Journal of Biomedical Informatics 57 (2015) 163–180

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

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

Methodological Review

Biclustering on expression data: A review

Beatriz Pontes [a,*], Raúl Giráldez [b], Jesús S. Aguilar-Ruiz [b]

[a] Department of Languages and Computer Systems, University of Seville, Seville, Spain
[b] School of Engineering, Pablo de Olavide University, Seville, Spain

ARTICLE INFO

Article history:
Received 22 January 2015
Revised 22 June 2015
Accepted 30 June 2015
Available online 6 July 2015

Keywords:
Microarray analysis
Gene expression data
Biclustering techniques

ABSTRACT

Biclustering has become a popular technique for the study of gene expression data, especially for discovering functionally related gene sets under different subsets of experimental conditions. Most of biclustering approaches use a measure or cost function that determines the quality of biclusters. In such cases, the development of both a suitable heuristics and a good measure for guiding the search are essential for discovering interesting biclusters in an expression matrix. Nevertheless, not all existing biclustering approaches base their search on evaluation measures for biclusters. There exists a diverse set of biclustering tools that follow different strategies and algorithmic concepts which guide the search towards meaningful results. In this paper we present a extensive survey of biclustering approaches, classifying them into two categories according to whether or not use evaluation metrics within the search method: biclustering algorithms based on evaluation measures and non metric-based biclustering algorithms. In both cases, they have been classified according to the type of meta-heuristics which they are based on.

1. Introduction

Technological advances in genomic offer the possibility of completely sequentialize the genome of some living species. The use of microarray techniques allows to measures the expression levels of thousands of genes under several experimental conditions. Usually, the resulting data is organized in a numerical matrix, called Expression Matrix [1]. Each element of the this data matrix denotes the numerical expression level of a gene under a certain experimental condition. With the development of microarray techniques, the interest in extracting useful knowledge from gene expression data has experimented an enormous increase, since the analysis of this information can allow discovering or justifying certain biological phenomena [2].

Various machine learning techniques have been applied successfully to this context [3]. Clustering techniques aim at finding groups of genes that present a similar variation of expression level under all the experimental conditions. If two different genes show similar expression tendencies across the samples, this suggests a common pattern of regulation, possibly reflecting some kind of interaction or relationship between their functions [1].

Yet despite their usefulness, the use of clustering algorithms has an important drawback, since they consider the whole set of samples. Nevertheless, genes are not necessarily related to every sample, but they might be relevant only for a subset of samples. This aspect is fundamental for numerous problems in the Biomedicine field [4]. Thus, clustering should be simultaneously performed on both dimensions, genes and conditions. Another restriction of the clustering techniques is that each gene must be clustered into exactly one group. However, many genes may belong to several clusters depending on their influence in different biological processes [5]. These drawbacks are solved by biclustering techniques, which have also been widely applied to gene expression data [6–10]. Biclustering was introduced in the 1970s by Hartigan [11], although Cheng and Church [12] were the first to apply it to gene expression data analysis. Other names such as co-clustering, bi-dimensional clustering, two-way clustering or subspace clustering often refer to the same problem formulation.

Tanay et al. [13] proved that biclustering is an NP-hard problem, and therefore much more complex than clustering [14]. Therefore, most of the proposed methods are based on optimization procedures as the search heuristics. The development an effective heuristic as well as the use of a suitable cost function for guiding the search are critical factors for finding significant biclusters in a micriarray. Nevertheless, not all existing biclustering approaches base their search on evaluation measures for biclusters. There exists a diverse set of biclustering tools that follow different strategies and algorithmic concepts which guide the search towards meaningful results.

* Corresponding author.
E-mail addresses: bepontes@us.es (B. Pontes), giraldez@upo.es (R. Giráldez), aguilar@upo.es (J.S. Aguilar-Ruiz).

http://dx.doi.org/10.1016/j.jbi.2015.06.028
1532-0464/© 2015 The Authors. Published by Elsevier Inc.

This paper provides a review of a large number of biclustering approaches existing in the literature and a classification which separates them into two main categories: biclustering algorithm based on evaluation measures, turn grouped according to their properties type of meta-heuristics in which they are based on; and non metric-based biclustering algorithms, turn grouped attending to their most distinctive property. In both cases we have focused on classical biclustering strategies, thus excluding in this study different specializations existing in the literature, such as biclustering based on a previous matrix binarization or biclustering for temporal series.

Next section presents an unified notation for bicluster representation, and a description of the different kind of expression patterns which biclustering algorithms aim at finding in their solutions. Third and fourth sections survey most important existing biclustering algorithms, based or not on the use of evaluation measures within the search, respectively. In both sections they have been classified according to the type of meta-heuristics in which they have been based on. Finally, a discussion on the methods under study is provided in the last section, together with the main conclusions derived from this work.

2. Definitions

Biclusters are represented in the literature in different ways, where genes can be found either in rows or columns, and different names refer the same expression sub-matrix.

Let, from now on, $\mathcal{B}$ be a bicluster consisting of a set I of $|I|$ genes and a set J of $|J|$ conditions, in which b_{ij} refers to the expression level of gene i under sample j. Then $\mathcal{B}$ can be represented as follows:

$$\mathcal{B} = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1|J|} \\ b_{21} & b_{22} & \dots & b_{2|J|} \\ \vdots & \vdots & \ddots & \vdots \\ b_{|I|1} & b_{|I|2} & \dots & b_{|I||J|} \end{pmatrix}$$

where the gene g_i is the ith row, i.e., $g_i = \{b_{i1}, b_{i2}, \dots, b_{i|J|}\}$, and condition c_j is the jth column, i.e., $c_j = \{b_{1j}, b_{2j}, \dots, b_{|I|j}\}$.

Genes and samples means in biclusters are frequently used in several evaluation measure definitions. We represent these values as b_{iJ} and b_{Ij} referring to the i row (gene) and j column (sample) means, respectively. Furthermore, the mean of all the expression values in $\mathcal{B}$ is referred to as b_{IJ}. Note that these definitions above may alter the original authors' notations in the contributions reviewed in this paper.

2.1. Bicluster taxonomy based on gene expression patterns

Several types of biclusters have been described and categorized in the literature, depending on the pattern exhibited by the genes across the experimental conditions [15]. For some of them it is possible to represent the values in the bicluster using a formal equation. We define the following elements: π represents any constant value for $\mathcal{B}$; $\beta_i (1 \leqslant i \leqslant |I|)$ and $\beta_j (1 \leqslant j \leqslant |J|)$ refer to constant values used in additive models for each gene i or condition j; and $\alpha_i, (1 \leqslant i \leqslant |I|)$ and $\alpha_j, (1 \leqslant j \leqslant |J|)$ correspond to constant values used in multiplicative models for each experimental gene i or condition j. Thus, biclusters can be categorized in the follows types:

- **Constant values.** A bicluster with constant values reveals subsets of genes with similar expression values within a subset of conditions. This situation may be expressed by: $b_{ij} = \pi$.
- **Constant values on rows or columns.** A bicluster with constant values in the rows/columns identifies a subset of genes/-conditions with similar expression levels across a subset of conditions/genes. Expression levels might therefore vary from gene to gene or from condition to condition. It can also be expressed either in an additive or multiplicative way:
  - Additive: $b_{ij} = \pi + \beta_i, b_{ij} = \pi + \beta_j$
  - Multiplicative: $b_{ij} = \pi \times \alpha_i, b_{ij} = \pi \times \alpha_j$
- **Coherent values on both rows and columns.** This kind of biclusters identifies more complex relations between genes and conditions, either in an additive or multiplicative way:
  - Additive: $b_{ij} = \pi + \beta_i + \beta_j$
  - Multiplicative: $b_{ij} = \pi \times \alpha_i \times \alpha_j$
- **Coherent evolutions.** Evidence that a subset of genes is up-regulated or down-regulated across a subset of conditions without taking into account their actual expression values. In this situation, data in the bicluster does not follow any mathematical model.

According to the former definitions, it is possible to formally describe two kind of patterns summarizing all the previous situations: shifting and scaling patterns [16]. They have been defined using numerical relations among the values in a bicluster.

A bicluster $\mathcal{B}$ follows a *perfect shifting pattern* if its values can be obtained by adding a constant-condition number β_j to a typical value for each gene (π_i). β_j is said to be the *shifting coefficient* for condition j. Graphically, a perfect shifting pattern gives a parallel behavior of the genes. In this case, the expression values in the bicluster fulfil the following equation: $b_{ij} = \pi_i + \beta_j$.

Similarly, a bicluster follows a *perfect scaling pattern* changing the additive value in the former equation by a multiplicative one. This new term α_j is called the *scaling coefficient*, and represents a constant value for each condition. In this case, the genes do not follow a parallel tendency. Although the genes present the same behavior with regard to the regulation, changes are more abrupt for some genes than for others. The following equation defines whether a bicluster follows a perfect scaling pattern or not: $b_{ij} = \pi_i \times \alpha_j$.

A bicluster may include some of the aforementioned patterns or even both of them, shifting and scaling, at the same time. This kind of pattern corresponds to the most general situation that can be described using a mathematical formula, when a bicluster exhibits coherent values on both rows an columns, for the additive and multiplicative model at the same time. When it is the case, it is said that the bicluster follows a *perfect shifting and scaling pattern*, and its values can be represented by this equation: $b_{ij} = \pi_i \times \alpha_j + \beta_j$. Nevertheless, to visually identify if some bicluster follows a combined pattern is more difficult that to find a single shifting or scaling pattern, since the effects of one have influence on the other.

2.2. Bicluster taxonomy based on structure

It is also interesting classify the biclustering mathods regarding to the way in which rows and columns from the input matrix are incorporated in biclusters. We named this as *Bucluster Structure*. In this sense, we can define the follows structures:

- **Row exhaustive.** Every gene must belong to at least one bicluster, that is, there are no genes not assigned to at least a bicluster.
- **Column exhaustive.** Every condition must belong to at least one bicluster, that is, there are no conditions not assigned to at least a bicluster.

- **Non exhaustive** the genes and conditions could become not assigned to any bicluster.
- **Row exclusive.** Each gene can to be part of one bicluster at most.
- **Column exclusive.** Each condition can to be part of one bicluster at most.
- **Non exclusive.** This aspect represent the possibility of obtaining overlapped biclusters, that is, several biclusters can share genes and/or condition.

3. Biclustering algorithms based on evaluation measures

As it has already been mentioned, the biclustering problem is NP-hard. This implies that an exhaustive search of the space of solutions may be infeasible. When a measure of quality of the possible solutions is available, the application of a meta-heuristics to solve the problem seems the most appropriate. Meta-heuristics make few or no assumptions about the problem being optimized and can search in very large spaces of candidate solutions by iteratively trying to improve a candidate solution with regard to a given quality measure. However, meta-heuristics do not guarantee that optimal solutions are ever found.

Many different meta-heuristics have been used in the biclustering context, where variants are continually being proposed, in which meta-heuristics are employed together with other kinds of search techniques. In this section we summarize the most important contributions to the biclustering problem when an evaluation measure is available. We have grouped them according to the type of meta-heuristic in which they are based on.

Although we do not get into the details of each evaluation measure, we refer the interested reader to [17]. This work provides a review of the different evaluation functions for biclusters, as well as a comparative study of them based on the type of pattern they are able to evaluate.

The most important biclustering approaches based on evaluation measures are reviewed next. The classification of the techniques has been carried out according to their most relevant characteristic. Note that different categories are not exclusive. Although we have grouped the algorithms attending to what we have considered to be their most distinctive property, some of them may as well be assigned to more than one group.

3.1. Iterative greedy search

Greedy algorithms follow the strategy of making a local optimal choice at each step, in order to find a global optimum. This kind of heuristic does not ensures obtaining an optimal global solution, but approximates it in a reasonable time. They work by either recursively or iteratively constructing a set of objects from the smallest possible constituent parts.

3.1.1. Direct Clustering (DC)

Direct clustering of a data matrix by Hartigan [11] was one of the first works ever published on biclustering, although it was not applied to genetic data. The algorithm is based on the use of a divide and conquer strategy, in which the input matrix is iteratively partitioned into a set of sub-matrices, until k matrices are obtained, where k is an input parameter for the number of desired biclusters. Although being very fast, the main drawback of this heuristic is that partitions cannot be reconsidered once they have been split. This way, some quality biclusters might be missed due to premature divisions of the data matrix.

Within the partitioning process, variance is used as the evaluation measure.

Variance of a bicluster $\mathcal{B}$ is calculated as:

$$VAR(\mathcal{B}) = \sum_{i=1}^{|I|}\sum_{j=1}^{|J|}(b_{ij} - b_{IJ})^2 \quad (1)$$

where b_{ij} and b_{IJ} represent the element in the ith row and jth column, and the mean of B, respectively.

At each iteration, those rows or columns that improve the overall partition variance are chosen. This will lead the algorithm towards constant biclusters. Due to the characteristics of the search, overlapping among biclsters is not allowed.

3.1.2. Biclustering of expression data by Chengu and Church (CC)

Cheng and Church [12] were the first in applying biclustering to gene expression data. Their algorithm adopts a sequential covering strategy in order to return a list of n biclusters from an expression data matrix. In order to assess the quality of a bicluster the algorithm makes use of the *Mean Squared Residue* (MSR). This measure aims at evaluating the coherence of the genes and conditions of a bicluster by using the means of genes and conditions expression values in it.

MSR is defined as:

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

where b_{ij}, b_{iJ}, b_{Ij} and b_{IJ} represent the element in the ith row (condition) and jth column (gene), the row and column means, and the mean of B, respectively.

MSR was the first quality metric defined for biclusters of expression data, and it has been included in many approaches from different authors, although it has been proven to not be the most effective. In fact, MSR is only able to capture shifting tendencies within the data [18].

Fig. 1 shows a scheme of CC. The algorithm takes as input the expression matrix EM and the threshold δ imposed on MSR. δ is used to reject non δ-biclusters. A list L of δ-biclusters is returned as output. After preprocessing the missing values of the input data matrix by replacing them with random numbers, the bicluster discovering process is repeated as many times as biclusters are desired. In each iteration, the bicluster B is initialized to the whole matrix. Next, three different phases for *multiple node deletion*, *single node deletion* and *node addition* are applied. These phases iteratively perform the removal and addition of rows and columns,

```
Input: Expression Matrix $EM$; Thresholds $\delta$
Output: List of Biclusters $L$
1 preprocess the missing values of $EM$
2 list $L = \emptyset$
3 Bicluster $\mathcal{B}$
4 repeat $n$ times
5   $\mathcal{B} = EM$
6   $\mathcal{B}_\delta$ = multiple node deletion phase($B$,$\delta$)
7   $\mathcal{B}'_\delta$ = simple node deletion phase($\mathcal{B}_\delta$,$\delta$)
8   $\mathcal{B}''_\delta$ = addition phase($\mathcal{B}'_\delta$)
9   $L = L \oplus \mathcal{B}''_\delta$
10  substitution phase($\mathcal{B}''_\delta$, $EM$)
11 end_repeat
12 return $L$
```

Fig. 1. Cheng and Church's algorithm.

ensuring that the result is a δ-bicluster. Finally, a *substitution phase* replaces the elements of the input matrix that are contained in the recently found bicluster with random values. This substitution is applied in order to prevent overlapping among biclusters, since it is very unlikely that elements covered by existing biclusters would contribute to any future bicluster. Although this strategy succeeds in avoiding the overlapping, CC presents several drawbacks due to this elements masking and also due to the use of a threshold for rejecting solutions, which is dependent on each dataset and must be computed before applying the algorithm.

3.1.3. SMSR-based biclustering (SMSR-CC)

A similar strategy to that of Cheng and Church has been adopted by Mukhopadhyay et al. [19] in order to incorporate their evaluation measure SMSR (*Scaling MSR*) into a search heuristics. SMSR is defined as follows:

$$SMSR(\mathcal{B}) = \frac{1}{|I| \times |J|} \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} \frac{(b_{iJ} \times b_{Ij} - b_{ij} \times b_{IJ})^2}{b_{iJ}^2 \times b_{Ij}^2} \quad (3)$$

Note that SMSR is an adaptation of MSR able to recognize scaling patterns. Nevertheless, it is not capable to identify shifting behaviors. As it is based on MSR, it also needs to initially set a limit value of SMSR for each dataset. Since SMSR is only able to recognize multiplicative models, the authors propose and adapted algorithm in which CC is applied twice, the fist time using MSR as evaluation measure and the second time using SMSR. Therefore, the strategy allows to independently find both types of patterns (shifting and scaling), but not simultaneously.

3.1.4. HARP algorithm

HARP (*Hierarchical approach with Automatic Relevant dimension selection for Projected clustering*) was presented by Yip et al. [20], and also introduced a new evaluation metric named *relevance index* (RI). RI measures the quality of a bicluster as the sum of the relevance indices of the columns. Relevance index R_{Ij} for column $j \in J$ is defined as:

$$R_{Ij} = 1 - \frac{\sigma_{Ij}^2}{\sigma_j^2} \quad (4)$$

where σ_{Ij}^2 (local variance) and σ_j^2 (global variance) are the variance of the values in column j for the bicluster and the whole data set, respectively.

Iteratively, the biclusters are merged by choosing experimental conditions that satisfy a relevance index threshold. HARP maximizes the quality when biclusters are constant, and its bottom-up merging strategy does not produce overlapped solutions.

3.1.5. Maximum Similarity Bicluster algorithm (MSB)

MSB was proposed by Liu and Wang [21] together with a similarity score for biclusters. The authors highlighted three different characteristics of their approach: (1) no discretization procedure is required, (2) MSB performs well for overlapping biclusters and (3) it works well for additive biclusters. This third property is a direct consequence of the use of the similarity score.

The algorithm starts with the whole matrix as the bicluster. Then a process of iteratively removing the row or column in the bicluster with the worst similarity score is performed, until there is one element left in the bicluster. During this process, $n + m - 1$ sub-matrices have been computed, where n and m refer to the number of rows and columns in the input matrix, respectively. MSB only outputs one bicluster, corresponding to the one in the $n + m - 1$ sub-matrices with the maximum similarity score.

MSB works for the special case of approximately squared biclusters. In order to overcome this issue and also to speed up the process, an extension algorithm named RMSBE (*Randomized MSB Extension algorithm*) is also presented. RMSBE makes use of the average of the similarity scores between some pairs of genes in the bicluster, as well as of randomly selection to choose the reference genes.

3.1.6. Weighted Fuzzy-Based Maximum Similarity Bicluster algorithm (WF-MSB)

Chen et al. [22] extended the MSB algorithm to propose a generalized fuzzy-based approach, named WF-MSB (*Weighted Fuzzy-based Maximum Similarity Biclustering*), for extracting a query-driven bicluster based on the user-defined reference gene.

One of the advantages of this aproach is that WF-MSB discovers both of the most similar bicluster and the most dissimilar bicluster to the reference gene. Also, the expression values of the biclusters extracted by WF-MSB have high difference values compared to the baseline of all expression values, meaning that these biclusters are more significant.

3.1.7. Biclustering by Iteratively Sorting with Weighted Coefficients (BISWC)

In their approach, Teng and Chan [23] alternately sort and transpose the gene expression data, using weighted correlations at the same time to measure gene and condition similarities. The weighted correlation index is a variation of *Pearson's correlation coefficient*, which was originally adapted by Bland and Altman [24] in order to add weights when working with multiple observations. In this work, Teng and Chan have redefined this index so that weights are assigned to the different features (genes or samples) according to their importance. This way, those features with more importance will have more impact than the others.

The algorithm is based on the dominant set approach of Pavan and Pelillo [25]. In order to find a bicluster, genes and conditions are iteratively sorted using weight vectors, which are also iteratively refined using the sorting vector of the previous iteration. In each iteration the matrix is transposed and the process is repeated over the other dimension, thus alternating from genes to conditions. At the end of the process, the highly correlated bicluster is located at one corner of the rearranged gene expression data matrix, and is defined using a threshold for the correlation coefficients between adjacent rows and columns.

To find more than one bicluster, the authors use the weight vectors. This way, any time a bicluster is found, the weights of those features that have not been included in it are enhanced, at the same time as reducing the weights of those features included in it. Using this approach, overlapping among biclusters is permitted but controlled and penalized any time a gene or condition is included in a solution.

3.1.8. Biclustering by Correlated and Large Number of Individual Clustered seeds (BICLIC)

BICLIC [26] has been recently proposed as a biclustering algorithm based on the use of *Pearson correlation coefficient* for biclusters evaluation. Regarding the search strategy, BICLIC is mainly made up of four different phases: finding biclusters seeds, expanding seeds, filtering expansion products and checking and removing duplicated biclusters.

The first phase produces an undetermined number of bicluster seeds by applying individual dimension-based clustering, where genes are labeled and merged according to their expression values in different conditions. These seeds are afterwards expanded in the second phase, where the expansion is performed in two ways: gene-wise and condition-wise, trying to merge each gene or condition iteratively until a certain threshold over the average Pearson's correlation coefficient is exceeded. The products of the second phase are called candidate biclusters, and although not all genes may show correlated patterns over all conditions in a candidate bicluster matrix, it is ensured that all genes and conditions are correlated with the seed bicluster. This way, in the third phase correlation-based biclusters are obtained by iteratively eliminating less correlated sets of genes and conditions. Nevertheless, different seed biclusters may converge to very overlapped correlation-based products. In order to eliminate duplicate already contained biclusters, a last phase is applied in which all biclusters are ordered in an increasing order of their sizes and compared. If every gene and condition in a certain bicluster is included in other bicluster, the smaller one is removed.

Although last phase of BICLIC checks for duplicated biclusters, no overlapping control strategy has been used in order to determine or limit the amount of overlapped elements between biclusters.

3.1.9. Intensive Correlation Search (ICS)

Ahmed et al. [27] have more recently proposed a biclustering technique (ICS) together with a similarity measure for evaluating biclusters based on their shifting-and-scaling correlation (SSSim).

SSSim (*Shifting and Scaling Similarity*) is defined using local means of genes for a baseline (reference) condition pair, combined with and other condition pairs. The resulting score varies from 0 to 1 when evaluation pairs of genes, where a value of 1 indicates that both genes exhibit a perfect shifting-and-scaling correlation. In order to evaluate biclusters, two input parameter called τ and α are used to consider correlations among genes and a partial or complete bicluster, respectively.

ICS (*Intensive Correlation Search*) strategy iteratively extract correlated subspaces for different pairs of genes. These correlated subspaces correspond to maximal set of samples for which gene expressions are correlated by more than the user defined threshold τ. Subspaces correspond to biclusters, which are initially formed by a pair of genes, being also iteratively extended to include more genes, until no more genes can be added to the bicluster. This extension phase is governed by the second user defined threshold, α. Initial gene pairs forming initial correlated subspaces must be formed by genes such that none of these are part of any bicluster that have been generated so far. Nevertheless, this constraint does not prevent genes to be part of several biclusters during the extension phase, allowing thus overlapping among biclusters.

3.2. Stochastic iterative greedy search

Some authors have preferred using a stochastic strategy in order to add a random component to the iterative greedy search, rending thus the algorithm to be non-deterministic. Most important stochastic iterative greedy approaches are reviewed in this section.

3.2.1. Flexible Overlapped biClustering (FLOC)

Yang et al. [28] proposed a different heuristic for coping with the random masking of the values in the data matrix. To address this issue and to further accelerate the biclustering process, the authors presented a new model of bicluster to incorporate null values. They also proposed an algorithm named FLOC (*FLexible Overlapped biClustering*) able to discover a set of k possibly overlapping biclusters simultaneously based on probabilistic moves.

The algorithm begins with the creation of k initial biclusters with rows and columns added to them according to a given probability. After that, these biclusters are iteratively improved by the addition or removal of one row or column at a time, determining the action that better improves the average of the MSR values of the k biclusters. Bicluster volumes are also taken into account within the possible actions, where bigger biclusters are preferred, and the variance is used to reject constant biclusters. The whole process ends when no action that improves the overall quality can be found.

3.2.2. Random Walk Biclustering (RWB)

Angiulli et al. [29] presented a biclustering algorithm based on a greedy technique enriched with a local search strategy to escape poor local minima. Their algorithm makes use of a gain function that combines three different objectives: MSR, gene variance and the size of the bicluster. These objectives are compensated by user-provided weights.

RWB produces one bicluster at a time. Starting with an initial random solution, it searches for a locally optimal solution by successive transformations that improve the gain function. A transformation is done only if there is reduction of MSR or an increase either in the gene variance or the volume. In order to avoid getting trapped into poor local minima, the algorithm executes random moves according to a probability given by the user. To obtain k biclusters RWB is executed k times by controlling the degree of overlapping among the solutions. This degree is controlled for genes and conditions independently by using two different frequency thresholds. This way, during any of the k executions of the algorithm, whenever a gene or condition exceeds the corresponding frequency threshold, it is removed from the matrix and therefore it will not be taken into account in the subsequent executions.

3.2.3. Reactive GRASP Biclustering (RGRASP-B)

GRASP is a multi-start meta-heuristics for combinatorial problems, consisting of iterations made up of two phases: construction of a greedy randomized solution and local search in which its neighborhood is investigated until a local minimum is found. The best overall solution is kept as the result. *Reactive GRASP* is a variant of GRASP in which the threshold parameter α associated to the candidate list of solutions is self-tuned, and its value is periodically modified according to the quality of the solutions recently obtained.

Dharan et al. [30] proposed a reactive GRASP biclustering method in which high quality bicluster seeds are generated using one-dimensional k-means clustering. Afterwards, these seeds are further enlarged by adding more rows and columns to them, employing the reactive GRASP method, and also making use of randomized heuristics for escaping from local optima. The algorithm makes use of MSR score as the cost function to evaluate the quality of the obtained biclusters. In order to obtain k biclusters, k seeds need to be generated in the first step, where their overlapping amount is controlled by the setting of the α values.

3.2.4. Pattern-Driven Neighborhood Search (PDNS)

Neighborhood search consists in iteratively improving an initial candidate solution by replacing it with a higher quality neighbor. It is usually obtained by performing little modifications on the former one.

PDNS has been recently proposed by Ayadi et al. [31] as a *Pattern-Driven Neighborhood Search* approach for the biclustering problem. Prior to the search, the method first applies a preprocessing step to transform the input data matrix into a behavior matrix M''. Each row of this behavior matrix represents the trajectory pattern of a gene across all the combined conditions, while each column represents the trajectory pattern of all the genes under a pair of particular conditions in the data matrix. M'' defines the problem search space as well as the neighborhoods used within the search process.

The initial bicluster is obtained by using two fast well-known greedy algorithms: CC and OPSM, and is encoded into its behavior matrix before being improved by PDNS. The algorithm alternates between two basic components: a descent-based improvement procedure and a perturbation operator. The descendent strategy is used to explore the neighborhood, moving to an improving solution at each iteration. This process is employed to discover locally optimal solutions. In order to displace the search to a new starting points, the perturbation operator is applied. It is carried out after the descent improvement stops according to one of two stopping criteria: the solution reaches a fixed quality threshold or a fixed number of iterations has been reached. A perturbed bicluster is then computed by a random replacement of 10 per cent of genes and conditions of the recorded best bicluster so far. This perturbed bicluster is used as a new starting point for the next round of the descent search, and the whole PDNS algorithm stops when the best bicluster is not updated for a fixed number of perturbations. For assessing the quality of two biclusters any time a replacement is taking place the ASR (*Average Spearman's Rho*)[32], which is based on the use of the *Spearman's rank correlation*.

The algorithm outputs one bicluster at a time. Therefore, in order to obtain several biclusters it must be run several times with different initial solutions. In this work, the authors use the output of two fast well-known algorithm as initial biclusters. Nevertheless, no overlapping control is carried out among the reported solutions.

3.3. Nature-inspired meta-heuristics

Nature-inspired meta-heuristics are characterized by reproducing efficient behaviors observed in the nature. Examples of this kind of heuristics include evolutionary computation, artificial immune systems, ants colony optimization or swarm optimization, among others. All of them make use of algorithmic operators simulating useful aspects of various natural phenomena and have been proven to be very effective for complex optimization problems. In this context, many biclustering approaches have been proposed based on the use of any of this kind of meta-heuristics, being evolutionary computation the most used. In the following, the most important biclustering approaches based on any nature-inspired meta-heuristics have been reviewed.

3.3.1. Simulated Annealing Biclustering (SA-B)

Simulated Annealing (SA) stochastic technique originally developed to model the natural process of crystallization and has been adopted to solve optimization problems [33]. SA algorithm iteratively replaces the current solution by a neighbor one if accepted, according to a probability that depends on both the fitness difference and a global parameter called the temperature. This temperature is gradually decreased during the process, decreasing the probability of randomly choosing the new solution when it gets lower.

The specific behavior of any simulated annealing approach is mainly given by the fitness function an the depth of the search at each temperature. In Bryan et al. [34] approach, the fitness of each solution is given by its MSR value, and ten times the number of genes successes needed to be achieved before cooling. This number determines the depth of the search, being a success an improvement on the fitness function. The initial temperature of the system, as well as the rate at which it is lowered are also important, since both of them determine the number of total iterations and also affect the convergence. The authors set them experimentally.

The algorithm must be run k times in order to obtain k biclusters. Solutions are masked in the original matrix in order to avoid overlap among them. In their work, Bryan et al. used the same method of Cheng and Church [12], replacing the original values with random ones, in an attempt to prevent them to be part of any further bicluster.

3.3.2. Crowding distance based Multi-Objective Particle Swarm Optimization Biclustering (CMOPSOB)

Particle Swarm Optimization (PSO) technique simulates the social behavior of a flock of birds or school of fishes which aim to find food. The system is initialized with a population of random solutions and searches for optima by updating generations. Potential solutions are called particles and fly through the problem space by following the current optimum. Particles have memory for retaining part of their previous state, and decide the next movement influenced by randomly weighted factors affecting the best particle previous position and the best neighborhood previous position. The global optimal solution is the best location obtained so far by any particle in the population and the process ends when each particle reaches its local, neighborhood and global best positions.

Liu et al. [35] based their biclustering approach on the use of a PSO together with crowding distance as the nearest neighbor search strategy, which speeds up the convergence to the Pareto front and also guarantees diversity of solutions. Three different objectives are used in CMOPSOB: the bicluster size, gene variance and MSR, which have been incorporated into a multi-objective environment based on the individuals dominance. Being a populational approach, several potential solutions are taken into account in each generation. Those non-dominated solutions in the last generation will be reported as the output.

3.3.3. Multi-Objective Multi-population artificial immune Network (MOM-aiNet)

Inspired by biological immune systems, *Artificial Immune Systems* (AIS) have emerged as computational paradigms that apply immunological principles to problem solving in a wide range of areas. Coelho et al. [36] presented an immune-inspired algorithm for biclustering based on the concepts of clonal selection and immune network theories adopted in the original aiNet algorithm by Castro and Von Zuben [37]. It is basically constituted by sequences of cloning, mutation, selection and suppression steps.

MOM-aiNet explores a multi-population aspect, by evolving several sub-populations that will be stimulated to explore distinct regions of the search space. After an initialization procedure consisting on random individuals made up of just one row and one column, populations are evolved by cloning and mutating their individuals. The authors apply three different kind of mutations: insert one row, insert one column or remove one element, either row or column. They only consider two different objectives: MSR and the volume, using the dominance among individuals in order to replace individuals with higher quality ones. A suppression step is periodically performed for the removal of individuals with high affinity (overlap), causing thus fluctuations in the populations sizes. Nevertheless, this step is followed by an insertion one, in which new individuals are created, giving preference to those rows or columns that do not belong to any bicluster. After a predefined number of iterations, MOM-aiNet returns all the non-dominated individuals within each sub-population.

3.3.4. Evolutionary algorithms for biclustering

Evolutionary algorithms are based on the theory of evolution and natural selection. Being the oldest of the nature-inspired meta-heuristics, they have been broadly applied to solve problems in many fields of engineering and science. Many biclustering approaches have been proposed based on evolutionary algorithms. Being a populational approach, a larger subset of the whole space of solutions is explored, at the same time that it helps them to avoid becoming trapped at a local optimum. These reasons make evolutionary algorithms very suited to the biclustering problem.

Starting by an initial population, evolutionary algorithms select some individuals and recombine them to generate a new population of individuals. This process is repeated for a number of generations until the algorithm converges or certain criterion is met. A general scheme of an EA is presented in Fig. 2. Given a certain problem that defines a search space (sets of possible biclusters in this context), the EA starts by generating the initial population, that is the initial set of candidate solutions (line 1). These individuals are evaluated (line 2) using problem-dependent metrics which provide a fitness (ϕ) for each candidate solution. Subsequently, offspring is produced by evolving the existing solutions, where fittest solutions often have a higher probability of being selected for reproduction. Thus, the population evolves in the iterative process (lines 3–9), by obtaining the new population based on the current population (P) and the intermediate population (P'') generated by applying the selection function and the genetic operators (crossover and mutarion). Stopping criteria is usually related to a significant improvement on the solutions through generations combined with a maximum number of iterations. Finally, the best solution or set of solutions of the las population are returned.

Next, we summarize the most relevant biclustering approaches based on evolutionary algorithms, both single or multi-objective. Although the majority of them aim at optimizing the popular metric residue (MSR), some of them are also based on correlation coefficients.

Bleuler et al. [38] (Bleuler-B) were the first in developing an evolutionary biclustering algorithm. They proposed the use of binary strings for the individuals representation, and an initialization of random solutions uniformly distributed according to their sizes. Bit mutation and uniform crossover are used as reproduction operators, and a fitness function that prioritises MSR. For those solutions with MSR below the threshold δ, bigger bicluster sizes are preferred. Tournament has been used as selection mechanism, where populations are completely replaced with new offspring. A diversity maintenance strategy is carried out which decreases the amount of overlapping among bicluster, and CC algorithm is also applied as a local search mainly to increase the size of the individuals. At the end, the whole population of individuals is returned as the set of quality biclusters.

SEBI was presented by Divina and Aguilar-Ruiz [14] as a *Sequential Evolutionary BIclustering* approach. The term sequential refers the way in which bicluster are discovered, being only one bicluster obtained per each run of the evolutionary algorithm. In order to obtain several biclusters, a sequential strategy is adopted, invoking the evolutionary process several times. Furthermore, a *matrix of weights* is used for the control of overlapped elements among the different solutions. This weight matrix is initialized with zero values and is updated every time a bicluster is returned. Individuals consists of bit strings, and are initialized randomly but containing just one element. Together with tournament selection, three different crossover and mutation operators are used with equal probability in reproduction: one-point, two-points and uniform crossovers, and mutations that respectively add a row or a column to the bicluster or the standard mutation. Elitism is also applied in order to preserve the best individual through generations. Individual evaluations are carried out by a single fitness function in which four different objectives have been put together: MSR, row variance, bicluster size and an overlapping penalty based on the weight matrix.

```
Input: Search Space
Output: Solution
1  $P$ = initialization()
2  evaluation($P$, $\phi$)
3  while termination condition not met do
4     $P'$ = selection($P$)
5     $P''$ = apply operators($P'$)
6     $P_{next}$ = reproduction($P$, $P''$)
7     $P = P_{next}$
8     evaluation($P$, $\phi$)
9  end_while
10 return best($P$)
```

Fig. 2. General scheme of an EA.

BiHEA (*Biclustering via a Hybrid Evolutionary Algorithm*) was proposed by Gallo et al. [39] and is very similar to the evolutionary biclustering algorithm of Bleuler et al. Both of them perform a local search based on CC algorithm, and return the set of individuals in the last population as the output. However, they differ in the crossover operator (BiHEA uses two-point crossover) and BiHEA also incorporates gene variance in the fitness function. Furthermore, two additional mechanisms are also added in order to improve the quality of the solutions. The first one is elitism, in which a predefined number of best biclusters are directly passed to next generation, with the sole condition that they do not get over a certain amount of overlap. The second one makes use of an external archive to keep the best generated biclusters through the entire evolutionary process, trying to avoid the misplacement of good solutions through generations.

Huang et al. [40] have proposed a new biclustering algorithm based on the use of an EA together with hierarchical clustering. The authors argue that with such a huge search space, the EA itself should not be able to find optimal or approximately optimal solutions within a reasonable time. Therefore, they propose to separate the conditions into a number of conditions subsets, also called *subspaces*. The evolutionary algorithm is then applied to each subspace in parallel, and a expanding and merging phase is finally employed to combine the subspaces results into the output biclusters. As it is related only to the condition dimension, the EA is called CBEB, from *Condition-Based Evolutionary Biclustering*, where the normalized geometric selection method is used as the selection function and the simple crossover and binary mutation methods are employed for reproducing the offspring. In both CBEB and the expanding and merging phase MSR score has been used for the evaluation of the potential solutions, always using the predefined threshold δ as the upper limit.

More recently, Evo-Bexpa (*Evolutionary Biclustering based in Expression Patterns*) [41] has been presented as the first biclustering algorithm in which it is possible to particularize several bicluster features in terms of different objectives. This way, if any previous information related to the microarray under study is available, the search can be guided towards the preferred types of biclusters (number of genes and conditions, overlapping amount or gene variance). These objectives have been put together by using a single Aggregate Objective Function (AOF). Furthermore, other objectives defined by the user can also be easily incorporated into the search, as well as any objective may be ignored. Furthermore, Evo-Bexpa bases the bicluster evaluation in the use of expression patterns, making use of the VE^t metric, able to find shifting and scaling patterns in biclusters, even simultaneously.

3.3.5. Multi-objective evolutionary algorithms

Apart from the specific homogeneity measure for biclusters, many authors have also incorporated other objectives to the search, such as the bicluster volume or gene variance. These requirements are often conflicting. For example, the bigger a bicluster is, the more probable to have a higher MSR value. Nevertheless, bigger biclusters with low MSR values are preferred. The four EA approaches review above make use of a single aggregate objective function which combines the different objectives. In this section we review those EA approaches that optimize the different objectives according to other multi-objectives schemes.

In the work of Mitra and Banka [42] (M&B), the first approach that implements a *Multi-Objective EA* (MOEA) based on Pareto dominance is presented. The authors base their work on the NSGA-II [43], and look for biclusters with maximum size and MSR value, as long as it is smaller than the upper bound δ. Also, a local search strategy based on the *node insertion* and *node deletion* phases of CC algorithm is applied to all of the individuals at the beginning of every generational loop. Populations are ranked according to the dominance criterion, crowding tournament selection is performed, the selected individuals are crossed and mutated, and the best individuals among the new and old populations are selected to remain in the next generation.

In MOGAB (*Multi-Objective GA-based Biclustering algorithm*) [44], the authors propose the use of a new individual representation, encoded as strings made up of two parts: in the first one, indexes of genes acting as cluster centers of sets of genes are represented, while the second one keeps the indexes of the conditions acting as cluster centers of sets of conditions. This way, each individual does not represent a bicluster, but a set of biclusters, obtained with the different possible combinations of clusters of genes and conditions. The initial population contains randomly generated individuals, where each gene or condition is equally probable to become the center for a gene or a condition cluster, respectively. MOGAB is also based on the NSGA-II strategy, and performs crowded binary tournament selection, single-point crossover and standard mutation, although these two last operators are carried out on gene and condition centers strings independently, and invalid individuals are marked when appear in order not to let them reproduce in the next generation. Elitism has also been incorporated in MOGAB to track the non-dominated Pareto-optimal solutions. Within the evaluation, MSR and row variances are computed for all the $\delta-$biclusters denoted by each individual. A fitness vector is afterwards created with the mean of their fitness. From the non-dominated individuals in the final population, all the $\delta-$biclusters are extracted and output as the final biclusters.

Since the boundaries of biclusters usually overlap, some authors believe the notion of fuzzy sets is useful for discovering such overlapping biclusters. In this regard, a fuzzy version of MOGAB, named MOFB (*Multi-Objective Fuzzy Biclustering algorithm*) is proposed in [45]. MOFB simultaneously optimizes fuzzy versions of MSR, row variance and volume and applies a interesting variable string length encoding scheme.

SMOB (*Sequential Multi-Objective Biclustering*) has been used together with different measures, such as MSR and VE (*Virtual Error* [46]) by Divina et al., adopting a sequential strategy similar to SEBI [14]. Unlike SEBI, where a single-objective EA was used, SMOB invokes a multi-objective evolutionary algorithm (MOEA) several times. Each time the MOEA is called, a bicluster is returned and stored in a list. The number of elements in the output list represents the number of found solutions. In these works, biclusters with high volume, good quality (being quality measured by an appropriate metric such as VE, MSA or MSR) and relatively high gene variance are addressed. Thus, three objectives have been individuated to be optimized, being in conflict with each other.

3.4. Clustering-based approaches

This section describes those biclustering algorithms which base their search on the use of a traditional one-dimension clustering algorithm, together with an additional strategy providing the second dimention analysis.

3.4.1. SVD and clustering-based approaches

- **Possibilistic Spectral Biclustering (PSB).** A biclustering approach based on the use of *Singular Value Decomposition* (SVD) together with one-dimensional clustering named PSB was proposed by Cano et al. [47]. The use of SVD techniques enhances the clustering process by performing dimensionality reduction.
  This algorithm consists of firstly applying SVD method on an eigenproblem formulated on the input matrix and get $\min(n, m)$ solution eigenvectors, where n and m refers to the number of genes and conditions in the input matrix. Using these eigenvectors several partition matrices are created. Two independent clustering algorithms are then executed on them: for those rows representing genes and for those representing conditions in the original matrix, respectively. Each combination of a cluster of genes and a cluster of conditions is a candidate bicluster, which will be post-processed in order to improve its quality when possible, or rejected if it is not considered a quality solution. Finally, the whole process is repeated with a linear inversion of the input expression matrix, in order to also obtain under-expressed genes.
  The clustering algorithm used in PSB is a variation of the *Improved Possibilistic Clustering* (IPC) by Zhang and Leung [48], which mixes possibilistic and probabilistic approaches. MSR is used at both the clustering and the crisping of the possibilistic biclusters, though in this last step the volume is also taken into account. Possibilistic clustering allows a considerable amount of overlapping, so the authors have also added to the process an overlapping control, in which a bicluster is checked for its overlapping amount before being added to the result set.
- **Biclustering with SVD and Hierarchical Clustering (SVD&HC-B).** Yang et al. [49] have recently proposed a strategy similar to the one of Cano et al. [47], by using *Singular Value Decomposition* (SVD) together with clustering and a final stage to merge and filter the clusters. In this approach, the authors make use of the *sub-matrix correlation score* also presented in their work. A upper bound δ is used to defined a $\delta-$corbicluster as a bicluster with a sub-matrix correlation score lower than δ.
  In a first step, two different matrices named $R^{(l)}$ (a group of basis genes) and $C^{(l)}$ (a group of basis conditions) are obtained by using SVD. Secondly, after centralizing the rows of these two matrices, clustering is applied to both of them by the *Mixed Clustering algorithm*, based on agglomerative hierarchical clustering and on the use of the sub-matrix correlation score as dissimilarity measure. The number of clusters produced by this technique is not known beforehand. This way, a set of m and n groups of clusters are obtained from both matrices. Every pair of these groups constructs a bicluster, obtaining $m \times n$ biclusters in this way. Nevertheless, not every pair of groups may be a $\delta-$corbicluster, and a final step is executed in order to obtain inclusion-maximal biclusters. This last step is carried out by the *Lift algorithm*, inspired in the *node-deletion* and *node-addition* phases proposed by Cheng and Church, but according to the sub-matrix correlation score. Since the clustering algorithm generates not mutually exclusive clusters, the biclusters obtained by this method are possible overlapped.

3.4.2. Biclustering based on Related Genes and Conditions Extraction (RGCE-B)

Yan et al. [50] have recently proposed a search strategy for obtaining biclusters with related genes and conditions of a given bicluster type. They apply hierarchical clustering to different data matrices generated from the input one, according to genes/conditions stability and different bicluster types. Additionally, as a preprocessing step, missing data is estimated with the James–Stein method and Kernel estimation principle, where k means is used to obtain the estimation matrix.

Stable and unstable genes/conditions are first selected base on the cosine of the angle between the vector $x = (1, 1, \ldots, 1)$ and each row/column of the data matrix. This way, and using a predefined threshold, those vectors more similar to x are considered to be stable, and unstable otherwise. Using this criterion, the gene expression matrix can be partitioned to submatrices of stable and unstable ones in both the row and column directions.

On the other hand, five different types of biclusters previously defined in the literature have been used (see Section 2.1): C (constant values), CR (constant values on rows), CC (constant values on conditions), ACV (additive coherent values) and MCV (multiplicative coherent values). Using these definitions it is possible to state that C and CR types exist in row-stable matrices, C and CC types exist in column-stable matrices, while ACV and MCV types exist in unstable matrices.

The process therefore consist in the creation of a sparse matrix with the same size as that of the original gene expression matrix for each type of biclusters, where the zero and non-zero entries correspond to the irrelative and related genes and conditions, respectively. After reducing the dimension of these matrices, biclusters of any type can be obtained based on the corresponding sparse matrix. In this approach, a hierarchical agglomerative single linkage method is used to perform clustering in both the row and column directions, for each sparse matrix. Biclusters are afterwards obtained by finding the overlapping parts between clusters in different directions. As a final post-processsing step, biculters are refined by deleting rows and columns according to a MSR threshold.

4. Non metric-based biclustering

In this section we review the most important biclustering approaches that exclude the use of any evaluation measure for guiding the search. We have classified them according to their most relevant aspects: specific algorithm, data structure representation or the main important basis. Likewise that in Section 3, it is important to note that different categories are not exclusive, that is, the algorithms have been grouped attending to what we have considered to be their most distinctive property, althought some of them may as well be assigned to more than one group.

4.1. Graph-based approaches

This section reviews four different biclustering approaches based on the use of graph theory. Some of them use nodes for bicluster elements representation, either genes, samples or both genes and samples, while some other make use of nodes for representing whole biclusters.

4.1.1. SAMBA

Tanay et al. [13] based their approach on graph theoretic coupled with statistical modeling of the data, where SAMBA stands for *Statistical-Algorithmic Method for Bicluster Analysis*. In their work, they model the input expression data as a bipartite graph whose two parts correspond to conditions and genes, respectively, and edges refer to significant expression changes. The vertex pairs in the graph are assigned weights according to a probabilistic model, so that heavy sub-graphs correspond to biclusters with high likelihood. Furthermore, they present two statistical models of the resulting graph, the second one being a refined version of the first in order to include the direction of expression change, allowing thus to detect either up or down regulation. Weights are assigned to the vertex pairs according to each model so that heavy sub-graphs correspond to significant biclusters. This way, discovering the most significant biclusters means finding the heaviest sub-graphs in the bipartite graph model, where the weight of a sub-graph is the sum of the weights of the gene-condition pairs in it. In order to cope with noisy data, Tanay et al. searched for sub-graphs that are not necessarily complete, assigning negative weights to non-edges.

In order to reduce the complexity of the problem, high-degree genes are filtered out, depending on a pre-defined threshold. According to the authors, the number of genes is reduced by around 20 per cent, considering it to be a modest reduction. The proposed algorithm is an iterative polynomial approach based on the procedure for solving the maximum bounded bi-clique problem, where a hash-table is used for the identification the heaviest bi-clique. A generalization of this method is applied in order to give the k heaviest non-redundant sub-graphs, where k is an input parameter. Before applying the algorithm, the graph structure may be created, using the signed or unsigned model depending on the input data.

4.1.2. MicroCluster

MicroCluster was developed by Zhao and Zaki [51] as a biclustering method for mining maximal biclusters satisfying certain homogeneity criteria, with possible overlapped regions. Biclusters with shifting patterns are detected by using exponential transformations. Furthermore, by means of these kind of transformations, scaling patterns may also be detected.

MicroCluster uses an enumeration method consisting of three steps. Firstly, a weighted, directed range multi-graph is created for representing the possible valid ratio ranges among experimental conditions and the genes that meet those ranges. A valid ratio range is an interval of ratio values satisfying several constraints on expression values. In this graph, vertices correspond to samples and each edge has an associated gene set corresponding to the range on that edge. The construction of this range multi-graph filters out most unrelated data. Once the multi-graph is created, a second step is applied for mining the maximal clusters from it, based on a recursive depth-first search. Although the output of this step is the final set of biclusters, a final step is optionally executed in order to delete or merge those biclusters according to several overlap conditions. This last step is also applied to deal with noise in data, controlling the noise tolerance.

4.1.3. QUBIC

QUBIC has been presented as a *QUalitative BIClustering algorithm* [52], in which the input data matrix is first represented as a matrix of integer values, either in a qualitative or semi-qualitative manner. In this representation, two genes are considered to be correlated under a subset of conditions if the corresponding integer values along the two corresponding rows of the matrix are identical. The qualitative (or semi-qualitative) representation is such that allows the algorithm to detect different kind of biclusters, also including scaling patterns. It is also suitable for finding both positively and negatively correlated expression patterns, where negative correlations will be represented by opposite signs across the entire row. The biclustering problem consist now in finding all the optimal correlated sub-matrices.

The first step of the algorithm correspond to the construction of a weighted graph from the qualitative or semi-qualitative matrix, with genes represented as vertices, and edges connecting every pair of genes. Edge weights are computed in the base of the similarity level between the two corresponding rows. After the graph has been created, biclusters are identified one by one, starting for each bicluster with the heaviest unused edge as a seed. This seed is used to build an initial bicluster and the algorithm iteratively adds additional genes into the current solution. The consistency level marks the end of the search for a bicluster, since it determines the minimum ratio between the number of identical non-zero integers in a column and the total number of rows in the sub-matrix.

4.1.4. CoBi

CoBi (*Pattern-based Co-Regulated Biclustering of Gene Expression Data*) [53] makes use of a tree to group, expand and merge genes according to their expression patterns. In order to group genes in the tree, a pattern similarity between two genes is defined given their degrees of fluctuation and regulation patterns. While the first is used to represent the expression levels variations among conditions,the regulation pattern represents the up, down and no regulation. this way, CoBi is able to find biclusters exhibiting both positive and negative regulation among genes.

The process consist of generating biclusters by means of a tree with a single pass of the dataset. After preprocessing the input data, computing both the degrees of fluctuation and the regulation patterns, CoBi starts by creating an initial BiClust tree. This tree will contain $M - 1$ initial nodes, where M is the number of experimental conditions in the datasets. The first step corresponds to the creating of leaf nodes for each branch of the tree, by forming clusters of genes based on the previously defined similarity. To maintain a moderate number of gene clusters a pruning step is performed, where a cluster is removed if its size is below a certain threshold. Next, in the second phase, the tree is iteratively expanded and merged to produce higher order biclusters. The merging is carried out in two different ways: at a non-leaf level and at a cluster level. Since this merging phase might produce clusters already contained in the same branch, they will be identified and removed from the tree. Once the tree reaches the end of expansion so that no further merging is possible, CoBi returns the list of all non-redundant obtained biclusters.

4.2. One-way clustering-based approaches

Similar to Section 3.4 regarding non metric-based biclustering approaches, this section describes those biclustering algorithms which base their search on the use of a traditional one-dimension clustering algorithm, together with an additional strategy providing the second dimention analysis. In this case, the algorithms reviews do not make use of any bicluster evaluation measures within the search.

4.2.1. Coupled Two-way Clustering (CTWC)

Coupled Two-Way clustering (CTWC), introduced by Getz et al. [54] defines a generic scheme for transforming a one-dimensional clustering algorithm into a biclustering algorithm. They define a bicluster as a pair of subsets of genes and conditions, or a pair of gene and conditions clusters. They also define a stable cluster as a cluster that is statistically significant according to some criterion (such as stability, critical size, or the criterion used by Hartigan [55]). Getz et al. also applied a normalization step based on euclidean distance as a previous step to the application of the algorithm.

The heuristics provided by CTWC consists in an iterative process restricting the possible candidates for the subsets of genes and samples, only considering and testing those genes and samples clusters previously identified as stable clusters. The iterative process is initialized with the full matrix. Both sets of genes and samples are used to perform two-way clustering, storing the resulting stable clusters in one of two registers of stable clusters (for genes or samples). Pointers that identify parent clusters are also stored, consisting thus in a hierarchical approach. These steps are iterated further, using pairs of all previously found clusters, and making sure that every pair is treated only once. The process is finished when no new clusters that satisfy the criteria are found.

The success of this strategy depends on the performance of the one-dimensional clustering algorithm. According to the authors, CTWC can be performed with any clustering algorithm. Nevertheless, many popular clustering algorithms (e.g. K-means, Hierarchical, SOM) cannot be used in this approach, since they do not readily distinguish significant clusters from non-significant clusters or make a priori assumption on the number of clusters [13]. Getz et al. [54] recommend the use of the SPC (superparamagnetic clustering algorithm) [56], which is especially suitable for gene microarray data analysis due to its robustness against noise and its ability to identify stable clusters.

4.2.2. Interrelated Two-way Clustering (ITWC)

Interrelated Two-Way Clustering (ITWC) developed by Tang and Zhang [57] is an algorithm similar to CTWC, combining the results of one-way clustering on both dimensions separately. A pre-processing step based on row normalization is also applied, where rows with little variation are removed from the input matrix. Although correlation coefficient is used as similarity measure to measure the strength of the linear relationship between two rows or two columns in the process, Tang and Zhang do not propose any quality metric for the evaluation of a sub-matrix as a whole. For this reason we have categorized this approach as a non-metric based.

The idea behind ITWC is to discover the relationships between gene and sample clusters while iteratively clustering through both dimensions to extract important genes and classify samples simultaneously. Within each iteration there are five main steps. First step performs clustering on rows, while in the second step clustering is performed in the column dimension, for each group of genes from step one. In this second step, only two clusters are obtained for each gene group. Third step combines the former steps results by computing diverse sets intersections for the sample groups, resulting in 2^k sample groups, where k is the number of gene clusters obtained in step one. Fourth step aims at finding heterogeneous groups of conditions (which do not share rows used for clustering), being the result of this step a set of highly disjoint biclusters. Last step of ITWC sorts the rows in descending order of the cosine distance between each row and a row representative of each bicluster. After that, only the first one third of rows is kept, reducing thus the row set for each heterogeneous group. A likelihood based on the correlation coefficient is calculated for each heterogeneous group, being the reduced genes of the group with the higher likelihood value the selected for the next iteration. These genes and the entire samples then form a new gene expression matrix from which a new iteration starts. Iterations will be terminated when a stable and significant pattern of samples has emerged. For this purpose, the authors use a criterion based on a coefficient of variation to measure how internally-similar and well-separated the partition is.

Biclusters identified by ITWC have no elements in common, due to the search strategy. This way, overlapping among biclusters is not allowed in this approach.

4.3. Probabilistic models

A probabilistic model is a model created using statistical analysis to describe data based on the use of probability theory. In this section, we review those algorithms which look for biclusters in a dataset making use of different types of probabilistic models.

4.3.1. Plaid Models (PM)

Lazzeroni and Owen [58] proposed *plaid models*, a tool for exploratory analysis of multivariate data. In this approach, the genes-condition matrix is represented as a superposition of layers, corresponding to biclusters. Several versions of the model are described in their work, being the most general the one in Eq. (5), which allows a gene to be in more than one bicluster or in none at all.

$$Y_{ij} \doteq \sum_{k=0}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk} \tag{5}$$

where Y_{ij} refers to the expression level of gene i under sample j in the input matrix, K is the number of biclusters, θ_{ij0} describes the background layer and θ_{ijk} represents four different types of models, depending on the types of biclusters (overlapped, exclusive ...). Each $\rho_{ik}\epsilon\{0,1\}$ is 1 if gene i is in the k'th bicluster, zero otherwise. Similarly, each $\kappa_{jk}\epsilon\{0,1\}$ is 1 if sample j is in the k'th bicluster, zero otherwise. Using this equation, a bicluster is assumed to be the sum of a bicluster background level plus row-specific and column-specific constants.

In order to find k biclusters in the data, Lazzeroni and Owen proposed a greedy algorithm that adds one layer at a time. The process seeks for a plaid model minimizing the sum of squared errors when approximating the data matrix to the model. For this purpose, an iterative approach is adopted with each cycle updating θ values, ρ values and κ values in turns. Assuming that residual data becomes more and more unstructured noise as each layer is removed from the data, authors propose a simple rule for stopping the process, in which only a small number of extra layers can be extracted once the data have been reduced to noise.

4.3.2. Rich Probabilistic Models (RPM)

Rich probabilistic models was proposed by Segal et al. [59] for studying relations between expression, regulatory motifs and gene annotations. The main advantage of this approach is the use of additional types of information in the analysis, such us functional roles or cellular location for genes. In the case of samples, the information would depend on the type of arrays used. It might be the treatment applied to the sample or growth conditions among others.

Starting with the input array and the available additional information, a predictive model is learnt from the data. The approach is based on the language of *Probabilistic Relational Models* (PRMs) that extend Bayesian networks to a relational setting, also allowing the inclusion of multiple types of information. Furthermore, the *Expectation Maximization* (EM) algorithm is used for parameter estimation with incomplete data.

The outcome of the algorithm can be interpreted as a collection of disjoint biclusters generated in a supervised manner, where overlapped is not allowed.

4.3.3. Gibbs Sampling (GS)

A biclustering algorithm based on the use of *Gibbs sampling* was proposed by Sheng et al. [60], where a simple frequency model is adopted for representing the expression pattern of a bicluster. According to the authors, using Gibbs sampling as the method for parameter estimation avoids the problem of local minima that often characterizes expectation maximization. In order to achieve a better performance of Gibbs sampling, microrray data is first discretized, resembling thus the problem of finding sub-sequences sharing similar alphabetic expressions.

Background model has been introduced as a single multinomial distribution. This is suitable for data sets where the genes share the same distribution under every condition. If this is not the case, several multinomial distributions might be used, each of which describing the background under an individual condition.

The probabilistic model adopted considers only the presence of a single bicluster in the data set. In order to discover more than one bicluster an iterative process is carried out, masking the genes selected for the recently found bicluster and running the algorithm on the rest of the data. This masking strategy prevents genes from appearing in several biclustering, avoiding thus the possibility of overlap.

4.3.4. Bayesian Biclustering model (BBC)

Gu and Liu [61] developed a *Bayesian Biclustering Model* (BBC), also based on the use of a Gibbs sampling procedure to make the Bayesian inference of biclusters. For a single bicluster, the same model as in the plaid model [58] is assumed. Nevertheless, for multiple biclusters, the amount of overlap among the biclusters in the original plaid model was too high. In order to overcome this situation, in the BBC model overlap is only allowed in one direction, either gene or condition. This means that if overlapped among genes is permitted, then any experimental condition would only appear in at most one bicluster, and vice versa. Also, biclustering is not exhaustive in this approach, since any element (gene or condition) can belong to either one or none of the found biclusters.

BBC works on normalized microarray data, although this step is not included in the algorithm. The authors conducted a study on how different normalization methods affect the performance of their algorithm, showing that the model is stable for two normalization methods developed by themselves, the *Interquartile Range Normalization* (IQRN) and the *Smallest Quartile Range Normalization* (SQRN), being both of them inspired in column standardization.

According to the authors, missing data can be easily handled by just treating them as additional unknown variables. Furthermore, other types of information might be incorporated into the model in order to improve the search.

4.3.5. Conserved gene expression Motifs (xMOTIFs)

Murali and Kasif [62] proposed the use of xMOTIFs (*conserved gene expression Motifs*) for the representation of gene expression data, where a xMOTIF is a subset of genes that is simultaneously conserved across a subset of samples, and a gene expression level is conserved across a set of samples if it is in the same state in each of the samples in the subset. Therefore, for each gene, a list of intervals representing the states in which the gene is expressed in the samples is required. In order to prevent the algorithm from finding too small or too large xMOTIFs, some constraints on their size, conservation and maximality have been added to its formal definition.

A probabilistic algorithm that exploits the mathematical structure of xMOTIFs to compute the largest xMOTIF was also developed by Murali and Kasif. In order to identify several xMOTIFs in the data, an iterative strategy has been adopted, where samples satisfying each xMOTIF are removed from the data, and the new largest xMOTIF is searched. This process continues until all samples satisfy some xMOTIF. This search strategy allows gene overlap and also sample overlap, whenever any sample does not take part in more than one xMOFIT with the same gene.

4.3.6. cMonkey

cMonkey [63] is a biclustering algorithm specifically designed for genetic data, which integrates sequence data, gene expression data and gene network association data. cMonkey is model-based, where distribution variables are parametrized using simple statistical distributions, being more robust to varying data size and quality. Each of the individual data types are modeled, using logistic regression to integrate them into a joint model.

Biclusters are modeled using a Markov chain process, in which an initial bicluster seed is iteratively optimized by updating its state based upon conditional probability distributions, also applying a simulated annealing procedure. Biclusters are optimized sequentially, starting with a new bicluster seed a predefined number of times. Seeds are initialized only with genes that have not been previously placed into any former bicluster, although in the subsequent iterations those genes can still be added to new biclusters. This way overlapping among biclusters is allowed but controlled, since a filtered set of biclusters is finally computed in order to reduce redundancy.

4.3.7. Penalized Plaid Model (PPM)

In their work, Chekouo and Murua [64] main concern is to shed light on associated statistical models behind biclustering algorithms. According to their findings, many of known techniques have an underlying hidden Bayesian touch. This characteristic motivated them to adopt a Bayesian framework to model biclustering, incorporating two main characteristics to previous works: a criterion to choose the number of biclusters and a penalized plain model for addressing overlapping among biclusters. In this model, hard-EM has been preferred against EM (*Expectation Maximization*) for parameter estimation, since the number of parameters increase exponentially with the number of biclusters K in the bicluster model.

Regarding overlapping, the word *penalized* has been incorporated to the algorithm in order to indicate this strategy, where the amount of overlapping can be controlled by imposing a prior restricting it. This way, a parameter λ ($\lambda \geqslant 0$) has been added to control the amount of bicluster overlapping, where the model becomes a non-overlapping one for very large values of λ. When λ cannot be fixed a priori to a suitable value, its value will be generated in the MCMC (*Markov chain Monte Carlo*) sampling, by using a Metropolis–Hastings step.

As for the estimation of the number of biclusters, the authors propose two modified versions of the deviance information criterion (DIC), marginal DIC (DIC_m) and conditional DIC (DIC_c). Although the first one has proven to work very well in the experiments, it is more computationally expensive. In this context, DIC_c has been presented as an alternative to DIC_m with a much faster computation, using a maximum a posteriori (MAP) estimator.

4.4. Linear algebra

This section describes the most relevant biclustering approaches based on the use of linear algebra. This way, they make use of vector spaces and linear mappings between such spaces for describing and finding the most correlated submatrices from the input data set.

4.4.1. Spectral Biclustering (SB)

Spectral biclustering was especially designed by Kluger et al. [65] for analyzing microarray cancer datasets. In this context, it is assumed that the expression matrix has a hidden checkerboard-like structure with blocks of high-expression levels and low-expression levels. Kluger et al. approach consist in finding these distinctive checkerboard patterns, by using eigenvectors and commonly used linear algebra approaches, such as the *Singular Value Decomposition* (SVD). Furthermore, normalization steps have also been integrated into the search, in order to put the genes on the same scale so that they have the same average level of expression across conditions, and likewise for the conditions.

Biclusters found by spectral biclustering have no elements in common, forbidding thus any kind of overlap among them. Moreover, this is an exhaustive approach, meaning that every gene and every condition will be included in one bicluster.

4.4.2. Iterative Signature Algorithm (ISA)

Iterative Signature Algorithm (ISA) was proposed by Bergmann et al. [66] and provides a definition of biclusters as *transcription modules* to be retrieved from the expression data. A transcription module consist of a set of co-regulated genes and the set of experimental conditions under which this co-regulation is the most stringent. Its size depends on the associated set of two thresholds that determine the similarity between the genes and conditions of the module, respectively.

In order to find transcription modules in the data, the signature algorithm consisting in a generalization of *Singular Value Decomposition* (SVD) is applied. The algorithm starts with a set of randomly selected genes or conditions, iteratively refining the genes and conditions until they match the definition of a transcription module. The method includes data normalization and the use of thresholds that determine the resolutions of the different transcription modules. In this search one bicluster is produced at each iteration. Initial seeds are randomly chosen without any overlap restriction, therefore, different biclusters may contain overlapped genes and/or conditions.

4.4.3. Non-smooth Non-negative Matrix Factorization (nsNMF)

A method based on the application of the *non-smooth non-Negative Matrix Factorization* (nsNMF) technique for discovering local structures from gene expression datasets was developed by Carmona-Saez et al. [67]. This approach consists in a variant of the classical NMF model, which adds non-smoothness constraints in order to produce more compact and localized feature representation of the data.

As well as NMF, nsNMF approximates the original matrix as a product of two sub-matrices, where the columns of the first one are basis experiments (also called factors) and the rows of the second constitute basis genes (or encoding vectors). Each factor determines a local gene expression feature or gene module, while each encoding vectors determine the set of experimental conditions highly associated to these modules. This way, biclusters are determined by the corresponding pairs of columns and rows of both matrices. Therefore, overlap among biclusters is not allowed in this approach.

4.4.4. Pattern-based Biclustering (BicPAM)

Henriques and Madeira [68] have recently proposed a biclustering strategy which integrates state-of-the-art pattern-based approaches together with other novelty solutions distributed through three different steps in their algorithm.

BicPAM applies and ordered composition of three phases: *mapping*, *mining* (or pattern discovery) and *closing* (or post-processing). All these steps rely on existing principles in pattern mining, enhancing them by the utilization of other innovative principles.

Mapping step consist in the itemization of a real-value matrix into an itemset matrix. This phase comprises normalization and discretization steps, including both of them several optional procedures. It also introduces and additional handling of missing values and tackling with varying levels of noise.

Mining step corresponds to the core step, where the target pattern miners are applied, and is driven by the considered pattern discovery approach. This phase depends on three points according to the adopted pattern-based approach to biclustering, the target pattern representation and the search strategy. Regarding the first point, BicPAM uses frequent itemsets and association rules to compose biclustering solutions. With reference to the second characteristic, BicPAM uses frequent closed patterns as the default representation, allowing thus overlapping among biclusters only if a reduction on the number of columns in a bicluster results in a higher number of rows. Finally, concerning the third point, BicPAM makes available several algorithmic choices to compose biclusters, including a variant of FP-Growth [69] as the default option. The novelty in this mining step resides in the incorporation of solutions for the discovery of biclusters allowing symmetries (negative correlations) and additive and multiplicative patterns.

Last stage (closing step) consist in the application of several post-processing tasks to affect the structure and quality of the target biclusters. Specifically, BicPAM enables the use of criteria structured according to the extension, merging and filtering of the results, according to several related thresholds, such as percentage of noise, overlapping degree or statistical significance levels. This closing step is also in charge of composing biclusters with flexible structures.

4.5. Optimal reordering of rows and columns

This last category include those biclustering strategies based on performing permutations of the original rows and columns in the data matrix, leading to a better arrangement of the elements previously to the search.

4.5.1. OPSM

Ben-Dor et al. [70] define biclusters to be order-preserving sub-matrices (OPSMs), in which the expression levels of all genes induce the same linear ordering of the experiments. This way, a sub-matrix is said to be order-preserving if there is a permutation of its columns under which the sequence of values in every row is strictly increasing. This strict condition might be relaxed for real expression data, where rows having a significant tendency to be similarly ordered are searched for instead. This relaxation introduces a new probability which will influence the generation of the probabilistic model used.

Guided by the probabilistic model, an efficient algorithm is also proposed for finding the hidden OPSM in the data. It consists of an iterative greedy heuristic algorithm based on the concepts of partial and complete models. Since finding the best model would be infeasible, their approach consists of growing *partial models* iteratively, and trying to converge to the best complete model. At the beginning, all partial models of first order are generated, picking afterwards the best ones and computing for each of them the possible extensions to partial models of the next order. This process is successively repeated until the models of the top order are reached, returning the best of them as the complete model. The algorithm can also be used to discover more than one OPSM in the same dataset, even when they are overlapped.

4.5.2. OREO

OREO was proposed by DiMaggio et al. [71] as an approach based on the *Optimal RE-Ordering* of the rows and columns of a data matrix so as to globally minimize a dissimilarity metric. Contrary to OPSM, this approach allow for monotonicity violations in the reordering, but penalize their contributions according to a selected objective function.

Two rows or columns are defined to be adjacent if the second one is directly below the first in the final arrangement. Using this definition, the final ordering may be represented by a matrix of binary 0–1 variables for each dimension. The optimal rearrangement is obtained using a metric of similarity together with one of the two proposed problem formulations. Although the objective function might be defined by the user, the authors propose three different choices: the relative difference, the squared difference or a metric similar to the root-mean squared deviation. All of them only applied for the values of adjacent elements.

Table 1
Biclustering algorithms based on evaluation measures.

Algorithm		Acronym	Ref.
Iterative greedy search	Direct Clustering	DC	[11]
	Cheng and Church	CC	[12]
	SMSR-based Biclustering	SMSR-CC	[19]
	HARP Algorithm	HARP	[20]
	Maximum Similarity Bicluster Algorithm	MSB	[21]
	Weighted Fuzzy-Based Maximum Similarity Biclustering Algorithm	WF-MSB	[22]
	Biclustering by Iteratively Sorting with Weighted Coefficients	BISWC	[23]
	Bic. by Correlated and Large number of Individual Clustered seeds	BICLIC	[26]
	Intensive Correlation Search	ICS	[27]
Stochastic iterative greedy search	FLexible Overlapped biClustering	FLOC	[28]
	Random Walk Biclustering	RWB	[29]
	Reactive GRASP Biclustering	RGRASP-B	[30]
	Pattern-Driven Neighborhood Search	PDNS	[31]
Nature-inspired meta-heuristics	Simulated Annealing Biclustering	SA-B	[34]
	Crowding distance based Multi-objective PSO Biclustering	CMOPSOB	[35]
	Multi-objective Multi-population Artificial Immune Network	MOM-aiNet	[36]
	Evolutionary Algorithms for Biclustering		
	Bleuler Alg.	Bleuler-B	[38]
	SEBI	SEBI	[14]
	BiHEA	BiHEA	[39]
	CBEB	CBEB	[40]
	EvoBexpa	EvoBexpa	[41]
	Multi-objective Evolutionary Algorithms for Biclustering		
	Mitra & Banka Alg.	M&B	[42]
	MOGAB	MOGAB	[44]
	Multiobjective Fuzzy Biclustering	MOFB	[45]
	SMOB	SMOB	[46]
Clustering-based approaches	Biclustering based on related genes and conditions extraction	RGCE-B	[50]
	SVD and Clustering		
	Possibilistic Spectral Biclustering	PSB	[47]
	Biclustering with SVD and Hierarchical Clustering	SVD&HC-B	[49]

The selected objective function would guide either a network flow model or a *Travelling Salesman Problem* (TSP) approach in order to obtain the optimal rearrangement of rows and columns. In both models, two different elements (rows or columns) are connected if they are adjacent, although in the case of TSP they are weighted edges, where the weight is computed using the objective function. Furthermore, both approaches require the use of additional constraints to avoid cyclic arrangements.

The algorithm begins by optimally re-ordering a single dimension of the data matrix. After that, the median is computed for each pair of adjacent elements (either rows or columns), where the top 10 percent of largest median values define the cluster boundaries between the re-ordered elements. These cluster boundaries are then used to partition the original matrix into several sub-matrices. Finally, the other dimension of each sub-matrix is re-ordered and clusters in this dimension are again defined using the median value of the objective function between neighboring elements in the final ordering.

5. Conclusions and discussion

This paper is the result of a wide study of different existing approaches for biclustering gene expression data. As the main contribution, we have provided a review of a large number of biclustering approaches, classifying them into two categories according to whether or not use evaluation metrics within the search method: biclustering algorithms based on evaluation measures, summarized in Table 1, together with the used metrics and the corresponding references; and non metric-based biclustering algorithms, summarized in Table 2, with their corresponding references. In both cases, they have been classified according to the type of meta-heuristics which they are based on. The classification of the forty-seven biclustering approaches presented in this work as well as their systematic organization constitutes a good starting point to new researchers interested in the field.

According to Tables 1 and 2, it can be derived that current research on biclustering is being focused more on algorithms based on evaluation measures than on non metric-based ones. This tendency might be due to the fact that guiding the heuristics through a quality measure allows the algorithm be more versatile. This way, the search strategy would not be affected by changes in the bicluster evaluation. Another interesting conclusion that can be pointed out is the importance of nature-inspired techniques for biclustering, since it constitutes the most explored field within stochastic strategies.

Table 3 summarizes the main characteristics of the methods under review. In this table, we have decided to use the same features information as in [6], although with a different structure. This way, for each algorithm it is specified the pattern type of biclusters found, the biclusters structure and the discovery strategy. We have therefore extended and updated the table in the survey [6], including the same information for the 47 reviewed algorithms in this work.

Pattern type columns specify the type of biclusters pattern found by each algorithm, as defined in Section 2.1. This way, a bicluster method might find constant patterns (either on row, columns or overall), coherent values (either on row, columns or simultaneous), or coherent evolutions. Although we have specified the type of coherent values (either additive, multiplicative or both), those algorithms obtaining only one type of pattern may be easily adapted to also obtain the other (although no simultaneously). In this column, we have also marked those algorithms able to incorporate negative correlations in the output biclusters. Biclusters structure columns exhibit the way in which rows and columns from the input matrix are incorporated in biclusters, as defined in Section 2.2. Last three columns in the table report the discovery strategy, depending on the way in which biclusters are obtained: one at a time (the algorithm outputs one bicluster for each run), one set at a time (the algorithm outputs several biclusters for each run), or simultaneously (the algorithm outputs several biclusters for each run, and they have been evaluated together, being their evaluation influenced by the whole set of biclusters).

As it can be derived from Table 3, the most common type of pattern obtained in biclusters is coherent values, either additive (shifting pattern), multiplicative (scaling pattern), or both of them independently. Recently, several approaches have been proposed able to find both patters simultaneously (combined shifting and scaling pattern). Furthermore, several recent strategies also detect negative correlations in biclusters. This situation indicates that an effort is being currently done in biclustering research to obtain combined patterns together with negative correlations. Regarding biclusters structure, the most extended strategy is the search of non-exhaustive non-exclusive biclusters, where every gene or condition might be present in none or more than one bicluster. Finally, there is no settled criterion for the discovery strategy, existing a similar number of algorithms for each approach.

Biclustering algorithms performance comparison, regarding the quality of the obtained results, is still an open research field, mainly due to the different results possibilities. This way, very different biclusters can be considered as equally important, depending also on the specific application under study. Current research in this field is being focused on performance comparison using synthetic datasets, where the final solutions are known beforehand, or carrying out a biological validation analysis for real datasets. In this case, the most common information sources for microarray data analysis validation and interpretation are KEGG (*Kyoto Encyclopedia of Genes and Genomes*) [72] and GO (*Gene Ontology*) [73]. Although KEGG database has been used when external relationships such as biological pathways are involved in the study, the biological knowledge used in bicluster validations are mostly gene annotations from GO, where obtained biclusters are evaluated according to the significant terms to which they are annotated in the ontology. Typical validation of a bicluster consists in getting all GO terms annotated to any of the genes in the bicluster and then apply a statistical significance test to determine if each term appearance is relevant. *Term-for-Term* (TFT) analysis represents the standard method of performing statistical analysis

Table 2
Non metric-based biclustering algorithms.

Algorithm		Acronym	Ref.
Graph-based approaches	SAMBA	SAMBA	[13]
	MicroCluster	MicroC	[51]
	QUBIC	QUBIC	[52]
	CoBi	CoBi	[53]
One-way clustering-based approaches	Coupled Two-way Clustering	CTWC	[54]
	Interrelated Two-way Clustering	ITWC	[57]
Probabilistic models	Plaid Models	PM	[58]
	Rich Probabilistic Models	RPM	[59]
	Gibbs Sampling	GS	[60]
	Bayesian Biclustering Model	BBC	[61]
	Conserved Gene Expression Motifs	xMOTIFs	[62]
	cMonkey	cMonkey	[63]
	Penalized Plaid Model	PPM	[64]
Linear algebra	Spectral Biclustering	SB	[65]
	Iterative Signature Algorithm	ISA	[66]
	Non-smooth Non-negative Matrix Factorization	nsNMF	[67]
	Pattern-based Biclustering	BicPAM	[68]
Optimal reordering of rows and columns	OPSM	OPSM	[70]
	OREO	OREO	[71]

Table 3
Biclustering algorithms' features.

	Algorithn	Measure	Type pattern								Biclusters structure						Strategy		
			Constant	Constant columns	Coherent values	Additive coherent val.	Multiplicative coherent val.	Simultaneous coherent val.	Coherent evolutions	Negative correlations	Row and col. exhaustive	Row or column exclusive	Exhaustive	Non exhaustive	Exclusive	Non exclusive	One at a time	One set at a time	Simultaneous
Metric-based biclustering algorithms	DC	Var	•								•					•		•	
	CC	MSR				•								•		•	•		
	SMSR-CC	SMSR					•							•		•	•		
	HARP	RI	•	•									•		•				•
	MSB	MSB				•								•		•	•		
	WF-MSB	MSB				•								•		•	•		
	BISWC	PCC			•									•		•	•		
	BICLIC	PCC			•					•				•		•		•	
	ICS	SSSim						•						•		•	•		
	FLOC	MSR				•								•		•			•
	RWB	Var, MSR				•								•		•	•		
	RGRASP-B	MSR				•					•					•		•	
	PDNS	MSR, ASR			•									•		•	•		
	SA-B	MSR				•								•		•	•		
	CMOPSOB	Var, MSR				•								•		•			•
	MOM-aiNet	MSR				•								•		•			•
	Bleuler-B	MSR				•								•		•		•	
	SEBI	MSR, Var				•								•		•	•		
	BiHEA	MSR				•								•		•		•	
	CBEB	MSR				•								•		•			•
	EvoBexpa	VE^t, Var						•		•				•		•	•		
	M&B	MSR				•								•		•			•
	MOGAB	MSR, Var				•								•		•			•
	MOFB	MSR, Var				•								•		•			•
	SMOB	MSR, VE, Var			•									•		•	•		
	RGCE-B	MSR				•								•	•			•	
	PSB	MSR				•				•				•		•		•	
	SVD&HC-B	MSR						•		•				•		•		•	
Non metric-based biclustering algorithms	SAMBA	–							•					•		•			•
	MicroC	–			•									•		•			•
	QUBIC	–			•					•				•		•		•	
	CoBi	–			•					•				•		•		•	
	C2wC	–			•									•		•		•	
	I2wC	–			•						•	•						•	
	PM	–			•									•		•	•		
	RPM	–		•							•				•				•
	GS	–			•						•	•					•		
	BBC	–			•							•		•				•	
	xMOTIFs	–							•					•		•			•
	cMonkey	–			•									•		•	•		
	PPM	–			•									•		•			•
	SB	–			•								•			•			•
	ISA	–			•									•		•	•		
	NsNnMF	–			•								•		•				•
	BicPAM	–						•		•				•		•			•
	OPSM	–							•					•		•	•		
	OREO	–			•								•		•				•

Table 4
Computational costs. N: number of genes of the input microarray matrix; M: number of conditions of the input microarray matrix; N_{iter}: number of iterations.

Algorithm	Computational complexity	Additional information
CC	$O(N \times M)$	
SMSR-CC	$O(N \times M)$	
HARP	$O(N)$ $O(N^2)$	using Conga line as cache structure using quad tree or priority queue as cache structure

MSB	$O(N \times (N+M)^2)$
	$O(N \times M \times (N+M)^2)$
		for constant biclusters
		for additive biclusters

BISWC	$O(M^2)$	
ICS	$O(M^4 - mc^2 \times nb \times (N - nb/2))$	mc: average size of correlated subspaces nb: average number of genes in biclusters
FLOC	$O((N+M)^2 \times K \times N_{iter})$	K: number of seeds typically $N_{iter} \ll N+M$
RWB	$N_{iter} \times Cu \times [(1-P) \times (N+M) + P]$	P: probability of random move Cu: computing new residue ($Cu \ll max(I,J)$)
RGRASP-B	$O(N \times M \times (N+M))$	
M&B	$O(F \times P^2_{size})$	P_{size}: population size F: number of objetive functions
SVD&HC-B	$O(N \times M \times l + N^2 \times l \times I + M^2 \times l \times J + I \times J \times N \times M \times (N+M))$	l: number of singular values $I \times J$: size of the original biclusters
SAMBA	$O((N+M) \times {2^d)}^{log_{(r+1)/r} r \times d} \times \log k)$	k: number of biclusters
MicroC	$O(N \times M^2 + C \times \log(C))$	r: max weight ratio d: degree of vertices C: number of clusters
CoBi	$O(M \times N^2 + MaxIter \times \zeta \times C^2)$	$MaxIter$: max. number iterations ζ: number edges C: numner clusters under an edge
GS	$O(M \times N)$	
xMOTIFs	$O(N \times ns \times nd)$	ns: number of samples randomly selected nd: number of sets of genes for each sample
ISA	$O(N_{iter} \times Ni \times (Nc \times \widetilde{N}g + Ng \times \widetilde{N}c))$	Ni: input sets $\widetilde{N}g$: average number of genes $\widetilde{N}c$: average number of conditions
BicPAM	$O(d \times wp + \binom{k}{k/2} \times r \times s)$	k: number of biclusters $r \times s$: biclusters average size $d \times \wp$: are related to the input matrix size and the pattern types
OPSM	$O(N \times M^3 \times N_{mod})$	N_{mod}: number of models

for over-representation in GO, although other approaches for term enrichment include the *parent–child method* of Grossmann et al. [74], topology based methods as described in Alexa et al. [75] and Falcon and Gentleman [76]. Also, a new model-based approach is described in Bauer et al. [77]. Although GO is mostly used as the main source of biological information, this kind of validation presents two main drawbacks. Firstly, biological knowledge in GO is not complete. This way, when a bicluster does not group known GO annotations, it may be because it is a bad bicluster, or because GO annotations are not complete. Secondly, GO terms are organized in levels in a hierarchical structure, according to their specificity. Using this structure, bigger biclusters are more probable to be enriched for more generic GO terms, situated in the higher levels. This means that using this kind of validation, those algorithms obtaining bigger biclusters would be tagged as better than those obtaining smaller biclusters, less probable to be enriched for the more specific terms.

Taking into account these limitations, together with the impossibility of testing all the methods under review, we present in Table 4 a performance comparison based on the computational requirements for those authors who included this information in their corresponding papers. Although some other authors pointed out the average computational time of their approaches, we consider this information not relevant since it is dependant on many different factors, such as the input database size, machine specifications or the used programming language. Since biclustering problem is NP-hard, algorithms must make tradeoffs between results quality and computational complexity. The nature of these tradeoffs affects their runtime efficiency, which is especially relevant for analyzing large datasets. In general, greedy approaches can be considered as the fastest, while nature-inspired meta-heuristics are inherently more costly, although their computational cost may be reduced by applying different strategies, such as the hybridization with local searches. As it can be seen in Table 4, the computational cost of biclustering algorithms is determined by the size of the input matrix ($N \times M$), although for the majority of methods is also influenced by some other additional parameters, depending on the specific strategy, such as the number of iterations, probabilities, etc. (this information is given in the second column of the table).

Due to the current absence of an established technique for biclustering algorithms performance comparison, and the great variety of existing biclustering methodologies, it may be a difficult task to choose the most appropriate technique for a biclustering analysis. In this paper, we have reviewed the most important existing approaches, pointing out their most relevant characteristics, both referring the search strategy and the type of the obtained results. In order to organize the contents of this paper, and to present a summary of this study, we provide the readers with several tables that list the outcomes of our work, in the hope they guide the reader towards the most convenient biclustering method.

Acknowledgement

This research has been supported by the Spanish Ministry of Economy and Competitiveness under Grant TIN2011-28956.

References

[1] P. Baldi, G.W. Hatfield, DNA Microarrays and Gene Expression – From Experiments to Data Analysis and Modeling, Cambridge University Press, 2011. pp. 1–213.

[2] R. Harpaz, R. Haralick, Exploiting the geometry of gene expression patterns for unsupervised learning, 2006.

[3] G. Piatetsky-Shapiro, T. Khabaza, S. Ramaswamy, Capturing best practice for microarray gene expression data analysis, 2003, pp. 407–415.

[4] H. Wang, W. Wang, J. Yang, P. Yu, Clustering by pattern similarity in large data sets, in: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, ACM, 2002, pp. 394–405.

[5] A.P. Gasch, M.B. Eisen, Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering, Genome Biol. 3 (11) (2002). research0059.10059.2.

[6] S.C. Madeira, A.L. Oliveira, Biclustering algorithms for biological data analysis: a survey, IEEE Trans. Comput. Biol. Bioinform. 1 (2004) 24–25.

[7] A. Tanay, R. Sharan, R. Shamir, Biclustering Algorithms: A Survey, Handbook of Computational Molecular Biology 9, 2005, 26–1.

[8] S. Busygin, O.A. Prokopyev, P.M. Pardalos, Biclustering in data mining, Comput. OR 35 (9) (2008) 2964–2987.

[9] K. Eren, M. Deveci, O. Küçüktunç, Ü.V. Çatalyürek, A comparative analysis of biclustering algorithms for gene expression data, Briefings Bioinform. 14 (3) (2013) 279–292.

[10] A. Oghabian, S. Kilpinen, S. Hautaniemi, E. Czeizler, Biclustering methods: biological relevance and application in gene expression analysis, PloS One 9 (3) (2014) e90801.

[11] J. Hartigan, Direct clustering of a data matrix, J. Am. Stat. Assoc. 67 (337) (1972) 123–129.

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

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

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

[15] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, On biclustering of gene expression data, Current Bioinform. 5 (2010) 204–216.

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

[17] B. Pontes, R. Giráldez, J.S. Aguilar-Ruiz, Quality measures for gene expression biclusters, PloS One 10 (3) (2015). pp. e0115497–e0115497.

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

[19] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, A novel coherence measure for discovering scaling biclusters from gene expression data, J. Bioinform. Comput. Biol. 7 (5) (2009) 853–868.

[20] K. Yip, D. Cheung, M. Ng, Harp: a practical projected clustering algorithm, IEEE Trans. Knowl. Data Eng. 16 (11) (2004). 1387–1387.

[21] X. Liu, L. Wang, Computing the maximum similarity bi-clusters of gene expression data, Bioinformatics 23 (2007) 50–56.

[22] L.-C. Chen, P.S. Yu, V.S. Tseng, Wf-msb: a weighted fuzzy-based biclustering method for gene expression data, Int. J. Data Min. Bioinform. 5 (1) (2011) 89–109.

[23] L. Teng, L. Chan, Discovering biclusters by iteratively sorting with weighted correlation coefficient in gene expression data, Signal Process. Syst. 50 (3) (2008) 267–280.

[24] J. Bland, D. Altman, Calculating correlation coefficients with repeated observations: Part 2–Correlation between subjects, British Med. J. 310 (6980) (1995) 633.

[25] M. Pavan, M. Pelillo, A new graph-theoretic approach to clustering and segmentation, Proceedings: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, IEEE, 2003, pp. 1–145.

[26] T. Yun, G.-S. Yi, Biclustering for the comprehensive search of correlated gene expression patterns using clustered seed expansion, BMC Genom. 14 (1) (2013) 144.

[27] H. Ahmed, P. Mahanta, D. Bhattacharyya, J. Kalita, Shifting-and-scaling correlation based biclustering algorithm, IEEE/ACM Trans. Comput. Biol. Bioinform. 11 (6) (2014) 1239–1252.

[28] J. Yang, H. Wang, W. Wang, P.S. Yu, An improved biclustering method for analyzing gene expression profiles, Int. J. Artif. Intell. Tools 14 (2005) 771–790.

[29] F. Angiulli, E. Cesario, C. Pizzuti, Random walk biclustering for microarray data, Inform. Sci. 178 (6) (2008) 1479–1497.

[30] S. Dharan, A.S. Nair, Biclustering of gene expression data using reactive greedy randomized adaptive search procedure, BMC Bioinform. 10 (Suppl. 1) (2009) S27.

[31] W. Ayadi, M. Elloumi, J. Hao, Pattern-driven neighborhood search for biclustering of microarray data, BMC Bioinform. 13 (Suppl. 7) (2012) S11.

[32] W. Ayadi, M. Elloumi, J.-K. Hao, A biclustering algorithm based on a bicluster enumeration tree: application to dna microarray data, BioData Mining 2 (1) (2009) 9.

[33] S. Kirkpatrick, C. Gelatt Jr., M. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[34] K. Bryan, P. Cunningham, N. Bolshakova, Application of simulated annealing to the biclustering of gene expression data, IEEE Trans. Inform. Technol. Biomed. 10 (3) (2006) 519–525.

[35] J. Liu, Z. Li, X. Hu, Y. Chen, Biclustering of microarray data with mospo based on crowding distance, BMC Bioinform. 10 (Suppl. 4) (2009) S9.

[36] G.P. Coelho, F.O. de Franca, F.J.V. Zuben, Multi-objective biclustering: when non-dominated solutions are not enough, J. Math. Modell. Algorithms 8 (2) (2009) 175–202.

[37] L. de Castro, F. Von Zuben, aiNet: an artificial immune network for data analysis, Data Min.: A Heuristic Approach 1 (2001) 231–259.

[38] S. Bleuler, A. Prelic, E. Zitzler, An ea framework for biclustering of gene expression data, Congress on Evolutionary Computation, 2004 (CEC2004), vol. 1, IEEE, 2004, pp. 166–173.

[39] C. Gallo, J. Carballido, I. Ponzoni, Bihea: a hybrid evolutionary approach for microarray biclustering, Adv. Bioinform. Comput. Biol. (2009) 36–47.

[40] Q. Huang, D. Tao, X. Li, A.W.-C. Liew, Parallelized evolutionary learning for detection of biclusters in gene expression data, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (2012) 560–570.

[41] B. Pontes, R. Giráldez, J.S. Aguilar-Ruiz, Configurable pattern-based evolutionary biclustering of gene expression data, Algorithms Molec. Biol. 8 (1) (2013) 4.

[42] S. Mitra, H. Banka, Multi-objective evolutionary biclustering of gene expression data, Pattern Recogn. 39 (12) (2006) 2464–2477. bioinformatic.

[43] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii, Lecture Notes Comput. Sci. 1917 (2000) 849–858.

[44] U. Maulik, A. Mukhopadhyay, S. Bandyopadhyay, Finding multiple coherent biclusters in microarray data using variable string length multiobjective genetic algorithm, IEEE Trans. Inform. Technol. Biomed. 13 (6) (2009) 969–975.

[45] U. Maulik, A. Mukhopadhyay, S. Bandyopadhyay, M.Q. Zhang, X. Zhang, Multiobjective fuzzy biclustering in microarray data: method and a new performance measure, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2008, June 1–6, 2008, Hong Kong, China, 2008, pp. 1536–1543.

[46] F. Divina, B. Pontes, R. Giráldez, J.S. Aguilar-Ruiz, An effective measure for assessing the quality of biclusters, Comput. Biol. Med. 42 (2) (2012) 245–256.

[47] C. Cano, L. Adarve, J. López, A. Blanco, Possibilistic approach for biclustering microarray data, Comput. Biol. Med. 37 (10) (2007) 1426–1436.

[48] J. Zhang, Y. Leung, Improved possibilistic c-means clustering algorithms, IEEE Trans. Fuzzy Syst. 12 (2) (2004) 209–217.

[49] W.-H. Yang, D.-Q. Dai, H. Yan, Finding correlated biclusters from gene expression data, IEEE Trans. Knowl. Data Eng. 23 (2011) 568–584.

[50] D. Yan, J. Wang, Biclustering of gene expression data based on related genes and conditions extraction, Pattern Recogn. 46 (4) (2013) 1170–1182.

[51] L. Zhao, M. Zaki, Microcluster: efficient deterministic biclustering of microarray data, IEEE Intell. Syst. 20 (6) (2005) 40–49.

[52] G. Li, Q. Ma, H. Tang, A.H. Paterson, Y. Xu, Qubic: a qualitative biclustering algorithm for analyses of gene expression data, Nucleic Acids Res. 37 (15) (2009). 101–101.

[53] S. Roy, D.K. Bhattacharyya, J.K. Kalita, Cobi: pattern based co-regulated biclustering of gene expression data, Pattern Recogn. Lett. 34 (14) (2013) 1669–1678.

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

[55] J.A. Hartigan, Clustering Algorithms, Wiley, 1975. pp. 1–364.

[56] M. Blatt, S. Wiseman, E. Domany, Superparamagnetic clustering of data, Phys. Rev. Lett. 76 (1996) 3251–3254, http://dx.doi.org/10.1103/PhysRevLett.76.3251.

[57] C. Tang, A. Zhang, Interrelated two-way clustering and its application on gene expression data, Int. J. Artif. Intell. Tools 14 (4) (2005) 577.

[58] L. Lazzeroni, A. Owen, Plaid models for gene expression data, Stat. Sinica 12 (1) (2002) 61–86.

[59] E. Segal, B. Taskar, A. Gash, N. Friedman, D. Koller, Rich probabilistic models for gene expression, Bioinformatics 17 (2001) 243–252.

[60] Q. Sheng, Y. Moreau, B. De Moor, Biclustering microarray data by gibbs sampling, Bioinformatics 19 (Suppl. 2) (2003) 196–205.

[61] J. Gu, J. Liu, Bayesian biclustering of gene expression data, BMC Genom. 9 (Suppl. 1) (2008) S4.

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

[63] N.S.B. David J. Reiss, R. Bonneau, Integrated biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks, BMC Bioinform. 7 (2006) 280.
[64] T. Chekouo, A. Murua, The penalized biclustering model and related algorithms, J. Appl. Stat. (2015) 1–23 (ahead-of-print).
[65] Y. Kluger, R. Basri, J. Chang, M. Gerstein, Spectral biclustering of microarray data: coclustering genes and conditions, Genome Res. 13 (2003) 703–716.
[66] S. Bergmann, J. Ihmels, N. Barkai, Iterative signature algorithm for the analysis of large-scale gene expression data, Phys. Rev. E 67 (3) (2003) 031902.
[67] P. Carmona-Saez, R. Pascual-Marqui, F. Tirado, J. Carazo, A. Pascual-Montano, Biclustering of gene expression data by non-smooth non-negative matrix factorization, BMC Bioinform. 7 (1) (2006) 78.
[68] R. Henriques, S.C. Madeira, Bicpam: pattern-based biclustering for biomedical data analysis, Algorithms Molec. Biol. 9 (1) (2014) 27.
[69] R. Henriques, S.C. Madeira, C. Antunes, F2g: efficient discovery of full-patterns, ECML/PKDD nfMCP (2013) 1–9.
[70] A. Ben-Dor, B. Chor, R.M. Karp, Z. Yakhini, Discovering local structure in gene expression data: the order-preserving submatrix problem, J. Comput. Biol. 10 (3–4) (2003) 373–384.
[71] P. DiMaggio, S. McAllister, C. Floudas, X. Feng, J. Rabinowitz, H. Rabitz, Biclustering via optimal re-ordering of data matrices in systems biology: rigorous methods and comparative studies, BMC Bioinform. 9 (1) (2008) 458.
[72] M. Kanehisa, S. Goto, KEGG: Kyoto Encyclopedia of Genes and Genomes, Nucleic Acids Res. 28 (2000) 27–30.
[73] M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, et al., Gene ontology: tool for the unification of biology. The Gene Ontology, Nature Genet. 25 (2000) 25–29.
[74] S. Grossmann, S. Bauer, P. Robinson, M. Vingron, Improved detection of overrepresentation of gene-ontology annotations with parent–child analysis, Bioinformatics 23 (22) (2007) 3024–3031.
[75] A. Alexa, J. Rahnenführer, T. Lengauer, Improved scoring of functional groups from gene expression data by decorrelating go graph structure, Bioinformatics 22 (13) (2006) 1600–1607.
[76] S. Falcon, R. Gentleman, Using gostats to test gene lists for go term association, Bioinformatics 23 (2) (2007) 257–258.
[77] S. Bauer, J. Gagneur, P. Robinson, Going bayesian: model-based gene set analysis of genome-scale data, Nucleic Acids Res. 38 (11) (2010) 3523–3532.