2009 Ninth International Conference on Intelligent Systems Design and Applications

An Overlapping Control–Biclustering Algorithm from Gene Expression Data

Juan A. Nepomuceno
Department of Computer Science
University of Sevilla, Spain
janepo@us.es

Alicia Troncoso Jesús S. Aguilar-Ruiz
Area of Computer Science
Pablo de Olavide University of Sevilla, Spain
ali@upo.es, aguilar@upo.es

Abstract

In this paper a hybrid metaheuristic for biclustering based on Scatter Search and Genetic Algorithms is presented. A general scheme of Scatter Search has been used to obtain high–quality biclusters, but a way of generating the initial population and a method of combination based on Genetic Algorithms have been chosen. Moreover, in the own algorithm the overlapping among biclusters is controlled adding a penalization term in the fitness function. Experimental results from yeast cell cycle are reported. Finally, the performance of the proposed hybrid algorithm is compared with a genetic algorithm recently published.

1 Introduction

Clustering and data mining techniques have been recently applied to analyze the huge volume of biological information generated by microarray data experiments [10]. Clustering techniques find groups of genes with similar behavior from a microarray. However, genes are not necessary related to every condition. Thus, the goal of the biclustering is to identify genes with the same pattern only under a specific group of conditions.

Many approaches have been proposed for biclustering in the context of microarray analysis [3]. Biclustering algorithms have two important aspects: the search algorithm and the measure to evaluate the quality of biclusters.

Most of proposed approaches are focussed on different search methods. An iterative hierarchical clustering is applied to each dimension separately and biclusters are built by the combination of the obtained results for each dimension in [7].

The Cheng and Church algorithm [4] built biclusters adding or removing genes or conditions in order to improve the measure of quality called Mean Squared Residue (MSR). In [15], an exhaustive biclusters enumeration by means of a bipartite graph-based model in which nodes were added o removed in order to find subgraphs with maximum weights. The FLOC algorithm [16] improved the method presented in [4] obtaining a set of biclusters simultaneously and adding missing values techniques. In [1], a simple linear model for gene expression was applied assuming normally distributed expression level for each gene or condition. Also, geometrical characterizations such as hyperplanes in a high dimensional data space have been used to find biclusters [8]. Recently, global optimization techniques such as Simulated Annealing [2] or Evolutionary Computation [6, 11] have been applied to obtain biclusters due to the good performance shown in several environments.

In the last few years, several papers were focussed on the measure proposed to evaluate the quality of biclusters. In [9] an analysis of the MSR was made, showing that this measure is good to find biclusters with shifting patterns but not scaling patterns. A new measure based on unconstrained optimization techniques was proposed in [12] as alternative to the MSR in order to find biclusters with certain patterns.

In this paper a biclustering algorithm, which incorporates a control of the overlapping among biclusters, based on the evolution of populations is presented. The proposed algorithm combines Scatter Search with some features of the Genetic Algorithms such as the way of generating the initial population and the offspring. Finally, the performance of the proposed methodology is compared with a genetic algorithm recently published [6] and with the Cheng and Church algorithm [4]. A Scatter Search has been selected due to the recent success obtained to solve different hard optimization problems and to the references about the application of Scatter Search for biclustering have not been found in the literature.

This paper is organized as follows. Section 2 presents basic concepts about populations–based algorithms focussing on Scatter Search. The description of the proposed method is described in Section 3. Some experimental results from a real dataset and a comparison between the proposed method and two biclustering algorithms are reported in Section 4. Finally, Section 5 outlines the main conclusions of the pa-

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.241

per and future works.

2 Populations-Based Algorithms: Scatter Search

Search strategies based on the evolution of populations are optimization techniques where a set of individuals codifying possible solutions evolves in order to find an optimal solution of the problem.

The proposed approach in this work, called OC-SS&GA, combines two evolutionary algorithms based on populations: Scatter Search and Genetic Algorithms. Scatter Search [14] was introduced in the seventies and recently it has been applied to many nonlinear and combinatorial optimization problems. Basically, a standard Scatter Search can be summarized by the following steps:

1. Generate an initial population in a deterministic manner to assure the diversity of the population regarding a distance.
2. A reference set is built with the best individuals from this initial population. The best individuals are not limited to a measure of quality provided by a fitness function but an individual that improves the diversity can be added to this reference set.
3. New individuals are created by the deterministic combination of individuals of the reference set and all individuals of the reference set are selected to be combined.
4. The reference set is updated using the new individuals and the combination is repeated until the reference set does not change.
5. The reference set is rebuilt and if the maximum number of iterations is not reached go to step 3.

The main ideas in Scatter Search are the diversification in order to avoid local minima and the intensification in order to find high–quality solutions. The diversity is introduced when the population is generated initially and when the reference set is rebuilt. The intensification is due to the combination method and the selection of the best solutions.

Genetic Algorithms differs to Scatter Search in some aspects such as the way of generating the initial population randomly, the selection of individuals to create offspring where a probabilistic procedure is applied to select parents, the evolution of the population which is based on the survival of the best depending only on the fitness function, the way of generating diversity using mutation operators and, mainly, the size of the population in Genetic Algorithms is bigger than that of the reference set in Scatter Search. A typical size in Genetic Algorithms is 100 and 10 in Scatter Search as the combination method in Scatter Search takes into account all pairs of individuals to create new individuals.

The underlying idea of Scatter Search is to emphasize systematic processes against existing random procedures in Genetic Algorithms.

3 Description of the Algorithm

In this section the pseudocode of the OC-SS&GA biclustering method is presented in Algorithm 1. Basically, the algorithm is a hybrid metaheuristic based on Scatter Search and Genetic Algorithms. High-quality biclusters are obtained by a general Scatter Search but the generation of the initial population and the combination method are caught from Genetic Algorithms. Furthermore, the algorithm avoid the overlapping among biclusters including in the fitness function penalization terms proportional to such overlapping.

Algorithm 1 : $OC - SS\&GA$ FOR BICLUSTERING

INPUT Microarray M, penalization factors M_1, M_2 and M_3, number of biclusters $numBi$ to be found, maximum number of iterations $numIter$ to obtain a bicluster, size of the population and size S of the reference set.
OUTPUT The set $Results$ containing $numBi$ biclusters.
begin
  $num \leftarrow 0$, $Results \leftarrow \emptyset$
  while ($num < numBi$) **do**
    Initialize population P randomly
    //Building Reference Set
    $R_1 \leftarrow S/2$ best biclusters from P (according to the fitness function)
    $R_2 \leftarrow S/2$ most scattered biclusters, regarding R_1, from $P \setminus R_1$ (according to a distance).
    $RefSet \leftarrow (R_1 \cup R_2)$
    $P \leftarrow P \setminus RefSet$
    //Initialization
    stable $\leftarrow$ FALSE, $i \leftarrow 0$
    while ($i < numIter$) **do**
      while (NOT stable) **do**
        $A \leftarrow RefSet$
        $B \leftarrow CombinationMethod(RefSet)$
        $RefSet \leftarrow S$ best biclusters from $RefSet \cup B$
        if ($A = RefSet$) **then**
          $stable \leftarrow TRUE$
        end if
      end while
      //Rebuilding Reference Set
      $R_1 \leftarrow S/2$ best biclusters from $RefSet$
      $R_2 \leftarrow S/2$ most scattered biclusters from $P \setminus R_1$
      $RefSet \leftarrow (R_1 \cup R_2)$
      $P \leftarrow P \setminus RefSet$
      $i \leftarrow i + 1$
    end while
    //Storage in Results
    $Results \leftarrow$ the best from $RefSet$
    $num \leftarrow num + 1$
  end while
end

All steps of the Algorithm 1 going to be detailed in the

sequel.

3.1 Biclusters Codification and Generation

After preprocessing and normalization steps, a microarray can be seen as a real matrix M composed by N genes and L conditions. The element (i, j) of the matrix means the level of expression of gene i under the condition j. A bicluster is a submatrix of the matrix composed by $n \leq N$ rows or genes and $l \leq L$ columns or conditions.

Biclusters are encoded by binary strings of length $N + L$ [6]. Each of the first N bits of the binary string is related to the genes and the remaining L bits to the conditions. For example, the bicluster shown in Figure 1 is encoded by the following string: 0010110000|01100. Thus, this string represents the bicluster composed by genes number 3, 5 and 6 and conditions 2 and 3 from a microarray comprising 10 genes and 5 conditions.

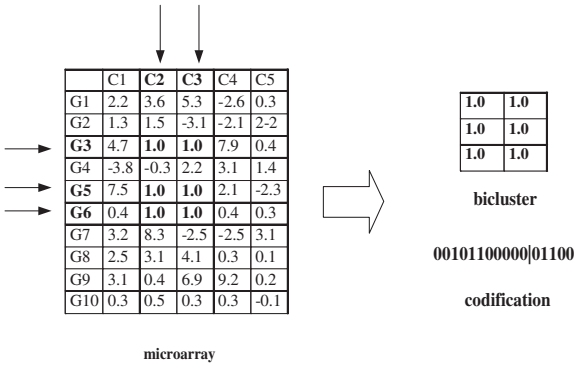

Figure 1. Microarray and bicluster along with its codification.

The initial population of biclusters is generated randomly as in Genetic Algorithms. Random strings composed by 0 and 1 are generated until the size of the population is reached.

3.2 Building Reference Set

The reference set is built taking into account both quality and scattering of biclusters. The quality of biclusters is measured evaluating the fitness function considered in the evolutionary process. A bicluster is better than another one if the fitness function value is lower than that of the second one. On the other hand, a distance has to be used in order to define what means scatter in this context. In this work, the distance used is the *Hamming* distance. The *Hamming* distance for two binary strings is defined by the number of positions for which their corresponding 0/1 values are different. For example, the *Hamming* distance between the string 001001001|001 and the string 001011001|101 is 2.

The reference set of size S is initially composed by the $S/2$ best biclusters from P (set R_1) and the $S/2$ biclusters from $P \setminus R_1$ (set R_2) with the highest distances to the set R_1 according to the *Hamming* distance.

3.3 Combination Method and Updating Reference Set

Combination method is the mechanism to create new biclusters in Scatter Search. All pairs of biclusters belonging to the reference set are combined generating $S * (S - 1)/2$ new biclusters. In the OC-SS&GA algorithm the typical uniform crossover operator used in Genetic Algorithms is the proposed combination method. This crossover operator is shown in Figure 2. A binary mask is randomly generated and a child is composed by values from the first parent when the mask set to 1, and from the second parent when the mask set to 0.

The reference set is updated with the S best biclusters, according to the fitness function, from the joining of the reference set and the new biclusters generated by the combination method. This process is repeated iteratively until the reference set does not change.

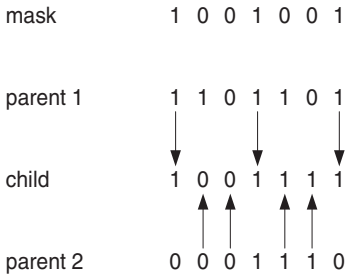

Figure 2. Uniform crossover operator of Genetic Algorithms.

3.4 Rebuilding reference set

After getting the stability of reference set in the updating process, this set is rebuilt to introduce diversity in the search process. This task is made by mutation operators in Genetic Algorithms. Thus, the reference set is composed by the $S/2$ best biclusters from the updated reference set (set R_1) according to the fitness function and the $S/2$ most distant from $P \setminus R_1$ according to the *Hamming* distance.

3.5 Overlapping Control

The overlapping between two biclusters $B1$ and $B2$ is the percentage of elements (i,j) from microarray M that are elements belonging to the biclusters $B1$ and $B2$. Obviously, the overlapping of one bicluster with itself is 100% and, therefore, it has not been considered. Thus, the overlapping between a bicluster and a set of biclusters is defined as the average of the overlapping between the bicluster and all biclusters belonging to the set.

In order to avoid the overlapping among biclusters, a set called $Results$ is defined as follows. The best bicluster belonging to each reference set obtained by the Scatter Search methodology is added to the set $Results$. The bicluster to be added has a low overlapping with the remaining biclusters that belong to the set $Results$ due to a penalization term included in the fitness function proportional to the overlapping between the bicluster and the set $Results$ (see section 3.6).

3.6 Biclusters Evaluation

The fitness function is used to evaluate the quality of biclusters. Cheng and Church proposed the MSR which measures the correlation of a bicluster. Given a bicluster comprising the subset of genes I and the subset of conditions J, the MSR is defined as follows,

$$MSR(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} R(i,j)^2$$

where

$$R(i,j) = e_{ij} - e_{Ij} - e_{iJ} + e_{IJ}$$

$$e_{Ij} = \frac{1}{|I|} \sum_{i \in I} e_{ij}$$

$$e_{iJ} = \frac{1}{|J|} \sum_{j \in J} e_{ij}$$

$$e_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} e_{ij}$$

In this work, non–overlapped biclusters with low residue and high volume are preferred. Therefore, the fitness function is defined by:

$$f(B) = MSR(B) + \frac{1}{rowVariance(B)} \quad +$$

$$+M_1 \left(\frac{1}{G}\right) + M_2 \left(\frac{1}{C}\right) + M_3 \left(Overlap(B, Results)\right)$$

where $MSR(B)$ is the MSR of the bicluster B, the second term is the inverse of the variance of rows of the bicluster, G and C are the number of genes and conditions of the bicluster B, respectively, and M_1, M_2 and M_3 are penalization factors to control the volume and the overlapping of the bicluster B with regards to the remaining biclusters that belong to the set $Results$.

4 Experimental Results

Yeast Saccharomyces cerevisiae cell cycle expression originated in [5] has been used to study the performance of the proposed algorithm. Original data were preprocessed in [4] replacing missing values with random numbers. The Yeast dataset contains 2884 genes and 17 experimental conditions.

The main parameters of the proposed algorithm are as follows: 100 for the number of biclusters to be obtained, 20 for the number of iterations to obtain each bicluster, 200 for the initial population size and 10 for the reference set size. The penalization factor for the number of genes has been set to the same order of magnitude to the range of values of the fitness function for Yeast dataset and to one order of magnitude larger than such range for the penalization factor related with the number of conditions [13].

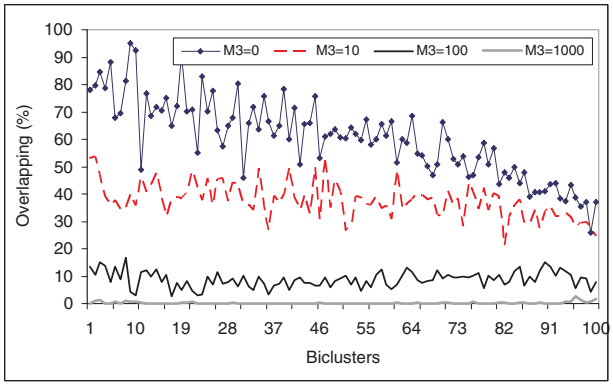

Figure 3. Overlapping among biclusters.

Figure 3 shows the percentage of overlapping among biclusters obtained by the OC-SS&GA algorithm for different penalization factors. The percentage of overlapping is referred to the overlapping between a bicluster and the set comprising the 99 remaining ones. The different values tested to control the overlapping have been set to 0 (without overlapping control), 10 (low overlapping control), 100 (medium overlapping control) and 1000 (large overlapping control). It can be noticed how this penalization parameter controls the overlapping among biclusters. When the overlapping is not controlled ($M_3 = 0$) the minimum overlapping among biclusters is 37%. However, the overlap-

ping of all biclusters is approximately zero when a large number ($M_3 = 1000$) is chosen for this parameter. For a low and medium penalization factor the overlapping ranges from 28% to 53% and from 3% to 17%, respectively. A moderate overlapping control is preferred, and therefore, a medium penalization is the only one considered in the sequel, that is, $M_3 = 100$.

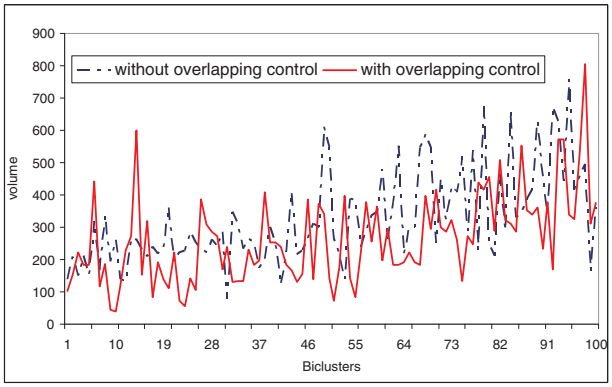

Figure 4. Volume of biclusters.

Figure 4 shows the volume of the hundred biclusters obtained for the proposed approach, without and with overlapping control. It can be notice that when the overlapping is controlled the volume of most of biclusters is lower than that of obtained biclusters when the overlapping is not taken into account.

Bicluster	MSR	Genes	Conditions	Row Variance
bi n1	76.63	8	13	224.97
bi n2	114.00	11	14	165.02
bi n23	179.82	17	13	231.09
bi n25	130.34	7	15	254.78
bi n27	222.92	22	14	408.31
bi n44	154.51	10	13	421.91

Table 1. Biclusters obtained by OC-SS&GA algorithm.

Table 1 provides information about six biclusters of the one hundred biclusters obtained by the OC-SS&GA algorithm. For each bicluster is shown an identifier of the bicluster, the value of its MSR, the number of genes, the number of conditions and its row variance. It can be observed that the OC-SS&GA algorithm find shifting and scaling patterns in gene expression data (see biclusters n23 and n44). These biclusters are shown in Figure 5. Noted that the genes forming each bicluster have different shapes indicating that the overlapping among these six biclusters is low.

Finally, a comparison between the results obtained by the proposed method without and with overlapping control and two representative techniques reported in the literature is presented. The use of MSR in the fitness function considered in the OC-SS&GA algorithm allows to establish a comparison with a previous evolutionary-based biclustering method called SEBI [6] and the well-known Cheng and Church algorithm [4]. CC searches biclusters iteratively taking into account the MSR value of each bicluster and SEBI uses MSR as part of its fitness function and it defines a mechanism to avoid the overlapping among biclusters.

Table 2 presents the average and the standard deviation (in brackets) of the MSR, the number of genes and the number conditions of the 100 biclusters found by the SS&GA, OC-SS&GA, SEBI and CC algorithms. The proposed algorithm, without and with overlapping control, improves the values of MSR regarding to that of the SEBI and CC algorithms. Obviously, the algorithm leads to higher MSR when the overlapping is taken into consideration. The number of genes of the biclusters obtained by the SS&GA and OC-SS&GA approaches is lower than the number of genes of the biclusters obtained by CC. This is due to the choice of the penalization factor for the number of genes. This factor can be increased if biclusters with more genes are considered more interesting. Finally, it can be stated that the proposed algorithm has a good performance yielding good results with respect to that of other techniques.

5 Conclusions

An algorithm for biclustering with overlapping control among biclusters has been presented in this work. The OC-SS&GA is an algorithm based on the evolution of populations, concretely, a hybrid metaheuristic based on Scatter Search and Genetic Algorithms. The algorithm avoid the overlapping among biclusters modifying the search taking into account the information of biclusters found in previous iterations. Experimental results from yeast cell cycle have been reported and the outcomes of the proposed approach have been compared with that of two representative biclustering techniques.

Future works will be focussed on the use the proposed biclustering algorithm with other fitness functions to measure the quality of biclusters.

Acknowledgments

The financial support given by the Spanish Ministry of Science and Technology, project TIN-68084-C02 and by the Junta de Andalucía, project P07-TIC-02611 is acknowledged.

References

[1] S. Bergmann, J. Ihmels and N. Barkai. Iterative signature algorithm for the analysis of large-scale gene expression data.

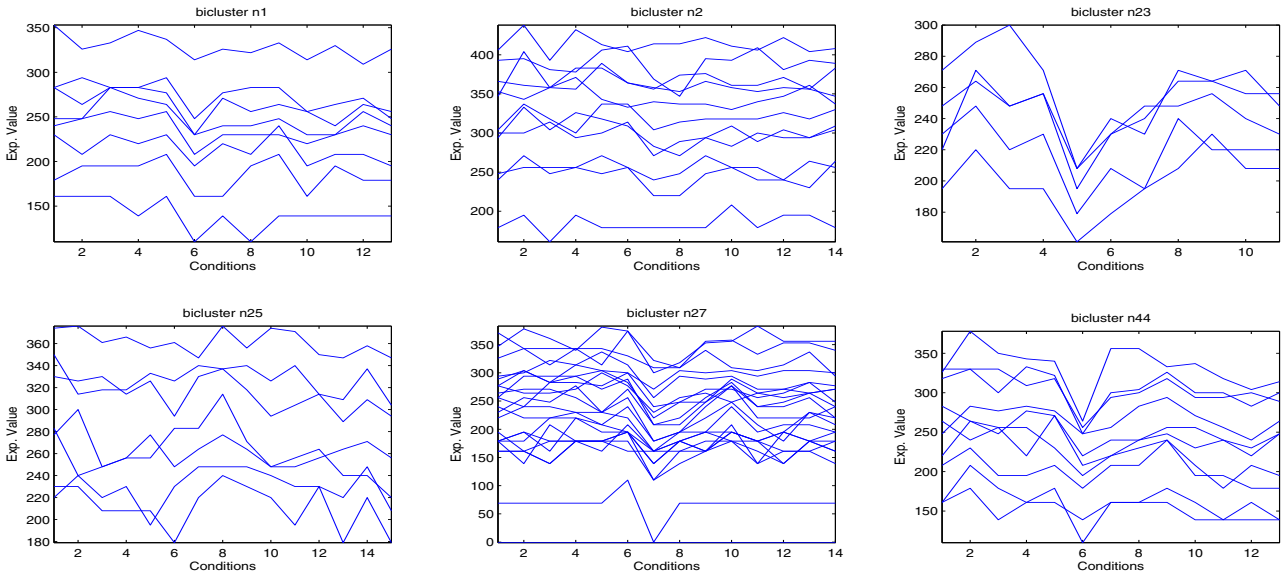

Figure 5. Biclusters from Yeast dataset

.

Algorithm	Average Residue	Average gene number	Average condition number
SS&GA	127.74 (33.33)	26.52 (12.53)	12.47 (1.17)
SS&GA-OC	178.27 (44.28)	21.23 (11.31)	12.48 (1.19)
SEBI	205.18 (4.49)	13.61 (10.38)	15.25 (1.37)
CC	204.29 (42.78)	166.71 (226.37)	12.09 (4.39)

Table 2. Comparison of the results.

Physical Review E, 67(3):31902-1–31902-18, 2003.

[2] K. Bryan, P. Cunningham, N. Bolshakova, T. Coll and I. Dublin. Biclustering of expression data using simulated annealing. *18th IEEE International Symposium on Computer-Based Medical Systems, pages 383–388, 2005.*

[3] S. Busygin, O. Prokopyev and P. M. Pardalos. Biclustering in data mining. *Computers and Operations Research*, 35(9):2964–2987, 2008.

[4] Y. Cheng and G. M. Church. Biclustering of Expression Data. *8th International Conference on Intelligent Systems for Molecular Biology*, pages 93–103, 2000.

[5] R. J. Cho et al. A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle. *Molecular Cell*, 2(1):65–73, 1998.

[6] F. Divina and J. S. Aguilar-Ruiz. Biclustering of Expression Data with Evolutionary Computation. *IEEE Transactions on Knowledge and Data Engineering, 18(5):590–602, 2006.*

[7] E. Levine, G. Getz and E. Domany. Couple two-way clustering analysis of gene microarray data. In *Proceedings of the National Academy of Sciences (PNAS) of the USA, 97 (22): 12079-12084, 2000.*

[8] R. Harpaz and R. Haralick. Exploiting the geometry of gene expression patterns for unsupervised learning. *18th International Conference on Pattern Recognition (ICPR 2006), pages 670–674, 2006.*

[9] J. S. Aguilar-Ruiz. Shifting and scaling patterns from gene expression data. *Bioinformatics*, 21(20):3840–3845, 2005.

[10] P. Larranaga et al. Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7(1):86–112, 2006.

[11] S. Mitra and H. Banka. Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition*, 39(12):2464–2477, 2006.

[12] J. A. Nepomuceno, A. Troncoso, J. S. Aguilar-Ruiz and J. Garcıa-Gutierrez. Biclusters Evaluation Based on Shifting and Scaling Patterns. *Lecture Notes in Computer Sscience*, 4881:840-849, 2007.

[13] J. Nepomuceno, A. Troncoso, J. S. Aguilar-Ruiz. A Hybrid Metaheuristic for Biclustering based on Scatter Search and Genetic Algorithms. *Lecture Notes in Bioinformatics*, in press, 2009.

[14] R. Marti and M. Laguna. *Scatter Search. Methodology and Implementation in C.* Kluwer Academic Publishers, Boston, 2003.

[15] A. Tanay, R. Sharan and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(1):136–144, 2002.

[16] J. Yang, H. Wang, W. Wang and P. Yu. Enhanced biclustering on expression data. 3th IEEE Simposium on Bioinformatics and Bioengeneering, pages 321–327, 2003.