Shifting Patterns Discovery in Microarrays with Evolutionary Algorithms

Beatriz Pontes[1], 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] Area of Computer Science, University of Pablo de Olavide
Ctra. de Utrera, km. 1, 41013, Sevilla, Spain
{rgirroj, sjsagurui}@upo.es

Abstract. In recent years, the interest in extracting useful knowledge from gene expression data has experimented an enormous increase with the development of microarray technique. Biclustering is a recent technique that aims at extracting a subset of genes that show a similar behaviour for a subset conditions. It is important, therefore, to measure the quality of a bicluster, and a way to do that would be checking if each data submatrix follows a specific trend, represented by a pattern. In this work, we present an evolutionary algorithm for finding significant shifting patterns which depict the general behaviour within each bicluster. The empirical results we have obtained confirm the quality of our proposal, obtaining very accurate solutions for the biclusters used.

Keywords: Gene Expression Data, Biclustering, Evolutionary Algorithm, Shifting Pattern.

1 Introduction

Microarray data are widely used due to the great potential in different biomedical fields as gene expression profiling, facilitating the prognosis and the discovering of subtypes of diseases. A microarray is a set of DNA/RNA sequences, where the gene expression data are organized in a two-dimensional array. Columns represent genes and rows represent experimental conditions, so that, each element in the matrix refers to the expression level of a particular gen under specific conditions.

In order to extract relevant knowledge from microarray expression data, clustering techniques have been applied [4]. The main application of this techniques is to group genes together according to any specific algorithm or mathematical formula related to their functional similarities over all conditions. However, relevant genes are not necessarily related to every condition [15]. Thus, biclustering [12] is a variation of clustering where the process consist of simultaneously

* This research was supported by the Spanish Research Agency CICYT under grants TIN2004–00159 and TIN2004-06689C0303.

mining columns and rows of the matrix. In the context of microarrays study, it is applied to identify groups of genes which exhibit similar behaviour under a specific subset of experimental conditions [8]. Bicluster analysis [14] takes into account the fact that not every gene in a microarray may be relevant for all the conditions, thus addressing in the two dimensions simultaneously the clustering problem. Biclustering methods for biological data analysis have been widely studied in the literature [5,6,13].

In [8], Cheng and Church showed that some biclusters should contain a subset of genes showing similar behaviour and not necessarily similar values, or in other words, such genes could follow a pattern of behaviour. Thus, two types of patterns [1], such as shifting and scaling patterns, should be found in biclusters. These patterns can be very useful for different aspects as to find more genes or conditions that should be included in a bicluster, or simply to describe the common conduct of the genes belonging to a certain bicluster.

In this work, we address the finding pattern problem with Evolutionary Algorithms (AE), which has been proven to have an excellent performance on highly complex optimization problems. Thus, we present a new EA-based tool for finding the shifting patterns which represents more accurately the behaviour of the genes in a given bicluster. The experimental results show that our approach obtains shifting patterns with an excellent performance.

The paper is organized as follows: in Section 2, an overview on patterns from gene expression data is presented. We provide a description of our algorithm in Section 3 and the experimental results are shown in Section 4. Finally, the last section summarises the main conclusions of this work.

2 Patterns from Biclustering in Microarrays

The genes included in a bicluster could follow a pattern of behaviour [8]. This idea was formally described in [1], where two kind of patterns were defined.

Let $\mathcal{M}$ be a microarray with N rows (conditions c_i, with $1 \leq i \leq N$) and M conditions (genes g_j, with $1 \leq j \leq M$). Each element in the matrix will be represented as $v_{ij} \in \mathcal{M}$. Also, let $\mathcal{B} \subseteq \mathcal{M}$ be a bicluster made up of $n \leq N$ conditions and $m \leq M$ genes. Each element in the bicluster will be represented as $w_{ij} \in \mathcal{B}$. With these premises, shifting and scaling patterns are defined as follows [1]:

A bicluster $\mathcal{B}$ shows a *shifting pattern* when the values w_{ij} can be obtained by adding a certain value β_i, constant for the i^{th} condition, to a typical value (π_j) for the j^{th} gene. Analogously, the definition of scaling pattern is similar to the scaling by replacing the additive factor β_i with multiplicative value α_i. Formally, a bicluster follows a shifting pattern (Equation 1) or a scaling pattern (Equation 2) when it follows the expressions:

$$w_{ij} = \pi_j + \beta_i + \xi_{ij} \tag{1}$$

$$w_{ij} = \pi_j \times \alpha_i + \xi_{ij} \tag{2}$$

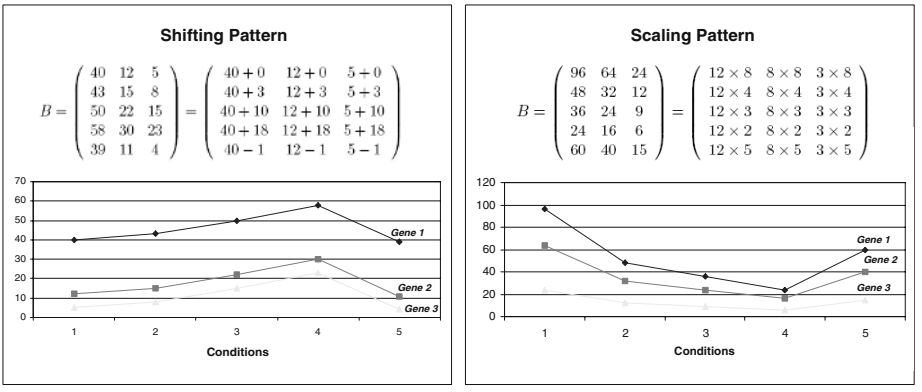

Fig. 1. Examples of biclusters with shifting and scaling paterns

where w_{ij} is the value for gen j and condition i within a bicluster; π_j is the fixed value for j^{th} gene; β_i in Equation 1 is the shifting value for condition i; α_i is the scaling factor for i^{th} condition in Equation 2; finally, ξ_{ij} is the error that the pattern makes for w_{ij} value. In both cases, when such error is 0 for all bicluster's values, we say we have a *perfect bicluster*.

In order to illustrate these definitions, Figure 1 shows an example of two biclusters that follow a perfect shifting pattern (on the left) and a perfect scaling pattern (on the right). In the shifting case, if we represent all the values for each gene, all the charts have the same shape and slope, but in a different range (they are parallel). However, in the scaling case, all the charts have similar shape but different slopes.

In this work we only propose an algorithm for finding shifting patterns, although both shifting and scaling patterns can be present in the data matrix simultaneously. In any case, we are working in order to suggest a scaling approach for future works.

3 Algorithm

A family of computational techniques inspired by the concept of evolution is known as Evolutionary Algorithms (EAs). These algorithms find the solutions to a particular problem by applying a random search on a set of possible solutions [7,11]. EAs use a finite subset of the search space, called population, in each iteration. Previously, these possible solutions were encoded according to the selected coding. The coding is the internal representation of the search space that the algorithm uses. Each encoded element of the population is an individual. Thus, beginning by a pseudo-randomly generated initial population, the evolutionary algorithm selects some individuals and recombine them to generate a new generation of individuals. This process is repeated for a number of generations until the algorithm converges. The selection of individuals is carried out according to their fitness, that is a measurement of the quality of each individual with regards

to the remaining ones. The process of calculating the fitness of the individuals is called evaluation. The evaluation consists in assigning a fitness value to every individual by applying a fitness function.

As aforementioned, our goal is to find the best shifting pattern which represents the general trend within a bicluster. In this work, we address this problem with EAs. Thus, we propose an algorithm which takes as inputs a bicluster and various configuration parameters, returning a set of β_i values (henceforth *beta set*) for such bicluster.

Each chromosome or individual ($\mathcal{I}$) is made up of a set of real numbers that represent the beta values in real coding ($\mathcal{I} = \{\beta_1, \beta_2, \ldots, \beta_i, \ldots, \beta_n\}$), corresponding to a shifting pattern proposal. All the individuals have the same length and equal to the number of conditions in the bicluster.

The initial population can be built in two different ways, generating the initial solutions randomly or by using an algorithm based on mutations of the values of the bicluster. For the experiments we present in this work, the second option has been used. In each iteration, each individual of the population is evaluated according to the fitness function defined in Equation 3, and based on the *Mean Absolute Error* (MAE) of each individual, that is, the mean of $|\xi_{ij}|$ values (Equation 4). We have also implemented other alternatives for the fitness function such as the mean squared error, but the obtained results were similar.

$$\phi(\mathcal{I}) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} \xi_{ij}}{n \times m} \tag{3}$$

$$\xi_{ij} = w_{ij} - \pi_j - \beta_i \tag{4}$$

At the end of each generation, the best individual is replicated to the next one (elitism). Later, a set of individuals (the number of individuals is given by the replication percentage) are selected through the roulette wheel method [10] and replicated to the next generation. Afterwards, the use of recombination and mutation operators allow us to combine a percentage of solutions selected by the roulette wheel for producing new individuals [11]. These operators modify the individuals in a random way. Crossover operator takes as input the number of points for the recombination and create offspring by exchanging the substrings of both parents, thus producing individuals in which the beta values are from both of them. The mutation operator is applied to each individual depending on the mutation probability. Whenever a solution is chosen for a mutation, a random beta value is selected for being changed. The new value is calculated by adding a value between zero and the mean of the error values committed for this beta. Note that this mean can be either positive or negative, depending on the range of the values in the bicluster. After a preset number of generations, the algorithm return the best found beta set.

4 Experimental Results

To show the quality of our tool, we conducted experiments on the biclusters obtained in previous work [2]. These biclusters were obtained by means of an

Table 1. Parameters values of the EA

Parameter	Value
Population size	100
Number of generations	100
Crossover probability	0.80
Mutation probability	0.50
Replication probability	0.20
Number of points in crosses	2

EA from two well-known datasets: yeast *Saccharomyces cerevisiae* cell cycle expression dataset [9]; and the human B-cells expression data [3]. In this section, we expose the empirical results obtained. In Table 1 the parameter settings for all the experiments presented here are shown.

The algorithm was applied with several kinds of biclusters. Thus, for instance, there are biclusters containing different number of genes and conditions or showing different grades of shifting behaviour. Of course, as all of them are obtained from real data, no of them exhibit a perfect shifting pattern, manifesting also scaling trends. In the case of finding a shifting trend within a perfect bicluster, the tool will perform an error value equal to zero in the first iteration.

We have performed our approach over all the biclusters obtained in [2], displaying here the most relevant results. While testing the algorithm, several parameters configuration were used, presenting in this work the ones which performed more interesting results. In all cases we have obtained a set of beta values corresponding to the best found shifting pattern.

4.1 Yeast Dataset

The graphics for five yeast biclusters are represented in Figure 2. In this figure, we expose two charts for each bicluster, the original bicluster (on the left) and the pattern shifted to the range of each of the genes (on the right). We can appreciate how the quality of the found pattern depends on the shifting trend followed by the bicluster. In general terms, we could say that the pattern tries to uniform the behaviour, thus ignoring some isolated local shapes. For instance, in the bicluster labeled 99_1, the global trend has been perfectly simulated by the result pattern.

Note that the less uniform the genes are, the worse the pattern will be. It means that if we run the algorithm on a bicluster with these characteristic, the shape of the pattern could be very different from some of the genes shapes. Nevertheless, a great number of genes does not implies a bad quality of the found pattern (see Figure 2, bicluster 1_1). Another important characteristic of our algorithm is its rapid convergence; for almost every bicluster in the data set we have experimented with, all of them present a similar convergence throughout the generations. The bicluster labeled 64_1 has the best final error value ($MAE = 10.8$), meaning that the pattern our approach has found for this bicluster is closer to the behaviour of the genes than the pattern in other biclusters. The worst value

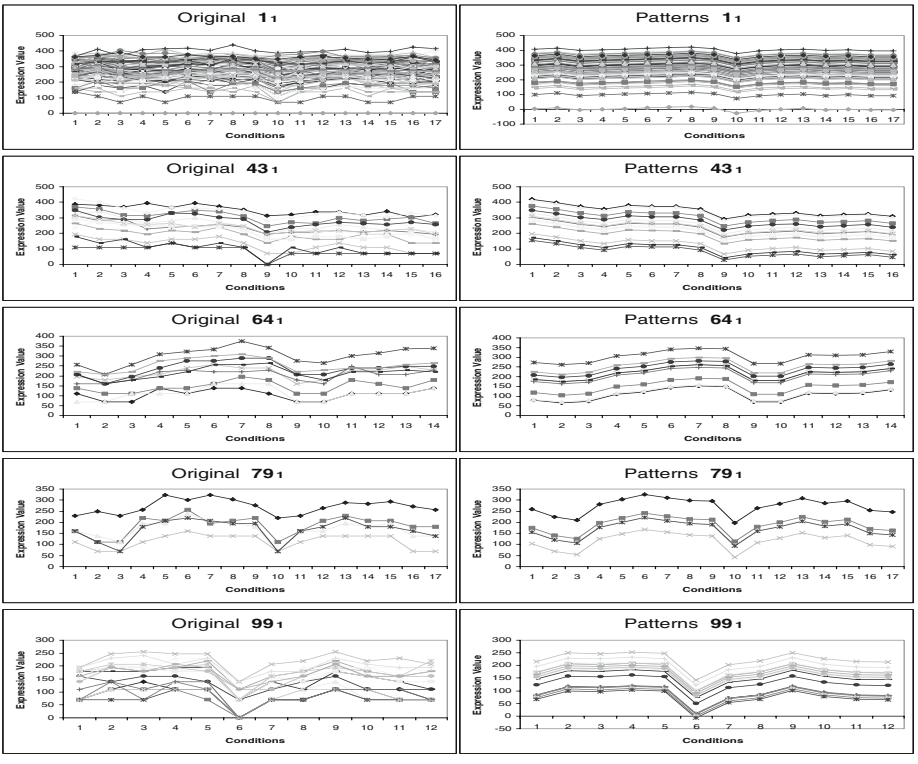

Fig. 2. Yeast biclusters analysed and their pattern result

of the fitness function in the last iteration is for bicluster 79_1 ($MAE = 11.7$), due to this bicluster has few genes but they are quite different one from each other. However, the differences among the error rates are not significant.

4.2 Human Dataset

Five out of hundred bicluster analysed are shown in Figure 3. This figure have a similar structure that the previous one. There exists some differences from the case of the yeast dataset. One point is that now the data include non-positive values, although it makes no difference for our method. But another issue is that the biclusters of the human dataset contain much more conditions. From this point of view, we could expect the fitness function values to be worse than for the previous dataset. Furthermore, the genes represented here are closer than in the previous case, thus the result patterns are closer too, as we can easily appreciate comparing the results in Figures 2 and 3, where we can see how the range of the outcome patterns is bigger in the first case.

Figure 3 shown different kinds of biclusters. For instance, the one labeled 101_1 contains only 3 genes but the mayor number of conditions (72). A medium-length bicluster would be 50_1, which is made up of 11 genes and 58 conditions.

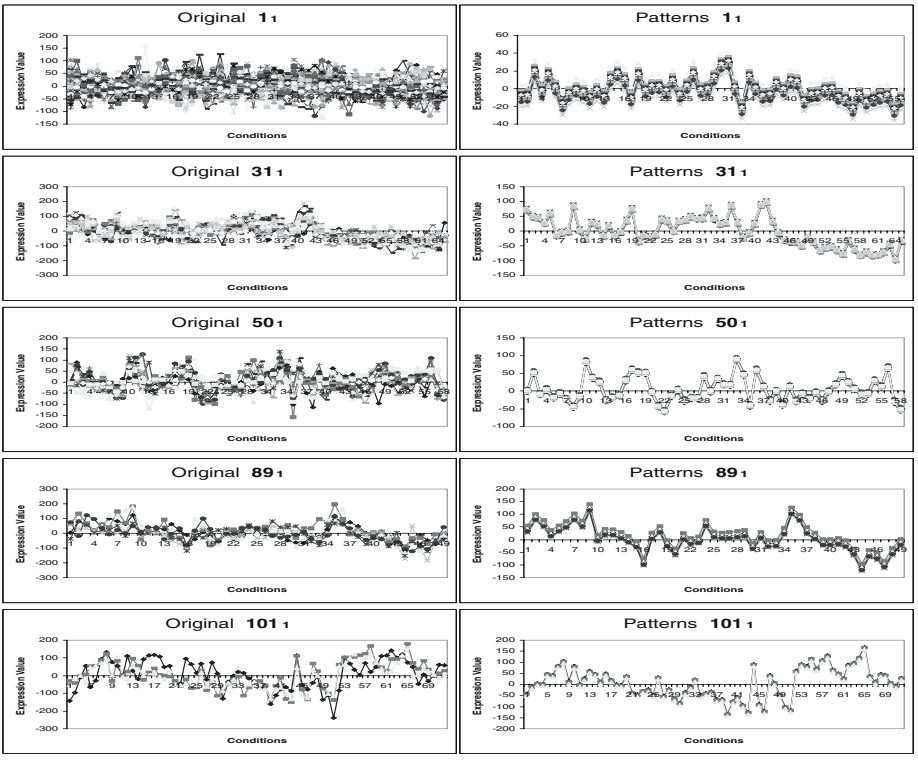

Fig. 3. Human biclusters analysed and their pattern result

Nevertheless, the final error value depends on the quality of each bicluster and not on its size. As we had predicted, the error values are greater than in the case of the yeast dataset, due mainly to the great number of conditions. However, although most of the biclusters tested by the tool show a similar fitness function behaviour (the best MAE was 24.7 for 101_1, and the worst was 27.4 for 31_1), the convergence is not so quickly as for the yeast dataset, although an established value has almost been reach in the last iterations.

5 Conclusions

This work has been developed on the idea that every gene in some types of biclusters follows a similar behaviour, and their graphical representations follow a similar trend with similar slopes. This behaviour is called shifting patterns. In this paper we have presented a novel EA-based tool capable of finding shifting patterns representing the general trend within a bicluster. Beginning from a given bicluster, our approach applies a typical EA to obtain the β_i coefficients that define the pattern. Experimental results over hundred of samples confirm

the quality of our approach for finding this kind of patterns, obtaining very accurate solutions for the biclusters used.

Future works will focus on finding both shitting and scaling patterns simultaneously. A first approach to the scaling problem would consist of considering it as a shifting problem, using the properties of the logarithms.

References

1. J. S. Aguilar-Ruiz. Shifting and scaling patterns from gene expression data. *Bioinformatics*, 21:3840–3845, 2005.
2. J. S. Aguilar-Ruiz and F. Divina. Biclustering of expression data with evolutionary computation. *IEEE Transactions on Knowledge & Data Engineering*, to be published.
3. A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D.Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt. 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. pages 166–173, Piscataway, NJ, 2000.
6. K. Bryan, P. Cunningham, and N. Bolshakova. Biclustering of expression data using simulated annealing. In *18th IEEE Symposium on Computer-Based Medical Systems*, pages 383–388, Dublin, Ireland, 2005.
7. L. D. Chambers et al. *Practical Handbook of Genetic Algorithms, volume III*. CRC Press, 1999.
8. 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.
9. 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.
10. K. A. DeJong. *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, 1975.
11. D. E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison Wesley, 1989.
12. J. A. Hartigan. Direct clustering of a data matrix. *ournal of the American Statistical Association*, 67(337):123–129, 1972.
13. 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.
14. A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18:136–144, 2002.
15. 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.