Applied Soft Computing 35 (2015) 637–651

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Scatter search-based identification of local patterns with positive and negative correlations in gene expression data

Juan A. Nepomuceno[a,*], Alicia Troncoso[b], Jesús S. Aguilar-Ruiz[b]

[a] Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Avd. Reina Mercedes s/n, 41012 Seville, Spain
[b] Department of Computer Science, School of Engineering, Pablo de Olavide University, Ctra. Utrera km. 1, 41013 Seville, Spain

ARTICLE INFO

Article history:
Received 15 April 2014
Received in revised form 22 May 2015
Accepted 18 June 2015
Available online 9 July 2015

Keywords:
Biclustering
Scatter search
Gene expression data

ABSTRACT

This paper presents a scatter search approach based on linear correlations among genes to find biclusters, which include both shifting and scaling patterns and negatively correlated patterns contrarily to most of correlation-based algorithms published in the literature. The methodology established here for comparison is based on a priori biological information stored in the well-known repository *Gene Ontology* (GO). In particular, the three existing categories in GO, *Biological Process*, *Cellular Components* and *Molecular Function*, have been used. The performance of the proposed algorithm has been compared to other benchmark biclustering algorithms, specifically a group of classical biclustering algorithms and two algorithms that use correlation-based merit functions. The proposed algorithm outperforms the benchmark algorithms and finds patterns based on negative correlations. Although these patterns contain important relationship among genes, they are not found by most of biclustering algorithms. The experimental study also shows the importance of the size in a bicluster in addition to the value of its correlation. In particular, the size of a bicluster has an influence over its enrichment in a GO term.

1. Introduction

Gene expression data provide the information that is collected from a group of microarray chips, each of which is built for a specific sample. The samples are generated according to a concrete experimental condition such as temperature, steps in the cell cycle or characterization of a patient. One single chip measures the expression level of thousands of genes in the sample under study [1]. After several preprocessing procedures, which comprise a process known as *low level microarray analysis*, the joining of the data from all of the samples constitutes the gene expression data to be analyzed. A gene expression matrix can be considered a two-dimensional numerical matrix, in which the rows are genes and the columns are the experimental conditions that are under study in each sample. A value in the matrix represents the gene expression value under a specific experimental condition. Data Mining techniques applied to infer knowledge from high-dimensional gene expression data sets comprise a process known as *high-level microarray analysis*. Most of these techniques are motivated by a simple idea, which is widely used in functional genomics: co-expression means co-regulation [2]. This assumption is called the guilt-by-association heuristic and is essential to study biological systems through “omic” data analysis.

Biclustering is a *unsupervised machine learning* technique that simultaneously clusters instances and features of the data set matrix. Unlike most of the *clustering* techniques, biclustering allows the overlapping among the results instead of making clusters which divide the data space. Therefore, the motivation is more to discover hidden information than to describe the data. Biclustering is a NP-hard problem considered first by Morgan and Sonquist [3], and later by Hartigan [4] and by Mirkin [5]. It can be found in the literature with other names, such as co-clustering [6] or subspace clustering [7]. In the context of gene expression data analysis, biclustering identifies patterns from gene expression data [8] and it was introduced by Cheng and Church [9].

Most of biclustering algorithms use the *mean squared residue* (MSR) measure [9] to obtain biclusters. Although scaling patterns are essential from a biological point of view, the MSR does not capture them when the gene variance values are high in the bicluster [10]. Recently, other measures based on correlations have been proposed to find biclusters. These measures are able to capture shifting and scaling patterns but do not obtain *activation–inhibition* expression patterns, which were presented in [11] and are a common feature in many molecular pathways [12,13].

[*] Corresponding author. Tel.: +34 954 559 769.
E-mail addresses: janepo@us.es (J.A. Nepomuceno), ali@upo.es (A. Troncoso), aguilar@upo.es (J.S. Aguilar-Ruiz).

http://dx.doi.org/10.1016/j.asoc.2015.06.019
1568-4946/© 2015 Elsevier B.V. All rights reserved.

BIclustering algorithm based on a **S**catter **S**earch scheme, called BISS, is presented in this paper. The BISS approach attempts to overcome all the drawbacks of the existing biclustering algorithms to find biclusters including activation–inhibition patterns, in addition to both shifting and scaling patterns. The scatter search metaheuristic is a population-based evolutionary optimization method that emphasizes systematic processes against random procedures, in contrast to genetic algorithms [14]. The initial population of solutions is built using a diversification method, which non-randomly generates solutions with special characteristics. The fitness function is based on linear correlations among genes but also it considers negative correlations. The optimization process constitutes the evolution of a small set of solutions and includes a local search procedure, which intensifies the search without losing information from the scatter solutions of the problem. To evaluate the proposed algorithm two experiments have been carried out. As initial step, the proposed algorithm has been compared to classical algorithms of biclustering to analyze its potential to obtain biclusters. Secondly, other existing approaches based on correlations have been used with the purpose of comparing the kind of patterns discovered by the proposed algorithm.

The remainder of this paper is organized as follows. A review of the bibliography of biclustering is provided in Section 2. In Section 3, the proposed algorithm is presented to account for two aspects: first, how the search engine works, and second, the description of the merit function. Section 4 summarizes the experimental results, including a comparison of the performance of our approach to other biclustering methods. Finally, Section 5 is devoted to conclusions and future research.

2. Related research

Many biclustering algorithms have been proposed in recent years. These algorithms can be classified according to the type of patterns that are found, the size of the biclusters or the heuristic strategies that are used [8,15,16]. There is not a common criterion to compare different algorithms [17,18]. Some comparison methodologies are based on statistical metrics [19] or on the study of the behaviour over known synthetic data sets [20]. However, a sufficiently accepted methodology is based on enrichment analyses and uses a priori biological information that is stored in known repositories, such as GO [21].

The first proposals for solving the problem of searching for local patterns in gene expression data stem from the clustering field. An iterative hierarchical clustering is separately applied to genes and conditions, and the resultant biclusters are the combination of obtained clusters for each dimension in [22]. In 2000, Cheng and Church were the first to consider the biclustering in the context of gene expression data [9]. The Cheng and Church algorithm (CHCH) is a deterministic greedy iterative search method that obtains biclusters with a low MSR. If I and J are the sets of rows (genes) and columns (conditions) in a specific bicluster, respectively, then the MSR is defined as:

$$MSR = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2 \qquad (1)$$

where a_{ij} is the expression value in row i and column j, a_{iJ} is the mean of the expression values in row i, a_{Ij} is the mean of the expression values in column j and a_{IJ} is the mean of the complete bicluster. The algorithm begins with the whole data matrix, and it iteratively adds or removes rows and columns until it finds a bicluster with a residue that is less than a given threshold. The process is repeated until the required number of biclusters is obtained. The number of biclusters to be obtained is an input parameter. The FLexible Overlapped biClustering (FLOC) algorithm [23] improves CHCH by obtaining simultaneously a set of biclusters and incorporating a strategy for addressing missing values in the process.

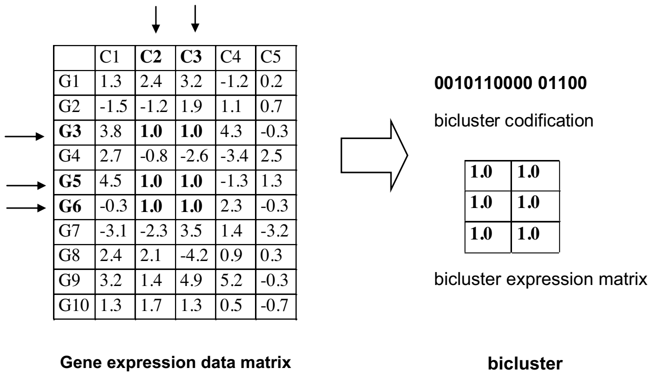
During recent years, several algorithms based on different techniques have been proposed. The Iterative Signature Algorithm (ISA) [24] is a nondeterministic greedy algorithm that finds up- and down-regulated biclusters. The algorithm starts with a random set of rows, and it iteratively updates columns and rows until convergence. Specifically, each column and row in a bicluster must have an average value that is less than several parameters, which measure symmetric requirements to obtain up- or down-regulated biclusters. This process is repeated using different seeds. The Order Preserving Submatrix (OPSM) algorithm [25] is a deterministic greedy algorithm that searches for biclusters according to a model that is based on linear ordering among rows. Most of the interesting patterns, such as constant values, shifting or scaling, are captured by this model. The biclusters are built through a scoring system, and the best bicluster is selected for each iteration of the algorithm. The Statistical-Algorithmic Method for Bicluster Analysis (SAMBA) [26] is a greedy algorithm that is based on an exhaustive bicluster enumeration using a bipartite graph model. It adds or removes nodes to find maximum weight subgraphs. The Plaid Model [27] algorithm is a statistical modelling approach that represents the input matrix as a superposition of layers, where each layer corresponds to a bicluster. It iteratively adjusts the parameters of each layer to handle its MSR. Spectral biclustering [28] identifies biclusters using techniques from linear algebra, especially eigenvector calculus. The idea comprises capturing up- or down-regulated biclusters with a variance that is lower than a given threshold. The characterization of the biclusters as hyperplanes in a high-dimensional space is the goal of several algorithms, which use image processing techniques [29] or Hough transform-based hyperplane detection algorithms [30].

The combination of a search engine and a measure characterizing the patterns that are sought is the methodology followed by a broad family of biclustering algorithms. These algorithms are optimization metaheuristics which are adapted to gene expression data, such as evolutionary approaches [31–33], multiobjective evolutionary approaches [34,35], greedy randomized adaptive search [36], simulated annealing [37], particle swarm Optimization [38] or estimation of distribution algorithms [39]. Most of these algorithms use the MSR as part of their merit function to characterize the types of patterns that are relevant to be found.

Recently, several biclustering algorithms using measures based on correlations have been proposed to obtain co-expressed genes [40–46]. In particular, BCCA [44] is a nondeterministic greedy algorithm which builds an initial bicluster composed of a pair of genes by removing experimental conditions while the Pearson correlation coefficient is lesser than a given threshold, and later, all the genes maintaining the correlation are added to this bicluster. In the same way, BICLIC [45] obtains a seed bicluster by applying clustering to each dimension separately, and second, the biclusters are expanded during the search process. The algorithm presented in [46] is based on a scatter search scheme. Although this algorithm could be considered to belong to the family of biclustering algorithms based on optimization metaheuristics, it uses the correlation instead of the MSR as a merit function. Although these methods use measures based on correlations, neither of them except for BICLIC captures negative correlations among genes.

Several biclustering algorithms which are specifically designed for binary datasets [21,47] or time series gene expression data [48,49] can be found in the literature. In the case of algorithms for binary datasets, a discretization step is necessary before the algorithm is applied, and therefore, a preprocessed matrix is used instead of the original gene expression data matrix. Thus, the preprocessing step is essential in this type of approach. In the case of time series data, the biclusters must have contiguous columns, and this constraint becomes a polynomial time problem [48]. The CCC-biclustering algorithm discretised the matrix and used string processing techniques based on suffix trees [48]. e-CCC-biclustering [49], proposed by the same authors, found approximate expression patterns, in other words, patterns in which a certain number of errors in the expression matrix is allowed. These possible errors can have an influence on the discretization process.

Currently, the study of new measures [50] and the integration of different sources of large biological information sets for the discovery of co-regulated genes, not only from gene expression data, is considered to be future topics for the biclustering community. In this context, the combination of gene expression data and sequence data to discover biclusters that represent regulatory modules has been proposed [51].



	C1	C2	C3	C4	C5
G1	1.3	2.4	3.2	-1.2	0.2
G2	-1.5	-1.2	1.9	1.1	0.7
G3	3.8	**1.0**	**1.0**	4.3	-0.3
G4	2.7	-0.8	-2.6	-3.4	2.5
G5	4.5	**1.0**	**1.0**	-1.3	1.3
G6	-0.3	**1.0**	**1.0**	2.3	-0.3
G7	-3.1	-2.3	3.5	1.4	-3.2
G8	2.4	2.1	-4.2	0.9	0.3
G9	3.2	1.4	4.9	5.2	-0.3
G10	1.3	1.7	1.3	0.5	-0.7

0010110000 01100

bicluster codification

1.0	**1.0**
1.0	**1.0**
1.0	**1.0**

bicluster expression matrix

Fig. 1. A toy example of a gene expression matrix and a bicluster with its binary codification.

3. Method

Scatter search is a population-based evolutionary metaheuristic in which a population comprises a small set of solutions and evolves until an optimal solution is reached [14]. One of the most important aspects in scatter search schemes with regard to other evolutionary metaheuristics, such as genetic algorithms, is that systematic processes are emphasized against random procedures. Moreover, the population evolves while accounting for only two strategies during the searching process: intensification and diversity. The goal of the intensification is to improve the quality of the solutions and thus capture the best solution. The goal of the diversity strategy is to maintain a set of scatter solutions to avoid local solutions that could stop the process in its early stages. The word scatter is motivated by the idea of considering solutions that are scattered whenever possible in the reference set to explore the complete data space during the search process. The small set of solutions, usually called the *reference set* in the scatter search literature, is built with a group of solutions which are selected by considering the previous two strategies in each iteration.

With regard to the merit function used in the BISS algorithm, the main motivation is that correlated genes imply co-expressed genes and, therefore, imply the same regulatory regime for a group of genes. Most of the relevant patterns are captured using correlation-based measures. These patterns are shifting and scaling patterns that express both the activation of genes at the same time with the same or proportional intensity and activation–inhibition patterns. It is essential to consider negative correlations to capture this second type of pattern. Furthermore, some extra terms to control the size of the biclusters are added to the fitness function.

3.1. Searching procedure

The input data set is the gene expression data matrix D, where an element (i, j) is the expression level of gene i under condition j. A bicluster is a submatrix of the matrix that is composed of a subset of rows and a subset of columns. A bicluster is encoded as a binary string in which the initial terms inform about what genes are included and the latter terms inform about what conditions. Fig. 1 shows an example where the string 0010110000||01100 represents a bicluster of a microarray with ten genes and five conditions, $\{g_i\}_{1\leq i\leq 10}$, and five conditions, $\{c_j\}_{1\leq j\leq 5}$. This string encodes the bicluster composed of the genes g_3, g_5 and g_6 and the conditions c_2 and c_3.

The pseudocode of the BISS algorithm is presented in Algorithm 1. This algorithm is based on an adaptation of a scatter search scheme for finding a bicluster, and the process is repeated according to the number of biclusters to be reported (the external loop is defined by lines 3 and 23). Therefore, the algorithm is a sequential covering algorithm, in other words, the number of biclusters to find is established as the input of the method, and each bicluster is independently obtained. For each bicluster, first, an initial population is generated (line 4), and second, the *reference set* is built (line 6). This set initially contains the most representative solutions from the initial population with respect to both quality and diversity criteria. The *reference set* evolves by generating new solutions (lines 10–15), and it is rebuilt when the process is stable (line 16). This process is repeated a number of times (the inner loop is defined by lines 9 and 19). The mechanism to build or re-build the *reference set* is to add both the best solutions from the initial population (line 6) or from previous reference sets (line 16) and the solutions from the initial population as scatter as possible regarding the solutions that belong to the set. Consequently, after building or rebuilding the *reference set*, the initial population must be updated by removing the solutions that were previously added into the *reference set* (lines 7 and 17). Moreover, a local search procedure is applied to make the search more efficient (lines 5 and 13). Finally, after a given number of iterations *numIter*, the best solution in the *reference set* according to the fitness function is chosen, and the bicluster encoded by this solution is stored in the set *results* (line 21).

Next, the different steps in Algorithm 1 are detailed. Specifically, we describe how the initial population is generated, the fitness function considered in order to find activation–inhibition patterns, in addition to both shifting and scaling patterns, how the minimum correlation is computed, what is the meaning of the improvement method, how the *reference set* is built and re-built and how new solutions are created.

3.2. Initial population

Initial populations of size 200 are usually recommended in scatter search algorithms [14]. These solutions are not randomly generated, but a systematic process called *diversification generation method* is used. This method generates new solutions from a seed solution by following a diversity rule. Specifically, if the seed is a binary string, x_i with $i = 1, \ldots, n$, where n is the number of bits, then new solutions are obtained with the following equation:

$$x\prime_{1+kh} = 1 - x_{1+kh} \quad \textbf{for } k = 0, 1, 2, 3, \ldots, \lfloor n/h \rfloor \tag{2}$$

where $\lfloor n/h \rfloor$ is the largest integer that is less than or equal to n/h, and h is an integer that is less than $n/5$. All the remaining bits of x' are equal to those of x. After generating all the possible solutions with that seed, if more solutions are needed, the diversification generation method is applied again, using the last solution as a new seed. Each solution representing a bicluster is improved by a local search procedure before it is stored in the population.

3.3. Fitness function

The fitness function used as a merit function in the BISS algorithm is based on linear correlations because correlated genes imply co-expressed genes. Most of the relevant patterns, such as shifting and scaling patterns, are captured using the correlations among the genes. Two genes follow a shifting pattern if they increase and decrease at the same time and with similar intensity. Moreover, they follow a scaling pattern if this behaviour occurs with different levels of intensity. The difference between these two types of patterns results from considering a linear combination with additive or multiplicative terms [10]. The activation–inhibition patterns are also very relevant from a biological point of view. These patterns can be observed when the increase in a gene value is related to a similar decrease in another gene's value [11], that is, they can be described by negative correlations. Linear correlation measures the grade of the linear dependency between two vectors so that it captures shifting, scaling and activation–inhibition patterns among genes. Note that although the algorithm presented in [46] uses a scatter search scheme and a fitness function based on linear correlations, negative correlations among genes are not captured because of the design of its fitness function and its improvement method.

The Pearson correlation coefficient between two vectors X and Y is defined by:

$$\rho(X, Y) = \frac{cov(X, Y)}{\sigma_X \sigma_Y} = \frac{\sum_i^n (x_i - \bar{x})(y_i - \bar{y})}{n\sigma_X \sigma_Y} \tag{3}$$

where $cov(X, Y)$ is the covariance of the variables X and Y, $\bar{x}$ and $\bar{y}$ are the average of the values of the variables X and Y and σ_X and σ_Y are the standard deviations of X and Y, respectively.

Note that $-1 \le \rho_{ij} \le 1$, which implies that if the value is close to 0, gene i and gene j will show different behaviour. However, if the value is close to ±1, then they will have the same or opposite tendencies. Values from 0 to 1 indicate that two genes are *positively correlated* and that they show the same tendency; if one of them increases its value, then the other also increases its value. Values from 0 to −1 indicate that the genes are *negatively correlated*; they have a complementary tendency: one of them increases while the other one decreases with the same intensity, and vice versa.

Given a bicluster B composed of N genes, the mean of absolute values of correlation between pairs of genes, noted $\rho_{|.|}(B)$, is defined as:

$$\rho_{|\cdot|}(B) = \frac{1}{\binom{N}{2}} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} |\rho_{ij}| \tag{4}$$

where ρ_{ij} is the correlation coefficient between the gene i and the gene j. Note that $\rho_{ij} = \rho_{ji}$, hence only $\binom{N}{2}$ elements have been considered. It is important to highlight that the absolute value is considered to avoid losing relevant information. A couple of genes with positive correlations could eliminate the effect of other two genes with negative correlations. For example, two genes with correlations equal to 0.9 would cancel to other two genes with correlation equal to −0.9 if the absolute value were not considered. Moreover, note that the Equation 4 captures the negative correlations among the genes, and therefore, activation–inhibition patterns are considered.

The fitness function used to evaluate the quality of the biclusters is defined by

$$f(B) = (1 - \rho_{|\cdot|}(B)) + \sigma_\rho(B) + M\left(\frac{1}{vol(B)}\right) \tag{5}$$

Table 1
Correlations among genes 1, 2, 3 and 4.

	g1	g2	g3	g4
g1	1	0.14	−0.01	−0.05
g2		1	0.98	−0.89
g3			1	−0.93
g4				1

where $vol(B)$ is the volume of the bicluster (i.e., the number of genes multiplied by the number of conditions), M is a penalty factor to control the volume, and σ_ρ is the standard deviation of the values ρ_{ij}. The analysis of the influence of the parameter M is described in the following section. The standard deviation is included to avoid that the value of the average correlation could be high although the bicluster could contain a subgroup of several non-correlated genes.

3.4. Minimum correlation method

A procedure for determining a minimum value for the correlation is conducted before the iterative process begins (line 2). This value depends on the data set, and it is an input parameter for the improvement method because the population is improved by removing the genes that have a correlation lower than the minimum correlation which is computed in this step. The procedure to select the minimum correlation, ρ, is now discussed. This procedure initially generates 100 random biclusters and computes the number of biclusters that are improved according to the fitness function by varying the parameter ρ from 0.1 to 0.9. A bicluster is improved if the fitness function decreases when removing the genes in the biclusters with a correlation lower than ρ. The ρ selected is the value that maximizes the number of biclusters which were improved.

3.5. Improvement method

This method is a local search procedure that improves biclusters in relation to the value of its fitness function [52]. The local search used in a scatter search algorithm depends on the nature of the problem. In this case, the objective of this method is to improve the average correlation of the bicluster by removing those genes which are not sufficiently correlated; in other words, the correlation with at least one gene in the bicluster is lower than the minimum correlation ρ that was previously established. The pseudocode of this procedure is summarized in Algorithm 2. From this pseudocode, it can be observed that the set R is composed of the poorly correlated genes which are finally removed (lines 5–11).

Fig. 2 presents an example of a bicluster that is composed of four genes and has an average correlation of 0.70 according to Eq. (4). It can be observed that genes 2 and 3 present scaling patterns and that gene 4 shows a negative correlation pattern regarding the genes 2 and 3. However, gene 1 does not follow a pattern. Table 1 shows the correlations among these four genes. If the minimum correlation is 0.5, then gene 1 must be removed because its correlation with genes 2, 3 or 4 is lower than 0.5. Once gene 1 is deleted, the average of the correlation of the bicluster composed of genes 2, 3 and 4 is 0.98.

3.6. To build and re-build the reference set

The *reference set* is a small set of solutions; usually, 10 solutions comprise a typical value in scatter search schemes [14]. This set is built using the best and most scattered solutions. The best solutions are solutions with a low value of the fitness function. The best solutions are selected from the initial population at the first time the *reference set* is built (line 6) or from previous reference set at the other times at which the set is re-built (line 16). It is important

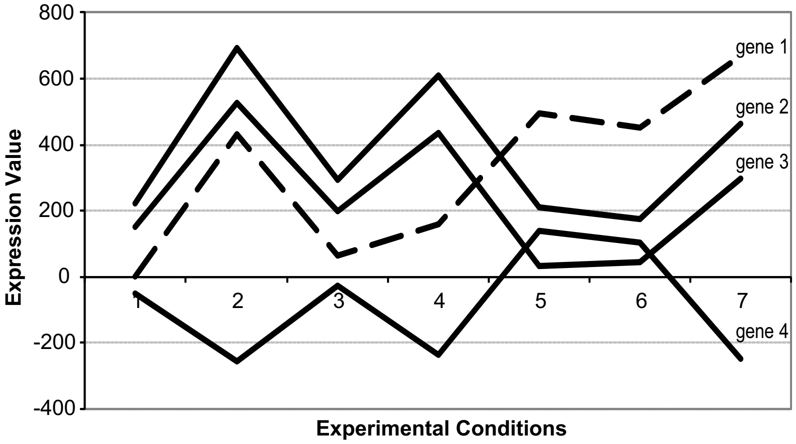


Fig. 2. A bicluster before (with the dashed line) and after (without the dashed line) applying the improvement method.

to update the initial population by removing all the solutions that are added to the *reference set* when building and re-building the set during the process. Thereby the exploration of the data space is as exhaustive as possible. Once the best solutions are selected, the most scattered solutions with respect to them are chosen from the initial population. The concept of scatter needs the definition of a distance. In this work, the *Hamming* distance has been used to measure the distance among two solutions because of the solutions are encoded by binary strings [14]. Although the BISS algorithm uses a small number of solutions, the searching process is efficient when these solutions are very representative of the data space. In other words, the best solutions from a quality point of view intensify the convergence to the optimum, and the most scattered solutions explore the complete data space to avoid local optima.

3.7. Subset generation method, solution combination method and reference set update method

The *reference set* evolves by creating new solutions from subset generation and solution combination methods. The method of subset generation creates all the possible subsets of pairs of solutions. If S is the size of the *reference set*, then $S \times (S-1)/2$ subsets can be generated. Then, all the pairs of solutions in the *reference set* are combined using a uniform crossover operator. This crossover operator randomly generates a mask. The child solution contains values from the first parent when there is a 1 in the mask and from the second parent when there is a 0. This crossover operator has been used due to the biclusters are encoded as binary strings. Finally, the *reference set* must be updated by choosing the best S solutions according to the fitness function. The election of the S solutions is made from the joining of the $S \times (S-1)/2$ solutions generated by the solution combination method and improved by the improvement method together with the S solutions, which were in the reference set.

4. Experiments

In this section, the results obtained from the application of the BISS approach are presented. First, Section 4.1 provides a detailed description of the three datasets used in this work. An analysis of the parameter configuration used for the BISS algorithm can be found in the Section 4.2. Finally, the results are gathered and discussed in Section 4.3. In particular, the performance of BISS is compared to the most representative approaches reported in the literature and to two specific algorithms based on correlations in Sections 4.3.1 and 4.3.2, respectively and a study of biological significance of several biclusters is made in Section 4.3.3.

4.1. Dataset description

Three datasets, two from *Saccharomyces cerevisiae* yeast and one from a *Homo sapiens* dataset related with the alzheimer disease, have been used in the experimental study presented in this paper. Many authors have evaluated their own approaches over the *Saccharomyces cerevisiae* organism, and as a consequence, the literature offers multiple results for this organism. Moreover, there is a good knowledge of this organism because of its use in the food industry (e.g., bread, beer, wine) and its importance in molecular biology studies.

One of the yeast datasets, called *GaschYeast*, and the Homo sapiens dataset, called *Alzheimer*, have been downloaded from the supplementary information in papers [21,53], respectively. The GaschYeast dataset is composed of 2993 genes and 173 samples and the Alzheimer dataset comprises 1663 genes and 33 samples. The other yeast dataset has been obtained from the Gene Expression Omnibus (GEO)[1] repository. Specifically, the dataset record *GDS1116* reported in a previous study [54] has been used in this work. This dataset is not a time series gene expression data set, and the dimensionality is 7084 genes and 131 samples. The missing values have been preprocessed using the GEPAS[2] tool. This tool basically removes the genes with a large number of missing values and replaces missing values using the mean or median of the row or column values of the gene expression data matrix. In this case, the genes with a percentage of missing values in the expression profile greater than 80% were removed, and the missing values were replaced with the average of the expression profile. After this preprocessing process, approximately 12% of the genes were removed, and the final GDS1116 microarray dataset was composed of 6229 genes and 131 samples.

4.2. Parameter configuration

The BISS algorithm presents two parameters, a penalty factor in its fitness function, which controls the size of biclusters to be obtained, and a minimum correlation, which is used to improve the biclusters in the evolutionary process using the improvement

[1] http://www.ncbi.nlm.nih.gov/geo/.

[2] http://www.gepas.org/.

Table 2
Results obtained by BISS for different values of penalization.

	BISS			
	$M=1$	$M=10$	$M=20$	$M=40$
Genes	11.0	281.4	386.5	415.1
Conditions	12.7	34.4	34.1	34.0
Size	138.0	10,057.7	13,244.8	13,929.8
$\rho_{\vert\cdot\vert}(B)$	0.89	0.30	0.23	0.21

method. In this section, the value for the penalization parameter is analyzed to determine which value of this parameter is the most adequate, and moreover, the minimum correlation is computed as it was previously described in Section 3.4. Only the results for GDS1116 dataset are shown but similar results have been obtained for the other datasets.

Table 2 reports the average of the number of genes, the average of the number of conditions, the average of the size or volume of the biclusters and the average of the values of the correlation for 100 biclusters obtained using the BISS algorithm for different values of the penalization parameter M, specifically, the values 1, 10, 20 and 40. The correlation of a bicluster is defined by Eq. (4). It can be observed that a low number for the penalization implies a small volume of the biclusters and, in particular, a low number of genes. The number of genes increases when the value of the penalization increases from 10 to 40, but the number of conditions remains constant. With regard to the correlation, a low penalization provides biclusters with correlations that are close to 1, and therefore, highly correlated genes (negatively or positively). Although the highest correlation is obtained when the penalization is the smallest value ($M=1$), this value provides too small biclusters. We can conclude that the value of M must be chosen depending on whether the user is interested in searching for biclusters with highly correlated genes or in searching for biclusters with a very large number of genes.

In addition to the results related to the number of genes or conditions and the correlations of the biclusters obtained by the BISS algorithm, it is also interesting to analyze the biological relevance of biclusters for different values of penalization. For this purpose, the GO repository has been used.

GO is an ontology or vocabulary in which genes and gene products are annotated in terms of their molecular functions, the biological processes in which they are involved and the cellular locations in which they are active. For this reason, GO has a tree structure that has three different sub-ontologies or branches: *Cellular Component* (CC), *Molecular Function* (MF) and *Biological Process* (BP). The CC category summarizes the parts of a cell or its extracellular environment, the MF category summarizes the elemental activities of a gene product at the molecular level, such as binding or catalysis and the BP category summarizes sets of molecular events or operations which have a defined beginning and end, in the functioning of integrated living units such as cells, tissues, organs and organisms.

The enrichment of biclusters according to GO is used as benchmark analysis to compare biclustering algorithms [21]. The comparison is based on the percentage of biclusters with a group of genes which are biologically relevant. A bicluster is said to be *over-represented* or *enriched* in a GO term if the group of genes that form the bicluster is statistically significant with respect to that term for a given significance level, that is, the p-value is lower than a certain threshold. The AGO tool presented in [55], which is based on the *GeneMerge* software [56], has been used in this paper to determine the enriched biclusters. *GeneMerge* uses the hypergeometric statistical test and the multiple-testing adjustments of Bonferroni as correction method.

Fig. 3 presents the percentage of enriched biclusters obtained using the BISS algorithm for different values of penalization M and different significance levels. The percentage of enriched biclusters has been obtained from the three GO components, CC, MF and BP, and the average is presented in this Figure for each level of significance.

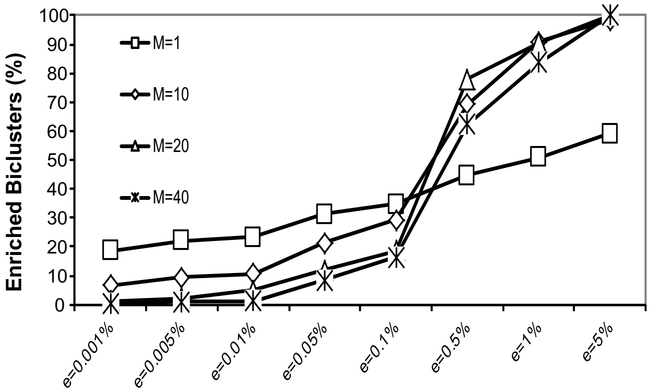


Fig. 3. Percentage of enriched of biclusters obtained by BISS for different values of penalization.

A significance level of 90% ($e=0.1\,\%$) or 95% ($e=0.05\,\%$) is usually used, that is, a bilcuster is enriched if its p-value is lower than 0.1 or 0.05, respectively. For a significance level of 95%, the percentage of enriched biclusters for $M=1$ is greater than that for $M=10$ (31.34% versus 21.34%). However, for a significance level of 90%, a similar percentage of enriched biclusters is provided for both penalization factors (34.67% and 29.34%, respectively).

From the observations presented in Table 2 and Fig. 3, it can be concluded that a high negative or positive correlation among genes implies biologically relevant biclusters according to GO. Thus, any value from $M=1$ to $M=10$ can be a good election depending on the required size of the biclusters. In this work, the value of $M=5$ has been chosen, with the goal of finding a balance between biological relevance and the size of the biclusters.

Fig. 4 shows the percentage of biclusters that improved according to the fitness function for different values of correlation. Here, 99% of the biclusters decrease the value of the fitness function when genes with an absolute value of correlation lower than 0.3 or 0.4 are removed. Thus, either of these two values are adequate to be used in the improvement method as the minimum correlation.

4.3. Results

The results obtained from the application of the BISS algorithm to the three datasets and the comparison between the performance of BISS and other approaches published in the literature are reported in this section. First, a comparison with the classical biclustering algorithms implemented in the BiCAT tool [57], such as CHCH [9], ISA [24] and OPSM [25] is presented and second, the performance of the BISS algorithm is compared to BCCA [44] and BICLIC [45] algorithms.

The experimental setting of the BISS algorithm is 5 for the penalization parameter and 100 for the number of biclusters to be obtained. Note that BISS algorithm is based on a scatter search heuristic that is repeatedly applied to obtain biclusters. Each bicluster is obtained independently, and hence, a high number of this input parameter controls the randomized nature of the scatter search.

4.3.1. Comparison with classical algorithms

A comparison with classical biclustering algorithms implemented in the BiCAT tool [57], such as CHCH [9], ISA [24] and OPSM [25], is presented in this subsection. Although the xMotifs [58] algorithm is included in the BICAT tool, it has not been

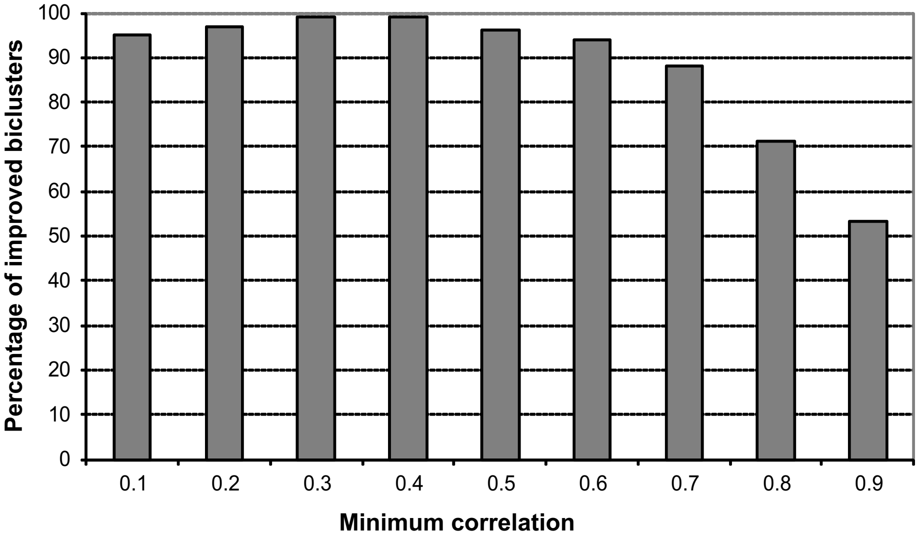


Fig. 4. Percentage of biclusters improved according to the fitness function.

possible to consider in this work because its implementation does not support datasets that have more than 64 samples.

There are many criteria that could be used to establish a comparison among different biclustering algorithms, such as the type of patterns that are sought, the number of biclusters generated or the size of the biclusters. In this paper, the results obtained by the different algorithms are compared according to the size of the biclusters, the correlation patterns and the biological information stored in the GO repository. With respect to the last criterion, the comparison methodology is based on the GO-enriched biclusters [21].

The configuration parameters of the CHCH, ISA and OPSM algorithms were the values recommended or used in the original papers. An input parameter of the CHCH algorithm is the number of biclusters to be obtained, but the ISA and OPSM algorithms do not present this option.

Table 3 shows different features of the biclusters obtained by the BISS, CHCH, ISA and OPSM algorithms. Specifically, the average of the number of genes and conditions, the average of the size of the biclusters, the smallest and the largest size of the biclusters, the average of the values of the correlation and the average of the number of pairs of negatively correlated genes are presented.

In this table, two correlations $\rho_{|\cdot|}(B)$ and $\rho(B)$ are provided, the correlation defined by Eq. (4) as well as the correlation when the absolute values are not considered in that equation. The smaller difference between these two values is, the smaller the negative correlation between pairs of genes of the biclusters is. Note that OPSM does not capture negative correlations for GaschYeast and Alzheimer datasets. Although OPSM presents a large number of pairs of negatively correlated genes for GDS1116, namely 290, 033.4, and ISA for GDS1116 and GashYeast, namely 1489.42 and 366.9, the negative correlation between pairs of genes is small due to the difference between the correlation with and without the absolute value is very small (0.95 and 0.89, 0.74 and 0.63, 0.60 and 0.52, respectively). On the other hand, it seems that ISA obtains three biclusters with a high negative correlation for Alzheimer dataset. However, these biclusters have only two conditions, and therefore, they do not provide any relevant information. Thus, it can be concluded that neither ISA nor OPSM are adequate for finding activation–inhibition patterns. Furthermore, it can be appreciated that there is a large variability in the sizes of the biclusters. For example, for GDS1116 dataset, the number of genes ranges from 22 to 300 when using the BISS algorithm. The ISA algorithm reports 60 biclusters, but some biclusters do not present relevant patterns because they have a low number of conditions. OPSM reports only 14 biclusters, and they present a large number of genes and a high correlation. However, the smallest bicluster has only 2 genes, and the largest bicluster has only 2 conditions; therefore, they do not focus on nontrivial patterns.

Fig. 5 more clearly depicts the different sizes of biclusters obtained by the BISS, CHCH, ISA and OPSM algorithms for GDS1116 dataset. A point represents a bicluster, where the number of genes is represented on the *x*-axis, and the number of conditions is represented in the *y*-axis. The OPSM approach has 5 biclusters, with more than 400 genes and too few conditions (from 2 to 6). Nevertheless, only biclusters with fewer than 400 genes are drawn in the Figure, with the goal of improving the visualization of the results. It can be observed that biclusters from the BISS approach (squares) can be classified into two clusters: one group with biclusters which have a large number of genes and another group with biclusters which have fewer than 50 genes. All the biclusters obtained by the BISS algorithm have more conditions than the biclusters obtained by the CHCH, ISA or OPSM algorithms. The ISA (circles) and CHCH (diamonds) algorithms present biclusters with fewer than 10 conditions, but the biclusters of ISA always have more than 50 genes.

Fig. 6 presents the percentage of overrepresented or enriched biclusters in one or more GO terms for the BISS, CHCH, ISA and OPSM algorithms for GDS116 and GaschYeast datasets. This percentage has been separately computed for each branch CC, MF and BP in GO and for the most commonly used significance levels 0.05% and 0.1%. It can be observed that the BISS approach obtains better results than the ISA algorithm for GaschYeast dataset and the CHCH algorithm for both datasets for the three branches of GO and for the two levels of significance. However, the results from the ISA and OPSM algorithms are better than those of the BISS method for GDS1116 dataset. The OPSM algorithm obtained a low number of biclusters, most of which are composed of a large number of positively correlated genes, as can be observed in Table 3.

The influence of the size of the biclusters on the percentage of enriched biclusters has been widely studied in the literature [44,21,18]. To avoid this impact, previous researchers [44] have considered a filtering process in the group of biclusters obtained for each algorithm before establishing a comparison based on the percentage of enriched biclusters. The filter comprises determining the maximum number of genes allowed in the biclusters and removing the biclusters that have more genes than that maximum. Specifically, 50 genes is considered to be the maximum number of genes.

Table 3
Results obtained by the BISS and the classical CHCH, ISA and OPSM algorithms for GDS1116, GaschYeast and Alzheimer datasets.

	Num. bicluster	Num. genes	Num. conditions	Min. size	Average size	Max. size	$\rho_{\|\cdot\|}(B)$	$\rho(B)$	Pairs of neg. corr. genes
GDS1116									
BISS	100	90.4	21.5	(22 × 13)	2370.29	(300 × 45)	0.65	0.21	4806.4
CHCH	100	19.0	4.68	(3 × 5)	66.9	(136 × 3)	0.49	0.00	266.9
ISA	60	182.1	5.5	(129 × 2)	1023.5	(280 × 8)	0.74	0.63	1489.42
OPSM	14	678.1	9.0	(2 × 21)	2207.1	(4186 × 2)	0.95	0.89	290,033.4
GaschYeast									
BISS	100	96.2	26.2	(20 × 18)	2680.4	(292 × 45)	0.75	0.35	3839.7
CHCH	100	70.6	19.1	(45 × 9)	1407.1	(222 × 90)	0.3	0.0	1986.9
ISA	66	76.3	8.7	(11 × 11)	645.7	(136 × 10)	0.60	0.52	366.9
OPSM	12	95.6	12.5	(4 × 18)	849.8	(387 × 7)	0.98	0.98	0.0
Alzheimer									
BISS	100	48.7	15.1	(36 × 10)	741.8	(62 × 18)	0.83	0.16	488.7
CHCH	100	13.4	6.7	(3 × 7)	170.55	(152 × 33)	0.54	0.09	34.2
ISA	3	51.67	2.0	(43 × 2)	103.3	(56 × 2)	1.0	0.1	609.67
OPSM	13	246.5	9.8	(12 × 13)	925.0	(809 × 3)	0.96	0.96	0.0

Fig. 7 shows the percentage of enriched biclusters after this filtering process for the BISS, CHCH, ISA and OPSM algorithms for GDS116 and GaschYeast datasets. Note that all the biclusters obtained by the ISA algorithm for GDS116 dataset are removed after the filtering process. It can be appreciated that the BISS algorithm obtained more enriched biclusters than ISA for GashYeast dat set and the CHCH algorithm for both datasets and for the three subcategories of GO. Moreover, BISS improved the OPSM algorithm for GDS116 dataset and it obtained similar results for GaschYeast dataset.

4.3.2. Comparison with correlation-based algorithms

The performance of the BISS algorithm is compared to BCCA [44] and BICLIC [45] algorithms in this subsection. These two methods were chosen because they also use a merit function based on correlations to search patterns from gene expression data. Moreover, the BICLIC algorithm was designed to find activation–inhibition expression patterns, which is one goal of this work.

The BCCA has two possible experimental settings depending on whether it considers overlapping. The BCCA without overlapping (BCCA-not) has the number of biclusters to be obtained and a correlation threshold as input parameters. The BCCA with overlapping (BCCA-yes) has the correlation threshold as the only input parameter. The correlation threshold depends on the data, and the size of the biclusters decreases when this threshold increases. The main difference between the two algorithms, BCCA-not and BCCA-yes, is that the BCCA-yes is an exhaustive search and thus it obtains a large number of biclusters with a high computational cost. The parameters of the BCCA-not is 100 for the number of biclusters to be obtained and 0.2 for the correlation threshold for GDS116 dataset and 0.8 for both GaschYeast and Alzheimer datasets. Note that the choice of these parameters imply a hard trial-and-error task for each dataset. For BCCA-yes, an experimental study of the correlation is not possible because of the computational cost, and a threshold of 0.85 has been selected, as it was indicated by the previous researchers [44].

The BICLIC has three input parameters: a correlation threshold and the minimum number of genes and conditions of the biclusters to be obtained. The values for these parameters are 0.9, 5 and 5 for GaschYeast dataset, 0.85, 25 and 10 for GDS116 dataset and 0.6, 5 and 2 for Alzheimer dataset. The values selected for GaschYeast dataset have been provided by the authors of BICLIC as default parameters in [45]. However, the parameters for GDS116 and Alzheimer datasets have been experimentally obtained by means of a trial-and-error task since the default parameters do not provide results.

Table 4 shows different features of the biclusters obtained using the BISS, BCCA-not, BCCA-yes and BICLIC algorithms. BISS and BCCA-not obtained 100 biclusters as it was expected. BCCA-yes and BICLIC obtained 1662, 17,322 and 368 and 5988, 14, 791 and 4405 biclusters for GDS1116, GaschYeast and Alzheimer datasets, respectively. Though the activation–inhibition patterns are important from a biological point of view [11], it can be noticed that the BCCA algorithm in both versions does not capture negative correlations among the genes; in other words, the number of

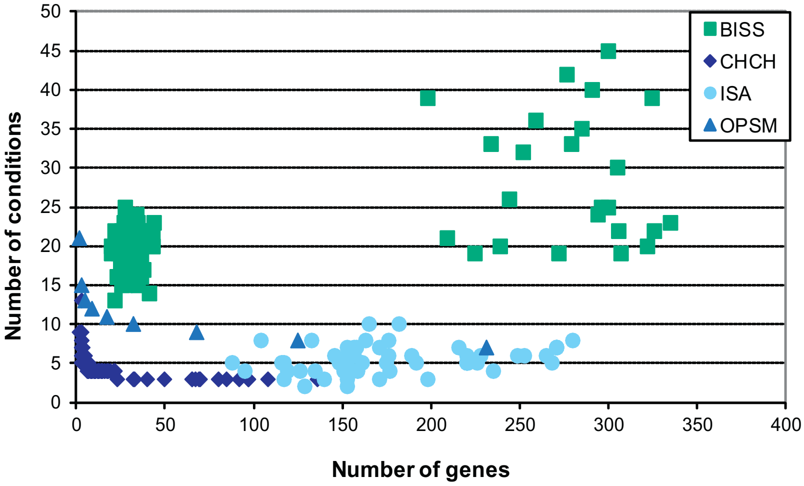


Fig. 5. Size of biclusters found by the BISS, CHCH, ISA and OPSM algorithms for GDS1116 dataset.

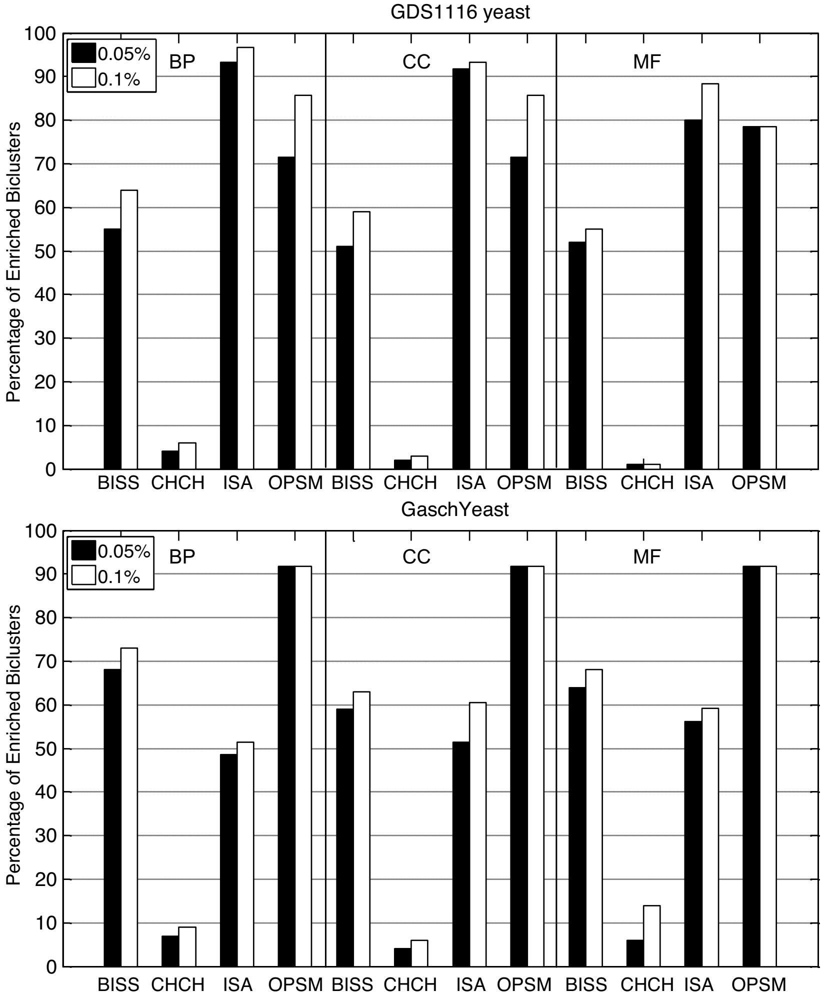


Fig. 6. Percentage of enriched biclusters obtained by BISS, CHCH, ISA and OPSM for the CC, MF and BP sub-ontologies of GO for GDS1116 and GaschYeast datasets.

negatively correlated genes is 0. This result is one of the most important differences between the BISS and the BCCA algorithms. On the other hand, BICLIC captures negative correlations for GaschYeast and Alzheimer datasets but not for GDS1116 dataset. However, it can be concluded that BICLIC obtains genes with a very low negative correlation due to the difference between the correlation with and without the absolute value is very small (0.76 and 0.61 for GaschYeast dataset and 0.69 and 0.68 for Alzheimer dataset).

Fig. 8 shows the different sizes of the biclusters obtained by the BISS, BCCA-not, BCCA-yes and BICLIC algorithms for GDS1116 dataset. It can be observed that BCCA-not, BCCA-yes and BICLIC provided biclusters with a number of conditions larger and a number of genes lower than that of the BISS approach or a number of

Table 4
Results obtained by the BISS, the BCCA-not, BCCA-yes and BICLIC algorithms for GDS1116, GaschYeast and Alzheimer datasets.

	Num. bicluster	Num. genes	Num. conditions	Min. size	Average size	Max. size	$\rho_{\lvert\cdot\rvert}(B)$	$\rho(B)$	Pairs of neg. corr. genes
GDS1116									
BISS	100	90.4	21.5	(22 × 13)	2370.29	(300 × 45)	0.65	0.21	4806.4
BCCA-not	100	16.5	122.7	(3 × 122)	2022.5	(90 × 122)	0.36	0.36	0.0
BCCA-yes	1662	10.4	86.3	(2 × 18)	943.0	(32 × 114)	0.85	0.85	0.0
BICLIC	5988	76.7	14.9	(25 × 10)	940.5	(685 × 10)	0.85	0.85	0.0
GaschYeast									
BISS	100	96.2	26.2	(20 × 18)	2680.4	(292 × 45)	0.75	0.35	3839.7
BCCA-not	100	2.6	22.7	(2 × 8)	54.0	(15 × 19)	0.77	0.77	0.0
BCCA-yes	17,322	59.4	97.7	(2 × 3)	5906.5	(121 × 163)	0.9	0.9	0.0
BICLIC	14,791	178.9	29.2	(5 × 6)	2249.3	(139 × 159)	0.76	0.61	5797.1
Alzheimer									
BISS	100	48.7	15.1	(36 × 10)	741.8	(62 × 18)	0.83	0.16	488.7
BCCA-not	100	9.1	14.8	(2 × 8)	97.8	(24 × 29)	0.86	0.86	0.0
BCCA-yes	368	8.9	16.1	(2 × 5)	118.8	(16 × 23)	0.88	0.88	0.0
BICLIC	4405	516.1	14.2	(5 × 13)	5888.9	(899 × 25)	0.69	0.68	2799.9

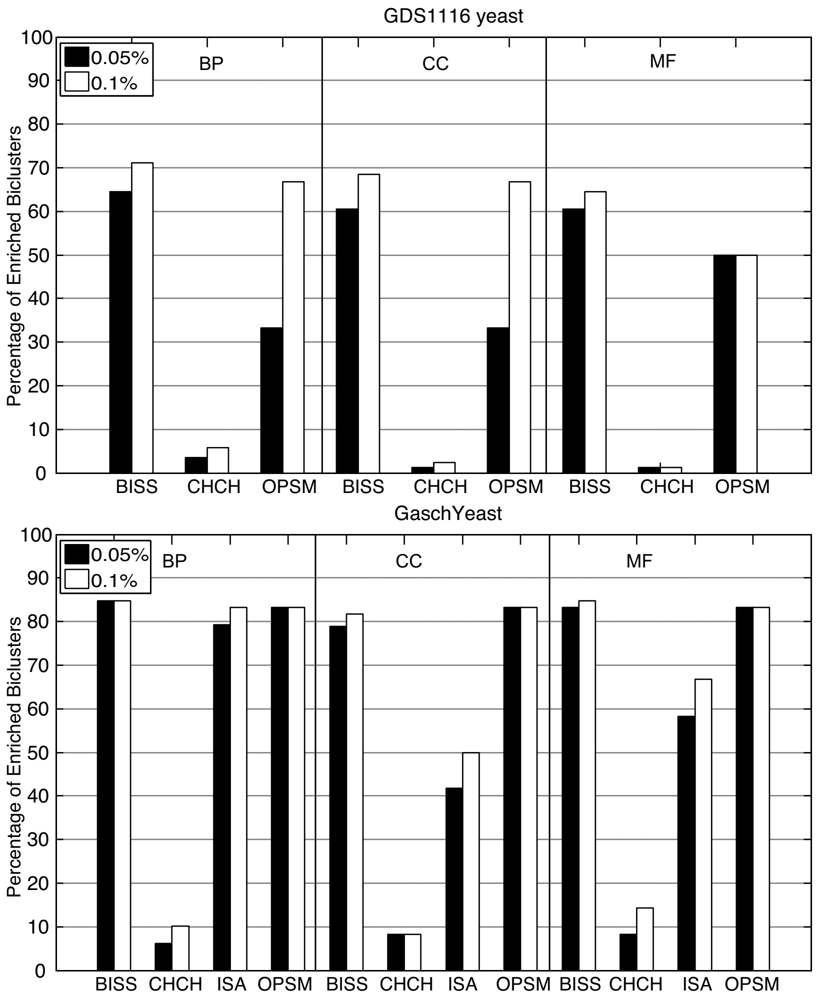


Fig. 7. Percentage of enriched biclusters obtained by BISS, CHCH, ISA and OPSM for the BP, CC and MF sub-ontologies of GO for GDS1116 and GaschYeast datasets after applying the filter.

genes larger but a number of conditions lower in the case of several biclusters obtained by BICLIC. Note that the biclusters obtained by the BISS and the BCCA-not, BCCA-yes and BICLIC algorithms are different and capture different types of patterns. For this reason, the biclusters are represented in different areas in the picture.

Fig. 9 presents the percentage of overrepresented or enriched biclusters in one or more GO terms for the BISS, BCCA-not, BCCA-yes and BICLIC algorithms for GDS1116 and GashYeast datasets when applying the filter described in Section 4.3.1, that is, only the biclusters with a number of genes lesser than 50 genes are considered. It can be observed that BISS improves BCCA-not, BCCA-yes and BICLIC for the three categories of GO and for the two levels of significance for GaschYeast dataset. Moreover, BISS presents better results than the BCCA-not, BCCA-yes and BICLIC in the CC and MF subbranches of GO for GDS1116 dataset. In the BP component, the results from the BISS and the BCCA-yes algorithms are similar for the level of significance 0.1% but BCCA-yes is better than BISS for the level of significance 0.05%. Nevertheless, BCCA-yes is unable to obtain negative correlation patterns.

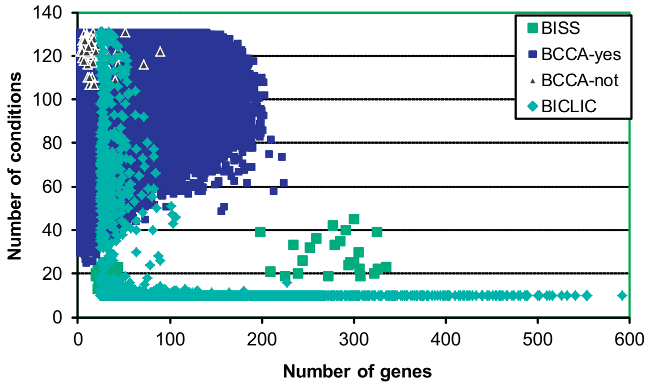


Fig. 8. Size of biclusters found by the BISS, BCCA-not, BCCA-yes and BICLIC algorithms for GDS1116 dataset.

A more challenging definition for an enriched bicluster can be established to emphasize the difference between the BISS and the BCCA-yes algorithm. Most of the time, the majority of the functionally enriched biclusters have a low number of annotated genes in GO with respect to their total number of genes. Thus, a bicluster is said to be *highly enriched* if the number of genes that share the same function in a GO term is greater than a given threshold. In this work, a minimum of 5 genes has been considered [55,56].

Fig. 10 presents the percentage of highly enriched biclusters for the BISS, BCCA-yes and BICLIC algorithms. The results obtained by the BISS are better than those of the BCCA-yes and BICLIC algorithms for BP and CC components for GDS1116 dataset. In component MF, BISS has 27.6% and 28.9% for level of significance 0.1% and 0.05% respectively, BCCA-yes has 27.5% for both levels and BICLIC has 30.4% and 30.8%. In addition, it can be observed that

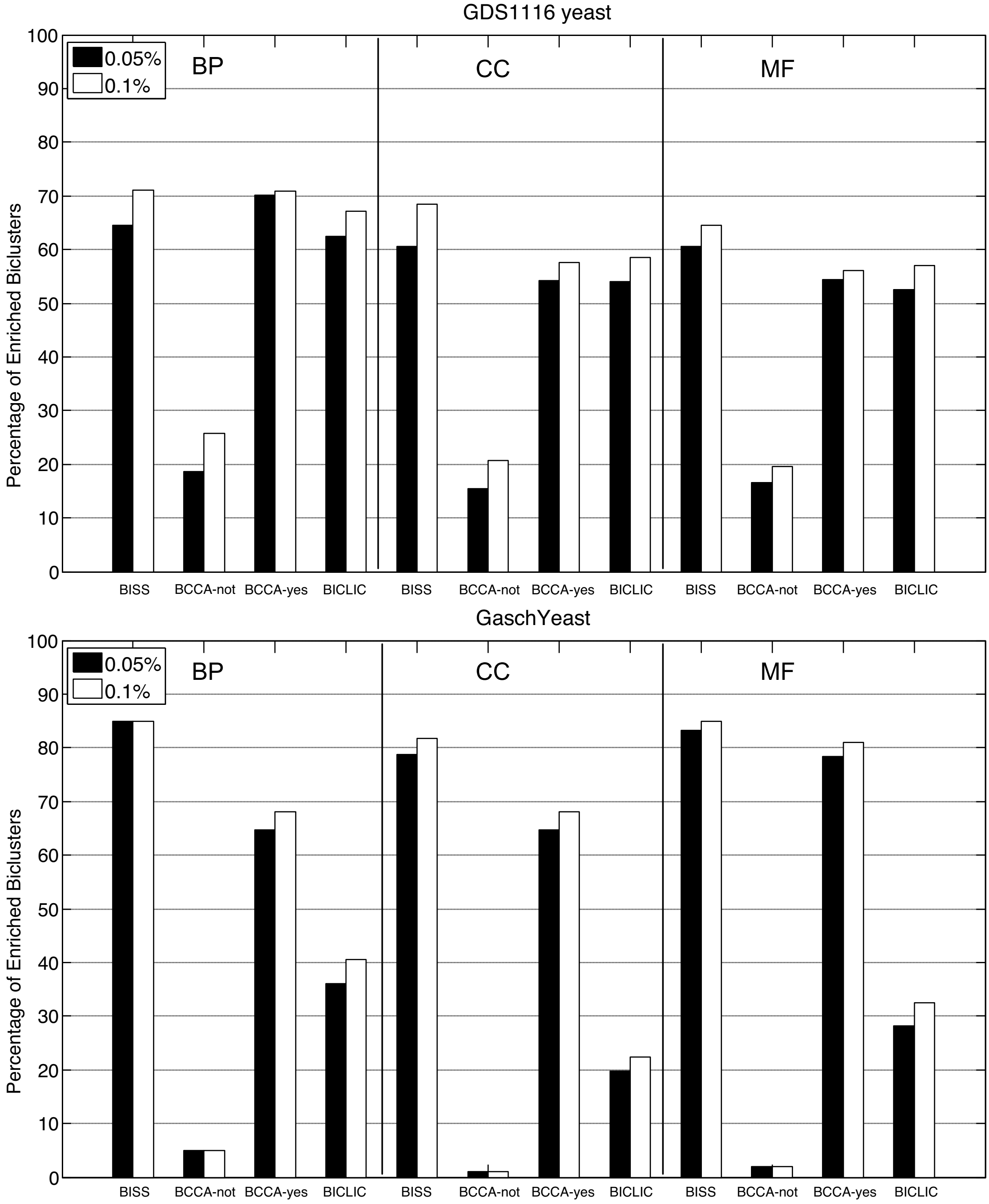


Fig. 9. Percentage of enriched biclusters obtained by BISS, BCCA-not, BCCA-yes and BICLIC for the BP, CC and MF sub-ontologies of GO for GDS1116 and GaschYeast datasets.

results obtained by BISS and BCCA-yes algorithms are similar and always better than BICLIC for GaschYeast data set.

4.3.3. Biological significance of biclusters

A study of the biological significance of several biclusters obtained by the BISS algorithm is presented in this subsection.

Fig. 11 shows the profile of a bicluster composed of 35 genes and 13 conditions obtained by the BISS algorithm for Alzheimer dataset. From this figure, both shifting and scaling patterns, and moreover, negative and positive correlation patterns can be observed. In particular, two positively correlated genes are plotted using dash lines and a third gene showing negative correlation with the two aforementioned genes is plotted using a bold line.

Table 5 reports the biological significance of several biclusters obtained by the BISS algorithm for GDS1116 and GaschYeast datasets. The FuncAssociate web-based tool has been used to generate the information presented in the table.[3] In particular, the size of the bicluster and the number of genes of the bicluster associated with a GO term, the number of genes in the GO term, the value of the adjusted p-value and the description of the GO term for two GO terms are shown. It can be observed that 22 and 12 genes of the bicluster # 1 are found in the GO:0022626 and GO:0022627 terms, which are composed of 163 and 64 genes, respectively. These GO

[3] http://llama.mshri.on.ca/funcassociate/.

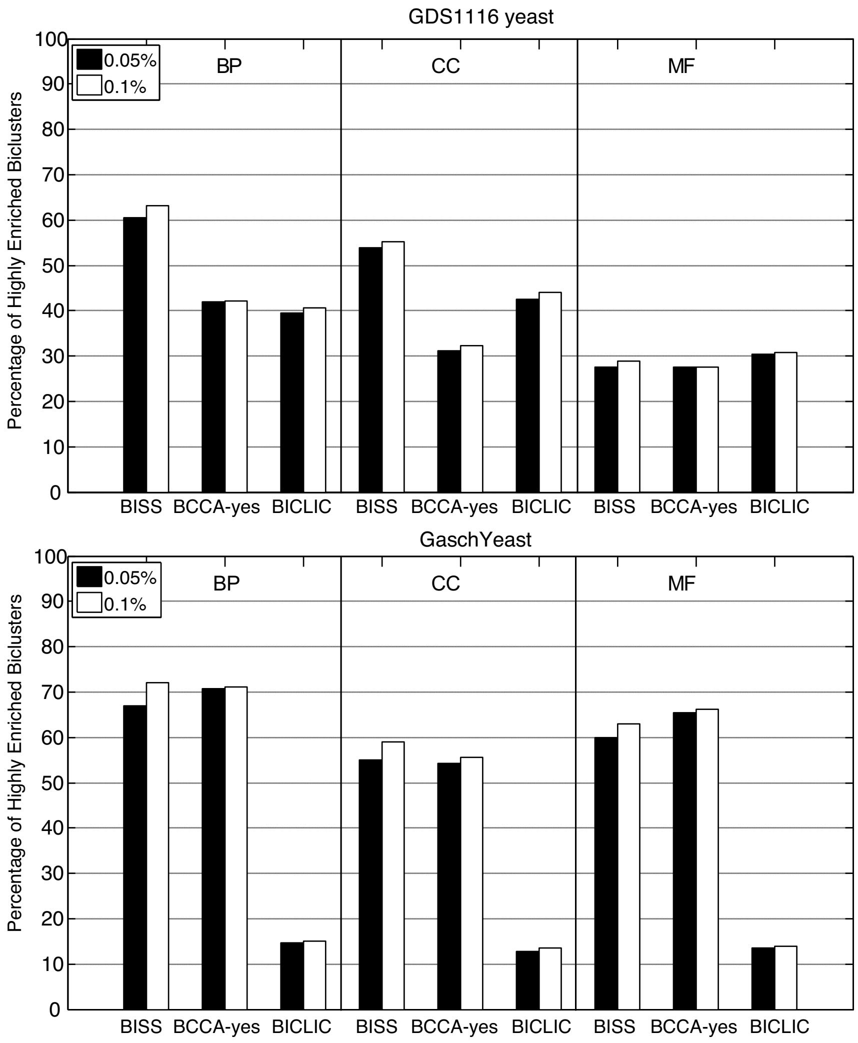


Fig. 10. Percentage of highly enriched biclusters obtained by BISS, BCCA-yes and BICLIC for the BP, CC and MF sub-ontologies of GO for GDS1116 and GaschYeast datasets.

terms belong to the cellular component subontology of GO and both are related to ribosomes located in the cytosol. Bicluster # 70 has 4 genes associated with GO:0000722 and GO:0003678 terms related to DNA metabolic processes such as the telomere maintenance and DNA helicase activity. On the other hand, biclusters # 1 and # 62 from GaschYeast dataset are related to ribosomal functionality, in particular, the half of the genes approximately are related to RNA metabolic processes.

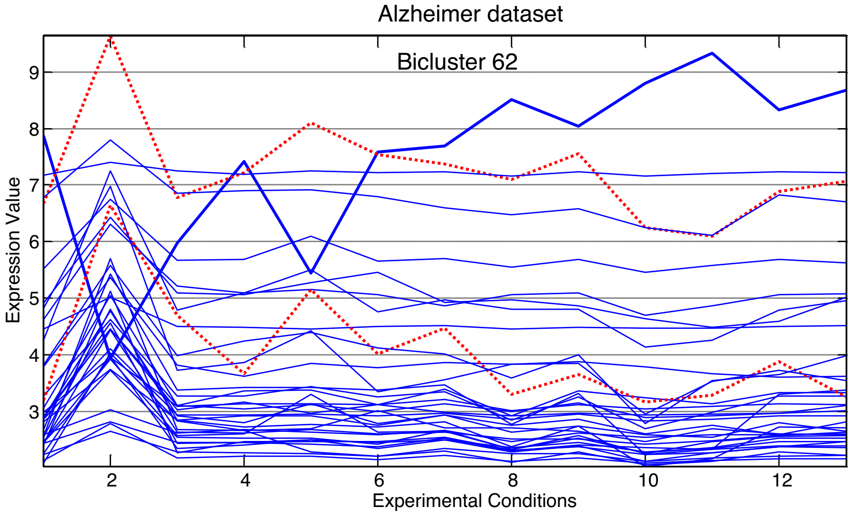


Fig. 11. Bicluster profile obtained by the BISS algorithm for Alzheimer dataset.

Table 5
Biological significance of several biclusters obtained by BISS for GDS1116 and GaschYeast datasets.

	Bicluster size	Genes from bi. in GO term	Genes in GO term	*p*-Value	GO term &description
GDS1116					
Bicluster #1	(38,22)	22	163	<0.001	GO:0022626 cytosolic ribosome
		12	64	<0.001	GO:0022627 cytosolic small ribosomal subunit
Bicluster #70	(24,19)	4	19	<0.001	GO:0000722 telomere maintenance via recombination
		4	36	0.003	GO:0003678 DNA helicase activity
GaschYeast					
Bicluster #1	(51,30)	12	64	<0.001	GO:0022627 cytosolic small ribosomal subunit
		23	163	<0.001	GO:0022626 cytosolic ribosome
Bicluster #62	(44,20)	21	483	<0.001	GO:0034660 ncRNA metabolic process
		19	426	<0.001	GO:0034470 ncRNA processing

4.3.4. *Computational complexity*

The computational complexity of the BISS algorithm mainly depends on the fitness function complexity (Eq. (5)) and the set of operations during the searching procedure (Algorithm 1).

The main term in the fitness function is the average correlation of the bicluster B (Eq. (4)), which is based on the correlation between pairs of genes in B (Eq. (3)). The complexity of the correlation between two genes is $O(m)$, where m is the number of conditions in B. Therefore, the complexity of the fitness function is $O(n^2m)$, where n is the number of genes in B. Note that in the worst case n and m are equal to the number of genes and conditions in the microarray, respectively. However, this case is almost unlike due to the biclusters are built in the initial population with a limit for the number of genes and conditions (Algorithm 1, line 4).

The BISS algorithm complexity is set to $O(k \cdot n^2m)$, where k is a constant such that $k = k_{numBi} \times k_{numIter} \times k_{convergence}$. Firstly, k_{numBi} is the number of biclusters to find which is an input parameter (Algorithm 1, line 3). Secondly, $k_{numIter}$ is a constant to control the number of times that the *reference set* is rebuilt (Algorithm 1, line 9). This constant is an inner parameter of the algorithm, which is usually chosen equal to 20 following the scatter search references [14]. Finally, $k_{convergence}$ is a constant to guarantee the convergence of the optimization process (Algorithm 1, line 10). Note that this constant is not explicitly written in the algorithm because the reference set is always stabilized before this constant is reached.

The complexity of CHCH and OPSM algorithms is $O(nm)$ and $O(nm^3 \cdot l)$, respectively, where l is the number of biclusters that OPSM reports. The ISA algorithm is lineal with respect to the number of genes and conditions but also regarding the seed that is used, and hence, a good initial parametrization is required. On the other hand, the complexity of the BCCA, which is also based on correlations, is $O(n^5)$. Although the BICLIC complexity is not provided, the experiments indicate that the algorithm running time is also high.

5. Conclusions

The BISS algorithm for searching local patterns in gene expression data has been proposed in this paper. This algorithm uses a merit function based on linear correlations among genes and a scatter search metaheuristic. The main motivation to use linear correlations among genes is that correlated genes imply co-expressed genes. One of the most important characteristics of the BISS algorithm is the type of discovered patterns. The BISS captures negative correlation patterns and not only positively correlated genes as other methods based on correlations. Negative correlation patterns have been recently considered in gene expression studies due to their biological relevance since they are a common feature in many molecular pathways. The BISS has been compared with a group of classical biclustering methods, including CHCH, ISA and OPSM, and with the recently published BCCA and BICLIC approaches, which are based on correlations. Firstly, experiments reported that BISS obtained good results when comparing with the benchmark algorithms and, secondly, that BISS captured negative correlation patterns unlike the BCCA and it improved the results of BICLIC which also captures negative correlations. On the other hand, the results show that the number of conditions of biclusters obtained by the BISS is greater than those obtained by CHCH, ISA and OPSM. Additionally, it can be concluded that the functional enrichment of the biclusters obtained by the BISS improves when the biclusters have less than 50 genes. Therefore, the penalization in the fitness function of the BISS to control the volume of the biclusters is crucial. In the case of biclusters with less than 50 genes, the BISS is better than the CHCH, ISA, OPSM and BICLIC and better than or similar to the BCCA algorithm.

Future research will be focused on different topics. First, other improvement and combination methods or solution codifications can be studied using the scatter search metaheuristic. Moreover, the integration of new measures in the fitness function as mutual information or other measures that explore the inner topological properties of the network induced by the correlation among the genes will be also studied.

Algorithm 1. BISS: BIclustering with scatter search

```
INPUT microarray $D$, number of biclusters to be found $numBi$, parameter to
  control the size of biclusters $M$
OUTPUT Set $Results$ with $numBi$ biclusters.
begin
 1:  $num \leftarrow 0$, $Results \leftarrow \emptyset$
 2:  $\rho \leftarrow$ MinimumCorrelation($D$,$M$)
 3:  while ($num < numBi$) do
 4:    $Population \leftarrow$ DiversificationGeneration()
 5:    $Population \leftarrow$ Improvement($Population$,$\rho$)
 6:    $Reference\ Set \leftarrow$ Build($Population$)
 7:    $Population \leftarrow Population \setminus Reference\ Set$
 8:    $i \leftarrow 0$
 9:    while ($i < numIter$) do
10:      while (NOT stable) do
11:        $A \leftarrow$ SubsetGeneration($Reference\ Set$)
12:        $B \leftarrow$ SolutionCombination($A$)
13:        $B \leftarrow$ Improvement($B$,$\rho$)
14:        $Reference\ Set \leftarrow$ Update($B$,$Reference\ Set$)
15:      end while
16:      $Reference\ Set \leftarrow$ Rebuild($Population$,$Reference\ Set$)
17:      $Population \leftarrow Population \setminus Reference\ Set$
18:      $i \leftarrow i + 1$
19:    end while
20:    $Bicluster \leftarrow$ the best one from $Reference\ Set$
21:    $Results \leftarrow Results \cup \{Bicluster\}$
22:    $num \leftarrow num + 1$
23:  end while
end
```

Algorithm 2. Improvement method

```
INPUT Bicluster $B = [g_1, \ldots, g_N]$, minimum correlation $\rho$
OUTPUT Bicluster $B' \subseteq B$ such that $|\rho(g_i, g_j)| \geq \rho \quad \forall g_i, g_j \in B'$
begin
  $i \leftarrow 1, B' \leftarrow \{g_i\}, R \leftarrow \{\}$
  while $(i < N)$ do
    $j \leftarrow i + 1$
    while $(j \leq N)$ do
      if $(|\rho(g_i, g_j)| \geq \rho)$ then
        if $(g_j \notin R)$ then
          $B' \leftarrow B' \cup \{g_j\}$
        end if
      else
          $R \leftarrow R \cup \{g_j\}$
        end if
        $j \leftarrow j + 1$
      end while
      $i \leftarrow i + 1$
  end while
end
```

Acknowledgements

We would like to thank Spanish Ministry of Science and Innovation, Junta de Andalucía and University Pablo de Olavide for the financial support under projects TIN2011-28956-C02-00, P12-TIC-1728 and APPB813097, respectively.

References

[1] P. Brown, D. Botstein, Exploring the new world of the genome with DNA microarrays, Nat. Genet. 21 (1999) 33–37.

[2] M. Florian, S. Rainer, Inferring cellular networks – a review, BMC Bioinf. 8 (2007) S5.

[3] J. Morgan, J. Sonquistz, Problems in the analysis of survey data, and a proposal, J. Am. Stat. Assoc. 58 (1963) 415–434.

[4] J. Hartigan, Direct clustering of a data matrix, J. Am. Stat. Assoc. 67 (1972) 123–129.

[5] B. Mirkin, Mathematical Classification and Clustering, Academic Press, Boston-Dordrecht, 1996.

[6] I. Dhillon, S. Mallela, D. Modha, Information-theoretic co-clustering, in: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York, NY, USA, 2003, pp. 89–98.

[7] R. Harpaz, R. Haralick, Mining subspace correlations, in: Proceedings of IEEE Symposium on Computational Intelligence and Data Mining, 2007, pp. 335–342.

[8] S. Madeira, A. Oliveira, Biclustering algorithms for biological data analysis: a survey, IEEE Trans. Comput. Biol. Bioinf. 1 (2004) 24–45.

[9] Y. Cheng, G. Church, Biclustering of expression data, in: Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology, vol. 8, 2000, pp. 93–103.

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

[11] T. Zeng, J. Li, Maximization of negative correlations in time-course gene expression data for enhancing understanding of molecular pathways, Nucl. Acids Res. 38 (2010) e1.

[12] V. Subbarayan, et al., Inverse relationship between 15-lipoxygenase-2 and PPAR-gamma gene expression in normal epithelia compared with tumor epithelia, Neoplasia 7 (2005) 280–293.

[13] G. Fontanini, et al., Bcl-2 protein: a prognostic factor inversely correlated to p53 in non-small-cell lung cancer, Brit. J. Cancer 71 (1995) 1003–1007.

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

[15] A. Tanay, R. Sharan, R. Shamir, Biclustering algorithms: a survey, in: Handbook of Computational Molecular Biology, Chapman and Hall, 2005.

[16] S. Busygin, O. Prokopyev, P. Pardalos, Biclustering in data mining, Comput. Oper. Res. 35 (2008) 2964–2987.

[17] L. Li, Y. Guo, W. Wu, Y. Shi, J. Cheng, S. Tao, A comparison and evaluation of five biclustering algorithms by quantifying goodness of biclusters for gene expression data, BioData Min. 5 (2012) 8.

[18] K. Eren, M. Deveci, O. Kucuktunc, U.V. Catalyurek, A comparative analysis of biclustering algorithms for gene expression data, Brief. Bioinf. 14 (2013) 279–292.

[19] R. Santamaría, L. Quintales, R. Therón, Methods to bicluster validation and comparison in microarray data, in: Proceedings of Intelligent Data Engineering and Automated Learning – IDEAL 2007, volume 4881 of Lecture Notes in Computer Science, Springer, Berlin/Heidelberg, 2007, pp. 780–789.

[20] D. Bozdağ, A.S. Kumar, U.V. Catalyurek, Comparative analysis of biclustering algorithms, in: Proceedings of the 1st ACM International Conference on Bioinformatics and Computational Biology, BCB '10, ACM, New York, NY, USA, 2010, pp. 265–274.

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

[22] G. Getz, E. Levine, E. Domany, Coupled two-way clustering analysis of gene microarray data, Proc. Natl. Acad. Sci. U. S. A. 97 (2000) 12079–12084.

[23] J. Yang, H. Wang, W. Wang, P. Yu, Enhanced biclustering on expression data, in: Proceedings of 3rd IEEE Symposium on Bioinformatics and Bioengineering, 2003, pp. 321–327.

[24] S. Bergmann, J. Ihmels, N. Barkai, Iterative signature algorithm for the analysis of large-scale gene expression data, Phys. Rev. E 67 (2003) 1–18.

[25] A. Ben-Dor, B. Chor, R. Karp, Z. Yakhini, Discovering local structure in gene expression data: the order-preserving submatrix problem, J. Comput. Biol. 10 (2003) 373–384.

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

[27] L. Lazzeroni, A. Owen, Plaid models for gene expression data, Stat. Sin. 12 (2002) 61–86.

[28] Y. Kluger, R. Basri, J. Chang, M. Gerstein, Spectral biclustering of microarray data: coclustering genes and conditions, Genome Res. 13 (2003) 703.

[29] R. Harpaz, R. Haralick, Exploiting the geometry of gene expression patterns for unsupervised learning, in: Proceedings of 18th International Conference on Pattern Recognition – ICPR 2006, vol. 2, 2006, pp. 670–674.

[30] X. Gan, A. Liew, H. Yan, Discovering biclusters in gene expression data based on high-dimensional linear geometries, BMC Bioinf. 9 (2008) 1–15.

[31] S. Bleuler, A. Prelic, E. Zitzler, An EA framework for biclustering of gene expression data, in: Proceedings of Congress on Evolutionary Computation, 2004 – CEC2004, vol. 1, 2004.

[32] F. Divina, J. Aguilar-Ruiz, Biclustering of expression data with evolutionary computation, IEEE Trans. Knowl. Data Eng. 18 (5) (2006) 590–602.

[33] C.A. Gallo, J.A. Carballido, I. Ponzoni, Microarray biclustering: a novel memetic approach based on the PISA platform, in: Proceedings of the 7th European Conference on Evolutionary Computation, Machine Learning and Data Mining – EvoBIO 2009, 2009, pp. 44–55.

[34] H. Banka, S. Mitra, Evolutionary biclustering of gene expressions, Ubiquity 7 (2006) 1–12.

[35] F. Divina, J. Aguilar-Ruiz, A multi-objective approach to discover biclusters in microarray data, in: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, ACM Press, New York, NY, USA, 2007, pp. 385–392.

[36] S. Dharan, A. Nair, Biclustering of gene expression data using reactive greedy randomized adaptive search procedure, BMC Bioinf. 10 (2009) S27.

[37] K. Bryan, Biclustering of expression data using simulated annealing, in: Proceedings of the 18th IEEE International Symposium on Computer-Based Medical Systems, USA, 2005, pp. 383–388.

[38] J. Liu, Z. Li, X. Hu, Y. Chen, Biclustering of microarray data with mospo based on crowding distance, BMC Bioinf. 10 (2009) S9.

[39] F. Liu, H. Zhou, J. Liu, G. He, Biclustering of gene expression data using EDA-GA hybrid, in: Proceedings of IEEE Congress on Evolutionary Computation – CEC 2006, 2006, pp. 1598–1602.

[40] W. Ayadi, M. Elloumi, J.-K. Hao, A biclustering algorithm based on a bicluster enumeration tree: application to dna microarray data, BioData Min. 2 (2009) 9.

[41] H.Y. Wen-Hui Yang, Dao-Qing Dai, Finding correlated biclusters from gene expression data, in: IEEE Trans. Knowl. Data Eng. IEEE Comput. Soc. Dig. Lib. IEEE Comput. Soc., 2010, pp. 568–584.

[42] G. Li, Q. Ma, H. Tang, A.H. Paterson, Y. Xu, Qubic: a qualitative biclustering algorithm for analyses of gene expression data, Nucl. Acids Res. 37 (2009) e101.

[43] D. Mishra, A. Rath, CPB: a model for biclustering, in: Proceedings of International Conference on Information Management and Engineering, 2009 – ICIME '09, 2009, pp. 629–632.

[44] A. Bhattacharya, R.K. De, Bi-correlation clustering algorithm for determining a set of co-regulated genes, Bioinformatics 25 (2009) 2795–2801.

[45] T. Yun, G.-S. Yi, Biclustering for the comprehensive search of correlated gene expression patterns using clustered seed expansion, BMC Genomics 14 (2013) 144.

[46] J.A. Nepomuceno, A. Troncoso, J. Aguilar-Ruiz, Biclustering of gene expression data by correlation-based scatter search, BioData Min. 4 (2011) 3.

[47] D.S. Rodriguez-Baena, A.J. Perez-Pulido, J.S. Aguilar-Ruiz, A biclustering algorithm for extracting bit-patterns from binary datasets, Bioinformatics 27 (2011) 2738–2745.

[48] S.C. Madeira, M.C. Teixeira, I. Sá-Correia, A.L. Oliveira, Identification of regulatory modules in time series gene expression data using a linear time biclustering algorithm, IEEE Trans. Comput. Biol. Bioinf. 7 (2010) 153–165.

[49] S.C. Madeira, A.L. Oliveira, A polynomial time biclustering algorithm for finding approximate expression patterns in gene expression time series, Algorithms Mol. Biol. 4 (2009) 8.

[50] N. Gupta, S. Aggarwal, Mib Using mutual information for biclustering gene expression data, Pattern Recognit. 43 (2010) 2692–2697.

[51] C. Huttenhower, K.T. Mutungu, N. Indik, W. Yang, M. Schroeder, J.J. Forman, O.G. Troyanskaya, H.A. Coller, Detailing regulatory networks through large scale data integration, Bioinformatics 25 (2009) 3267–3274.

[52] J.A. Nepomuceno, A. Troncoso, J. Aguilar-Ruiz, A local search in scatter search for improving biclusters, in: NABIC 3rd Congress on Natural and Biologically Inspired Computing, 2011, pp. 521–526.

[53] M. Ray, J. Ruan, W. Zhang, Variations in the transcriptome of Alzheimer's disease reveal molecular networks involved in cardiovascular diseases, Genome Biol. 9 (2008) R148.

[54] K.L. Brem RB, The landscape of genetic complexity across 5700 gene expression traits in yeast, Proc. Natl. Acad. Sci. U. S. A. 5 (2005) 1572–1577.

[55] F.M. Al-Akwaa, Y.M. Kadah, An automatic gene ontology software tool for bicluster and cluster comparisons, in: Proceedings of the 6th Annual IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology – CIBCB '09, IEEE Press, Piscataway, NJ, USA, 2009, pp. 163–167.

[56] C.I. Castillo-Davis, D.L. Hartl, Genemerge-post-genomic analysis, data mining, and hypothesis testing, Bioinformatics 19 (2003) 891–892.

[57] S. Barkow, S. Bleuler, A. Prelic, P. Zimmermann, E. Zitzler, Bicat: a biclustering analysis toolbox, Bioinformatics 22 (2006) 1282–1283.

[58] T. Murali, S. Kasif, Extracting conserved gene expression motifs from gene expression data, in: Proceedings of Pacific Symposium on Biocomputing, 2003, pp. 77–88.