the expression matrix. This weight w_{ij} is equal to the number of biclusters stored in the *Results* list (see Figure 1) that contain e_{ij}. When a bicluster x is evaluated inside `MOEB`, a penalty $P(x)$,

$$P(x) = 1 - \frac{V_x - \sum_{i,j \in x} w_{e_{ij}}}{V_x}$$

is added to the fitness of x, where V_x is the volume of x. In this way, if a bicluster has low volume and it covers elements of the expression matrix that are already contained in many biclusters already found, $P(x)$ will be high. On the other hand, if the bicluster has a high volume and it overlaps with few biclusters, the penalty will be lower. If the bicluster x does not overlap with any biclusters then $P(x)$ is zero.

4. EXPERIMENTS

In order to asses the goodness of the proposed method for finding biclusters in expression data, we conduct experiments on three well known datasets. The first dataset is the yeast *Saccharomyces cerevisiae* cell cycle expression dataset originated from [10]. The expression matrix contained in this dataset consists of 2884 genes and 17 conditions. For the yeast dataset δ was set to 300. The second dataset is the human B–cells expression data originated from [5]. The human dataset consists of an expression matrix of 4026 genes and 96 conditions. For the human dataset δ was set to 1200.

The two datasets are taken from [9], were the original data are preprocessed. The most important preprocessing operation regards missing values: missing values are replaced with random values, although it is known the existing risk that these random numbers can affect the discovery of biclusters [19]. The expectation was that these random values would not form recognizable patterns. These values of δ used in the two datasets are taken from [9].

The third dataset is the *Colon Cancer* dataset. This dataset originated from [10], and it contains an expression matrix consisting of 2000 genes and 62 conditions. This dataset was preprocessed as in [9], where each entry x of the original dataset was substituted by the value $100 \cdot log(10^5 \cdot x)$. For this dataset the value of δ was set to 500, becasue the expression matrix contained in the dataset has a size that is about the double of that contained in the yeast dataset.

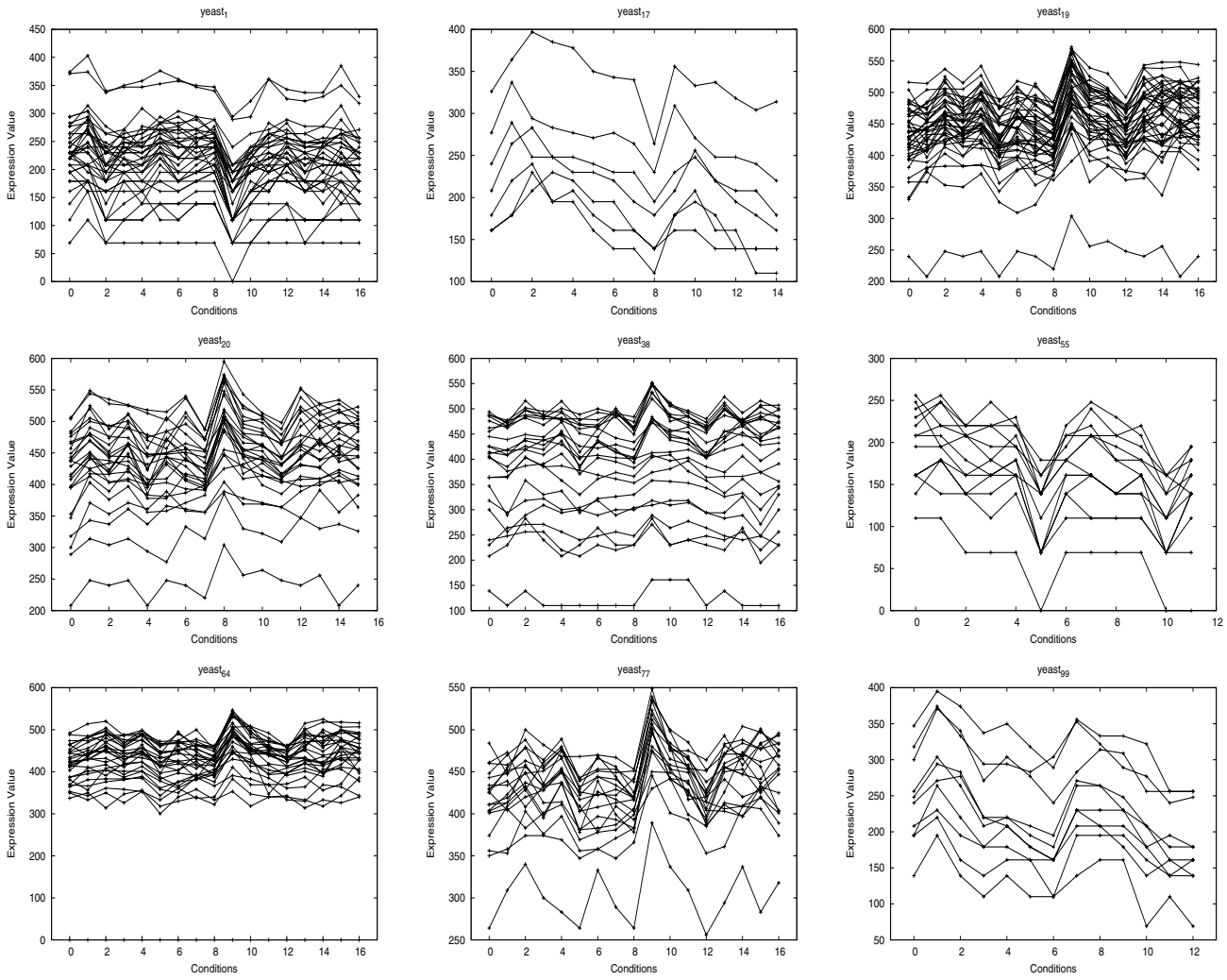

Figure 3: Nine biclusters found on the yeast dataset.

A similar reasoning was adopted in [9] for determining the values of δ for some datasets.

In these experiments we used the default parameters of the evolutionary algorithms, which have proven to be effective when used in the single–objective EA on which SMOB is based on (see [12]). These parameters are shown in Table 1.

Table 1: Standard parameter values for SMOB.

Parameter	Value
Generations	100
Population size	200
Crossover probability	0.85
Mutation probability	0.2

In Figure 3 nine biclusters out of the one hundred found on the yeast dataset are shown. Biclusters 77 and 19 are particularly interesting. They present a similar behavior, but they only have two genes in common. It is interesting to notice that in both biclusters a gene assume a behavior similar to that of all the other genes, even if its level of expression is much lower than the others on all the conditions. In both biclusters the expression values of all the genes increase in unison under the ninth condition. A similar behavior can be noticed also in bicluster 20, where two genes assume lower expression levels than that assumed by the main group of genes on all the conditions. Nevertheless the behavior of all the genes is similar under all the sixteen conditions contained in the bicluster. Bicluster 99 is also interesting. It contains ten genes showing strikingly up–regulation and down–regulation under thirteen conditions. This similar behavior is also highlighted by the high row variance characterizing this bicluster. Another interesting fact to notice is that the first bicluster found by SMOB (bicluster 1) is not as flat as the first biclusters found by SEBI. This confirms the validity of a multi–objective approach to the biclustering problem. Information about the biclusters shown in Figure 3 is summarized in Table 2. The one hundred biclusters found on the yeast dataset cover 47.02% of the gens, 100% of the conditions, and in total 40.39% of the elements of the expression matrix.

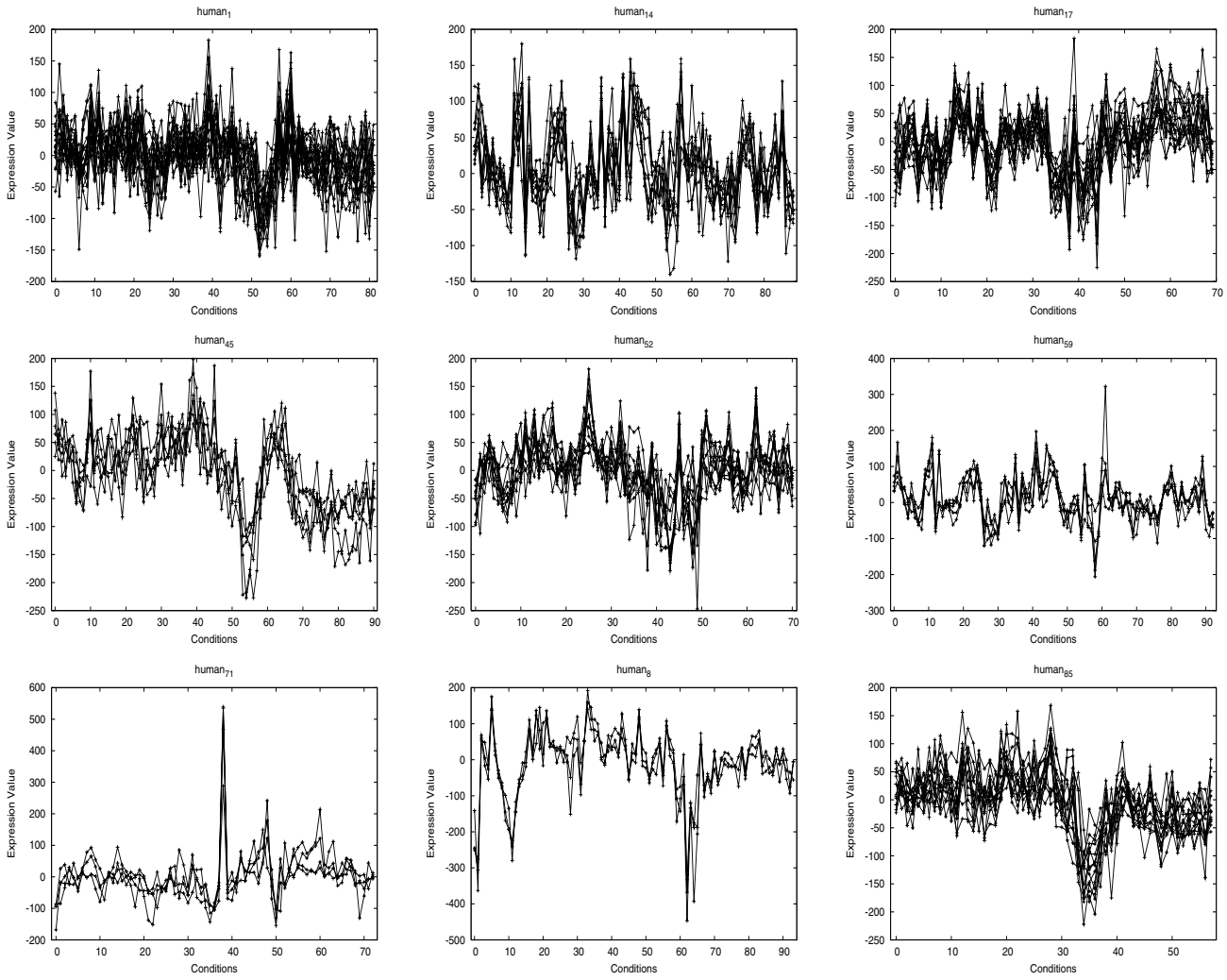

Figure 4: Nine biclusters found on the human lymphoma dataset.

Table 2: Information regarding the biclusters found on the yeast dataset shown in Figure 3.

Bicluster	Genes	Conditions	Residue	Row Variance
1	36	17	216.59	629.68
17	7	15	206.61	979.39
19	41	17	205.21	623.70
20	26	16	205.36	600.99
38	23	17	201.92	371.72
55	13	12	201.91	996.74
64	29	17	211.69	450.95
77	19	17	202.18	701.68
99	10	13	203.06	1248.97

According to the characterization of shifting and scaling patterns in biclusters described in [2], biclusters 17, 55 and 99 are particularly interesting due to the similar behavior of genes under several conditions (15, 12 and 13, respectively) and the wide range of levels of expression (aprox. 300, 250 and 350, respectively) for few genes (7, 13 and 10, respectively), what produces excellent results of row variance (aprox. 979, 996 and 1248, respectively). Anyway, it is important to note, as it is pointed out in [2], that a mean squared residue of about 200 might also indicate that some scaling factor is present in the biclusters.

Figure 4 shows nine biclusters out of one hundred found by `SMOB` on the human dataset. Information regarding these biclusters is given in Table 3. All the biclusters shown contain genes that behave in a very similar way. Bicluster 8 is particularly interesting. It is characterized by its very high row variance, together with very small mean squared residue. This is reflected in the very similar changes in expression level assumed by the three genes under the ninety–four conditions of the bicluster. Another interesting bicluster is the number 71. In this case, it can be noticed that under one

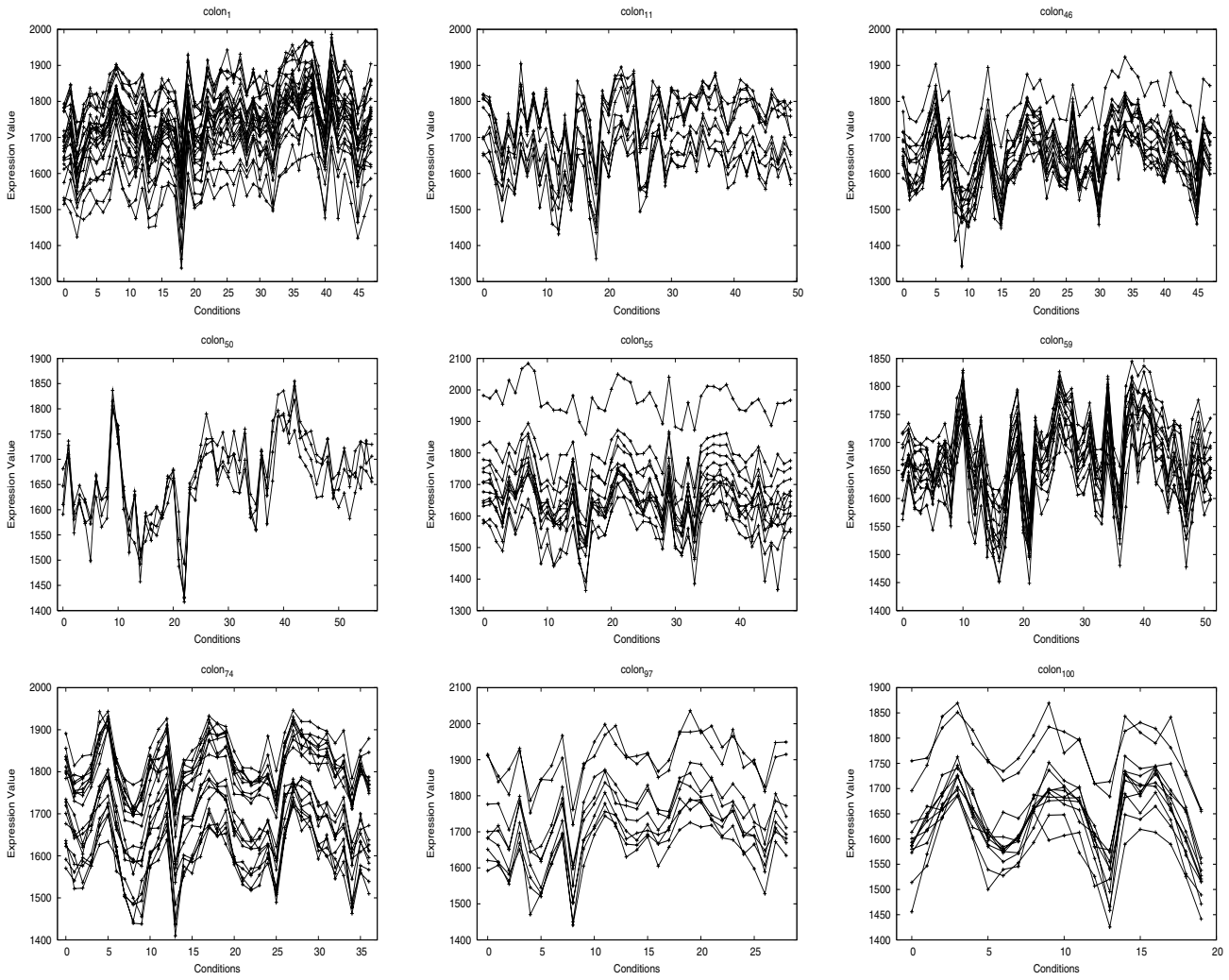

Figure 5: Nine biclusters found on the colon cancer dataset.

Table 3: Information regarding the biclusters found on the human dataset shown in Figure 4.

Bicluster	Genes	Conditions	Residue	Row Variance
1	21	82	1175.08	2028.91
14	10	89	967.74	2784.85
17	17	70	1167.99	2945.48
45	6	91	1155.33	4931.82
52	12	71	1147.07	2806.31
59	5	93	851.08	3725.49
71	4	73	1111.24	6082.14
8	3	94	688.34	9067.13
85	15	58	1148.79	3156.68

condition, all the genes have an very high increment in their expression levels, while for the rest of the conditions they behave in a similar way. Bicluster 17 contains seventeen genes that oscillate in the same way under all the 70 conditions contained in the bicluster. For this dataset, there is no evidence of clear shifting patterns. The one hundred biclusters found on the human dataset covered 33.52% of the elements of the expression matrix (45.05% of the genes and 100% of the conditions).

Nine biclusters found on the colon datasets are reported in Figure 5. Infomation about these biclusters is shown in Table 4.

Also in these biclusters, shifting patterns can be clearly noticed in biclusters 100, 97, 74, 46 and 55. In particular, in bicluster 100, two groups of genes, one consisting of two gens and the other of the remaining ten genes, can be noticed. The two group of genes have different magnitude of expression level, but nevertheless they show the same behavior under the twenty conditions contained in the bicluster. Bicluster 74 presents shifting patterns, where three group of genes can be noticed to have the same up and down regulations

Table 5: Performance comparison between SMOB, SEBI and CC. The third column reports the average mean squared residue of the 100 biclusters found on each dataset, the fourth column reports the average volume, while the fifth and the sixth columns report the average number of genes and conditions contained in the biclusters, respectively. Standard deviation is reported between parentheses.

	Dataset	Avg. residue	Avg. Volume	Avg. genes	Avg. cond.
SMOB	Yeast	206.17 (15.82)	453.48 (231.76)	27.28 (14.88)	15.46 (1.88)
	Human	1019.16 (120.78)	709.13 (378.05)	11.60 (12.55)	78.47 (19.46)
	Colon	472.01 (20.94)	658.96 (376.53)	15.45 (12.42)	48.93 (11.01)
SEBI	Yeast	205.18 (4.49)	209.92 (171.39)	13.61 (10.38)	15.25 (1.37)
	Human	1028.84 (29.19)	615.84 (278.35)	14.07 (5.39)	43.57 (6.20)
	Colon	492.46 (6.23)	403.48 (215.70)	9.86 (4.51)	40.91 (8.00)
CC	Yeast	204.29 (42.78)	1576.98 (2178.46)	166.71 (226.37)	12.09 (4.39)
	Human	850.04 (153.91)	4595.98 (3353.72)	269.22 (204.71)	24.5 (20.92)

Table 4: Information regarding the biclusters found on the colon dataset shown in Figure 5.

Bicluster	Genes	Conditions	Residue	Row Variance
1	26	48	482.85	3889.90
11	10	50	485.00	4661.39
46	15	48	453.26	4794.21
50	4	57	492.40	6389.89
55	13	49	469.85	3748.42
59	15	52	474.98	4133.75
74	19	37	454.95	3839.95
97	9	29	446.52	4874.55
100	12	20	467.60	3919.60

under the thirty–seven conditions included in the bicluster (this result shows the good performance of multi–objective evolutionary computation on this task, as this bicluster was not previously identified by other approaches). Bicluster 50 is also particularly interesting. It contains four genes showing a very similar behavior under fifty–seven conditions. This fact is reflected by the high row variance characterizing this bicluster. Also in the case of the colon dataset, the first bicluster found (labelled 1 in Figure 5) is very interesting. It contains twenty–six genes that behave in a very similar way under forty–eight conditions. The one-hundred biclusters found on the colon dataset cover 45.05% of the genes, 100% of the conditions for a total of 33.53% of the elements of the expression matrix.

In order to analyze the behavior of the multi–objective evolutionary algorithm, we have also conducted experiments with a single–objective evolutionary algorithm (SEBI) and a non–evolutionary algorithm (CC). Table 5 shows a comparison between the results obtained by SMOB and the results obtained by SEBI [3] and CC [9].

As far as the yeast dataset is concerned, in general the three algorithms obtained similar results. The results obtained by SMOB and SEBI are usually more stable. CC could find bigger biclusters. However, the first biclusters found by CC are huge, although not very interesting, since they are characterized by a very low row variance, i.e., they are "flat". Instead, we have seen that the first bicluster found by SMOB is interesting, and this is not true only for the yeast dataset, but also for the other datasets used in the experimentation carried out in this paper.

The results obtained on the human dataset by SMOB are better than those obtained by SEBI. SMOB found biclusters characterized by a slightly lower mean squared residue and with higher volumes. In general, SMOB and SEBI find smaller biclusters, partially due to the overlapping policy adopted by them. On the colon dataset, SMOB again performed better than SEBI. The biclusters found have lower mean squared residue and much higher volume. We have seen that the biclusters found by SMOB are very interesting on this dataset, including the first biclusters, unlike CC, where some uninteresting biclusters have to be found before interesting biclusters are discovered [9].

We believe that the overlapping policy adopted by both SMOB and SEBI is more appropriate than the one adopted by CC. In fact, after a bicluster is found, CC substitutes the values of the entries covered by the bicluster with random values. It is known the existing risk that these random numbers can affect the quality of further biclusters discovered by the algorithm [19].

In short, SMOB finds more stable biclusters, including clearly some relevant shifting patterns, which reveal the co–regulation of groups of genes under subsets of experimental conditions.

5. CONCLUSIONS

In this paper we have proposed a multi–objective evolutionary algorithm, called SMOB, for finding biclusters in gene expression data.

The algorithm is based on the single–objective algorithm, called SEBI, introduced in [3, 12]. In that algorithm, the fitness of an individual encoding one bicluster was based on a weighted sum of the mean squared residue, the volume and the row variance. All these three objectives are to be optimized, and they are in conflict with each other. Thus, a multi–objective approach seems a natural solution for the problem of finding biclusters in gene expression data. Moreover, by using a multi–objective algorithm, the definition of a fitness, based on the Pareto front, is straightforward, and it reduces at the same time the number of parameters of the algorithm.

Experimental results show that SMOB is able to find interesting biclusters on expression data. The biclusters found by SMOB improve the results obtained by SEBI, both in terms of volume and of mean squared residue. It is worth noticing that the first biclusters found by CC are usually not very significant, while even the first biclusters found by SMOB are

interesting, and that the biclusters found by SMOB are, in general, characterized by a higher row variance.

As a future development of SMOB, we are considering to implement a version of the algorithm by using a specific representation, named *natural encoding* [4], which seems to be very useful for this problem.

Acknowledgement

This research was supported by the Spanish Research Agency CICYT under grants TIN 2004-00159 and TIN2004-06689C0 303.

6. REFERENCES

[1] A. Tanay and R. Sharan and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 19 (Sup. 2):196–205, 2002.

[2] J. S. Aguilar-Ruiz. Shifting and scaling patterns from gene expression data. *Bioinformatics*, 21:3840–3845, 2005.

[3] J. S. Aguilar-Ruiz and F. Divina. Evolutionary biclustering of microarray data. In *Proceedings of the 3rd European Workshop on Evolutionary Compuation and Bioinformatics*, Lecture Notes in Computer Science, pages 1–10. Springer-Verlag, 2005.

[4] J. S. Aguilar-Ruiz, R. Giraldez, and J. C. Riquelme. Natural encoding for evolutionary supervised learning. *IEEE Transactions on Evolutionary Computation*, page in press, 2007.

[5] A. A. Alizadeh and et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[6] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data: the order-preserving submatrix problem. In *RECOMB '02: Proceedings of the sixth annual international conference on Computational biology*, pages 49–57, New York, NY, USA, 2002. ACM Press.

[7] 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.

[8] S. Bleuler and E. Zitzler. Order preserving clustering over multiple time course experiments. In *EvoWorkshops 2005*, number 3449 in Lecture Notes in Computer Science, pages 33–43. Springer-Verlag, 2005.

[9] Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the 8th International Conference on Itelligent Systems for Molecular Biology (ISMB'00)*, pages 93–103, 2000.

[10] R. Cho, M. Campbell, E. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. Wolfsberg, A. Gabrielian, D. Landsman, D. Lockhart, and R. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2:65–73, 1998.

[11] D. Corne, K. Deb, and P. J. Fleming. The good of the many outweighs the good of the one: evolutionary multi-objective optimization. *IEEE Connections Newsletter*, 1(1):9–13, February 2003.

[12] 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.

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

[14] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[15] J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

[16] L. Lazzeroni and A. Owen. Plaid models for gene expression data. Technical report, Stanford University, 2000.

[17] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 1(1):24–45, 2004.

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

[19] J. Yang, W. Wang, H. Wang, and P. S. Yu. δ–clusters: Capturing subspace correlation in a large data set. In *Proceedings of the 18th IEEE Conference on Data Engineering*, pages 517–528, 2002.

[20] J. Yang, W. Wang, H. Wang, and P. S. Yu. Enhanced biclustering on expression data. In *Proceedings of the 3rd IEEE Conference on Bioinformatics and Bioengineering*, pages 321–327, 2003.