A COMPARATIVE STUDY OF CLUSTERING AND BICLUSTERING OF MICROARRAY DATA

Haifa Ben Saber¹,² and Mourad Elloumi¹,³

¹Laboratory of Technologies of Information and Communication and Electrical Engineering (LaTICE) at National Superior School of Engineers of Tunis (ENSIT) - Tunis University, Tunis, Tunisia
²Time Université - Tunis University, Tunis, Tunisia
³University of Tunis-El Manar, Tunisia

ABSTRACT

There are subsets of genes that have similar behavior under subsets of conditions, so we say that they coexpress, but behave independently under other subsets of conditions. Discovering such coexpressions can be helpful to uncover genomic knowledge such as gene networks or gene interactions. That is why, it is of utmost importance to make a simultaneous clustering of genes and conditions to identify clusters of genes that are coexpressed under clusters of conditions. This type of clustering is called biclustering.

Biclustering is an NP-hard problem. Consequently, heuristic algorithms are typically used to approximate this problem by finding suboptimal solutions. In this paper, we make a new survey on clustering and biclustering of gene expression data, also called microarray data.

KEYWORDS

Clustering, biclustering, heuristic algorithms, microarray data, genomic knowledge.

1. INTRODUCTION

A DNA Microarray is a glass slide covered with a chemical product and DNA samples containing thousands of genes. By placing this glass slide under a scanner, we obtain an image in which coloured dots represent the expression level of genes under experimental conditions [1]. As shown in Figure 1, the obtained coloured image can be coded by a matrix $M$, called gene expression data, or microarray data, where the $i^{th}$ row represents the $i^{th}$ gene, the $j^{th}$ column represents the $j^{th}$ condition and the cell $m_{ij}$ represents the expression level of the $i^{th}$ gene under the $j^{th}$ condition. Simultaneous clustering of rows (genes) and columns (conditions) of this matrix enables to identify subsets of genes that have similar behaviour under subsets of conditions, so we say that they coexpress, but behave independently under other subsets of conditions. This type of clustering is called biclustering. Biclustering of microarray data can be helpful to discover coexpression of genes and, hence, uncover genomic knowledge such as gene networks or gene interactions. Biclustering is an NP-hard problem [3]. Consequently, heuristic algorithms are typically used to approximate this problem by finding suboptimal solutions. In this paper, we make a new survey on bi clustering of microarray data.

In this paper, we make a brief survey on clustering algorithms of microarray data. There are three main types of clustering algorithms: Geometric, model-based and formal concepts based. So, the
rest of the chapter is organized as follows: In the first part, we briefly review geometric clustering algorithms. Then, we present model-based clustering algorithms.

After that, we present formal concepts based clustering algorithms. Finally, we review some clustering web tools and microarrays datasets commonly used. In the second part, we make a survey on biclustering of gene expression data. First, we introduce some definitions related to biclustering of microarray data. Then, we present some evaluation functions and biclustering algorithms. Next, we show how to validate biclusters via biclustering tools on microarrays datasets. Finally, we present our conclusion.

Figure 1. Coding of the generated colored image to a microarray data

2. CLUSTERING OF MICROARRAY DATA

Let introduce some definitions related to a biclustering of microarray data [3].

2.1 Geometric Clustering Approaches

In geometric clustering approaches, we distinguish hierarchical and partitioning clustering approaches.

2.1.1 Hierarchical Clustering approaches

The aim of hierarchical approaches is to create a hierarchical decomposition of a set of objects \( E \). On \( E \), we have a dissimilarity measure such that the closest objects are grouped in the clusters with the smallest index. There exist two principal approaches: divisive and agglomerative that we describe hereafter; for details see for instance.

- Divisive approach: this approach is also called top-down approach. We start with just one cluster containing all objects. In each successive iteration, we split up clusters into two or more clusters until generally each object is in one cluster. Note that other stop conditions can be used and the division into clusters are defined by the verification or not of a property.
• Agglomerative approach: opposed to the divisive approach, we start by assuming n clusters, each object forms a singleton cluster. In each successive iteration, we merge the closest clusters until obtaining one cluster which is the set E. In the following, we focus on this approach which is the most frequently used.

The advantages of agglomerative hierarchical clustering algorithms are:
• They can produce an ordering of the objects (genes/conditions), which may be informative for data display.
• They generate smaller clusters which can be helpful for analysis. The drawbacks of agglomerative hierarchical clustering algorithms are:
• They cannot relocate objects (genes/conditions) that may have been badly clustered at an early stage.
• The use of different metrics for measuring distances between clusters may generate different results.

2.1.2 Partitioning Clustering Approaches

Partitioning clustering consists in splitting the objects (genes/conditions) into K homogeneous clusters. The most popular partitioning algorithms are K-means and Self-Organizing Map (SOM).

• The K-means algorithm is one of the simplest clustering algorithm. Actually, the K-means version commonly used is due to Forgy [forgy65]. It follows a simple and easy way to cluster given objects (genes or conditions) in a number K of clusters (K is fixed a priori).
• The SOM algorithm, introduced by Kohonen [7,15], can be viewed as a spatially smoothed version of K-means algorithm clustering. SOM operates as follows: First, it randomly choose nodes, i.e., centers of clusters of genes. Then, at each iteration, it chooses an object (gene) and finds a node that is the closest to the object, according to the Euclidean distance. If the object is not closest to the node of its own cluster then SOM moves it into the cluster of the closest node and defines the new nodes of both clusters. SOM repeats this process until no object moves from one cluster to another. By its simplicity SOM has much success for decades even if until now, there is not any criterion whose optimization implies the formula of the updates of the cluster means. Actually, it does not exist the proof of the convergence in the general context, the convergence has been proved only on special cases and in particular for the one dimensional data.

2.2 Model-Based Clustering Approaches

The model-based clustering approaches can be used for different type of data by using appropriated mixtures such as Gaussian, von Mises-Fisher, multinomial and Bernoulli mixtures. The model-based clustering considers two approaches: the Maximum Likelihood (ML) and the Classification Maximum Likelihood (CML) approaches. The former is based on the maximization of the observed likelihood of data, and the latter one is based on the maximization of the Classification (or complete data) likelihood. These maximizations can be performed respectively by the Expectation Maximization (EM) and the Classification EM (CEM) algorithms. Note that this approach offers considerable flexibility, and provides solutions to the problem of the number of clusters. Its associated estimators of posterior probabilities give rise to a fuzzy or hard clustering using the Maximum A Posteriori principle (MAP). Hereafter we review the definition of the mixture model and classical clustering algorithms used.

2.2.1 Finite Mixture Model

The finite mixture models underpin a variety of techniques in major areas of statistics including cluster analysis. With a mixture model-based approach clustering, it is assumed that the data to be clustered are generated by a mixture of underlying probability distributions in which each component represents a different cluster.
2.2.2 EM Algorithm

The EM algorithm is a method for maximizing the log-likelihood iteratively, using the maximization of the conditional expectation of the complete-data log-likelihood given a previous current estimate and the observed data. In mixture modeling, we take the complete-data to be the vector \((x,z)\).

2.2.3 Classification EM Algorithm

The Classification EM (CEM) algorithm is seen as a hard version of EM. The main modifications imported to EM concern therefore the conditional maximization of complete data log-likelihoods with respect to \(z\), a classification step is introduced between the two steps E and M. The classification step consists in assigning each point \(x_i\) to the component which maximizes the conditional probability \(s_{ik}\). Hence, the CEM estimators depends on the parameter of the chosen distribution. In particular, note that when we consider the Gaussian mixture model under certain constraints, CEM is an extension of the K-means algorithm.

2.2.4 Stochastic EM Algorithm

The Stochastic EM algorithm (SEM) is a stochastic version of EM, it incorporates between the E and the M steps a restoration of the unknown component labels by drawing them at random from their current conditional distribution, starting from an initial parameter, it is a stochastic step S, in which the algorithm assigns each point at random to one of the mixture components according to the multinomial distribution with parameters the values of the posterior probabilities. Note that this stochastic version does not converge point wise, SEM generates a Markov chain in which the distribution is more or less concentrated around the ML estimates.

2.2.5 Formal Concepts Based Clustering Approaches

The Analysis of Formal Concepts (AFC) [21] is a domain of applied mathematics which restructures the theory of the lattices to facilitate its use in applications of the real world and to as well allow the interpretation of its concepts from the theoretical framework by mathematicians as by non-mathematicians. Basic notions of lattices theory are binary relation, formal context and formal concept.

The construction of the Galois lattice of a binary relation can be broken up into three steps, namely: the enumeration of the maximum rectangles (closed), the search of a partial order relation between these rectangles, and the construction of the lattice chart. We distinguish three types of formal concepts based clustering algorithms{}: batch, incremental and assembly algorithms, by considering the criterion of distribution of the algorithms, according to their strategies of data acquisition starting from a formal context.

2.2.5.1 Batch Algorithms

It is the first generation of the algorithms of extraction of the Galois lattices. By taking in entry the entire formal context, these algorithms calculate the formal concepts and the order between these concepts simultaneously or sequentially. Among most known of these algorithms, we quote the algorithm of Chein generating the concepts by levels: the algorithm is iterative contain less than two elements. The elements not removed after the stop of the algorithm are the concepts of the formal context considered.
The incremental algorithms consider the formal context line by line (or column by column) and build the Galois lattice by successive additions of line or column while preserving its structure.

2.2.5.3 Assembly algorithms

These algorithms constitute an evolution of the incremental algorithms which generalize the incremental character to set (groups) objects/attributes [46]. They divide a formal context into two parts vertically or horizontally then calculate the lattice of concepts corresponding to each part and finally assemble the lattices obtained in only one;

3. BICLUSTERING OF MICROARRAY DATA

Let introduce some definitions related to a biclustering of microarray data.

**Biclusters**: Let \( I = \{1,2,...,n\} \) be a set of indices of \( n \) genes, \( J = \{1,2,...,m\} \) be a set of indices of \( m \) conditions and \( M(I,J) \) be a data matrix associated with \( I \) and \( J \). A bicluster associated with the data matrix \( M(I,J) \) is a couple \( (I',J') \) such that \( I' \subseteq I \) and \( J' \subseteq J \).

**Types of biclusters**: A bicluster can be one of the following cases:

- Bicluster with constant values on rows:
  \[ m_{ij} = c + a_i \]  
  \[ m_{ij} = c * a_i \]
  where \( c \) is a constant and \( a_i \) is the adjustment for the row \( i \).

- Bicluster with constant values on columns:
  \[ m_{ij} = c + b_j \]
  \[ m_{ij} = c * b_j \]
  where \( b_j \) is the adjustment for the column \( j \).

- Bicluster with coherent values: There are two types of biclusters with coherent values. Those with additive model and those with multiplicative model defined respectively by:
  Those with additive model:
  \[ m_{ij} = c + a_i + b_j \]
  And those with multiplicative model:
  \[ m_{ij} = c * a_i * b_j \]

- Bicluster with coherent evolution: It is a bicluster where all the rows (resp. columns) induce a linear order across a subset of columns (resp. rows).

**Groups of biclusters**: A group of biclusters can be one of the following types [4]:

1. Single bicluster (Figure 2. (a)),
2. Exclusive rows and columns group of biclusters (Figure 2. (b)),

97
3. Non-overlapping group of biclusters with checkerboard structure (Figure 3. (c)),
4. Exclusive rows group of biclusters (Figure 2. (d)),
5. Exclusive columns group of biclusters (Figure 2. (e)),
6. Non-overlapping group of biclusters with tree structure (Figure 2. (f)),
7. Non-overlapping non-exclusive group of biclusters (Figure 2. (g)),
8. Overlapping group of biclusters with hierarchical structure (Figure 2. (h)),
9. Or, arbitrarily positioned overlapping group of biclusters (Figure 2. (i)).

We note also that a natural way to visualize a group of biclusters consists in assigning a different color to each bicluster and in reordering the rows and the columns of the data matrix so that we obtain a data matrix with colored blocks, where each block represents a bicluster. The biclustering problem can be formulated as follows: Given a data matrix $M$, construct a group of biclusters $B_{opt}$ associated with $M$ such that:

\[ f(B_{opt}) = \max_{B \in \text{BC}(M)} f(B) \]  

(2.7)

where $f$ is an objective function measuring the quality, i.e., degree of coherence, of a group of biclusters and $\text{BC}(M)$ is the set of all the possible groups of biclusters associated with $M$. This problem is NP-hard [4,5].

4. EVALUATION FUNCTIONS

An evaluation function is an indicator of the performance of a biclustering algorithm. There are two main classes of evaluation functions: Intra-biclusters evaluation functions and inter-biclusters evaluation functions.

4.1. Intra-biclusters evaluation functions

An intra-biclusters evaluation function is a function that measures the quality of a bicluster, i.e., it quantifies the coherence degree of a bicluster. There are several intra-biclusters evaluation functions.
• The $E_{AVSS}(I',J')$ is defined as follows [6]:

$$E_{AVSS}(I',J') = \frac{\sum_{i \in I'} \sum_{j \in J'} s_{ij}}{|I'||J'|}$$  \hspace{1cm} (3.1)

where $(I',J')$ is a bicluster, $s_{ij}$ is a similarity measure among elements of the row $i$ and the column $j$ with others elements belonging to $I'$ and $J'$. It follows that a number of these functions are particular cases of the Average Similarity Score (AVSS).

• The Average Row Variance (ARV) is defined as follows [7]:

$$E_{ARV}(I',J') = \frac{\sum_{i \in I'} \sum_{j \in J'} (m_{ij} - m_{i'})^2}{|I'||J'|}$$  \hspace{1cm} (3.2)

where $m_{i'}$ is the average over the row $i$. It follows that the biclusters that contain rows with large changes in their values for different columns are characterized by a large row variance. The ARV guarantees that a bicluster captures rows exhibiting coherent trends under some subset columns.

• The Mean Squared Residue (MSR) is defined as follows [8]:

$$E_{MSR}(I',J') = \frac{\sum_{i \in I'} \sum_{j \in J'} (m_{ij} - m_{i'} - m_{j'} + m_{i'} m_{j'})^2}{|I'||J'|}$$  \hspace{1cm} (3.3)

where $m_{i'}$ is the average over the whole bicluster, $m_{j'}$ is the average over the column $j$, $m_{i'}$ is the average over the row $i$. The $E_{MSR}$ represents the variation associated with the interaction between the rows and the columns in the bicluster. It follows that a low (resp. high) $E_{MSR}$ value, i.e., close to 0 (resp. higher than a fixed threshold $d$), indicates that the bicluster is strongly (resp. weakly) coherent. The $E_{MSR}$ function is inadequate to assess certain types of biclusters. For example, the $E_{MSR}$ function is good for biclusters of coherent values with additive model but not for coherent values with multiplicative model.

• The Volume (V) is defined as follows [7]:

$$E_{V}(I',J') = |I'||J'|$$  \hspace{1cm} (3.4)

This function enables to have the maximum-sized bicluster that does not exceed a certain coherence value expressed as a MSR score. $E_{i}(I',J')$ finds the maximum-sized bicluster that does not exceed a certain coherence value [9] expressed as a MSR score. Hence, discovered biclusters have a high $E_{i}(I',J')$ maximized and lower $E_{MSR}$ than a given threshold $\delta \geq 0$.

• The Mean Square Error (MSE) is defined as follows [10]:

$$E_{MSE}(I,J) = \frac{\sum_{i \in I} \sum_{j \in J} (m_{ij} - m_{i'} - m_{j'} + m_{ij})^2}{|I||J|}$$  \hspace{1cm} (3.5)

where $m_{ij}$ is the average over the whole matrix, $m_{i'}$ is the average over the column $j$ of the whole matrix and $m_{i'}$ is the average over the row $i$. This function identifies constant biclusters.

• The Average Correlation Value (ACV) is defined as follows [5, 11]:
\[
E_{ACV}(J', J'') = \max \left\{ \sum_{i \in J'} \sum_{j \in J'} \frac{|r_{ij}| - |J'|}{|J'||J'| - 1}, \sum_{k \in J''} \sum_{l \in J''} \frac{|r_{kl}| - |J''|}{|J''||J''| - 1} \right\}
\] (3.6)

where \( r_{ij} \) (resp. \( r_{kl} \)) is the Pearson’s correlation coefficient associated with the row indices \( i \) and \( j \) (resp. \( k \) and \( l \)) in the bicluster \((J', J'')\) [8]. The values of \( E_{ACV} \) belong to \([0;1]\), hence, a high (resp. low) \( E_{ACV} \) value, i.e., close to 1 (resp. close to 0), indicates that the bicluster is strongly (resp. weakly) coherent. However, the performance of the \( E_{ACV} \) function decreases when noise exists in the data matrix [5, 11].

- The **Average Spearman’s Rho** (ASR) is defined as follows [2]:

\[
E_{ASR}(J', J'') = 2 \max \left\{ \frac{\sum_{i \in J'} \sum_{j = i+1}^{i+1} \rho_{ij} - |J'|}{|J'||J'| - 1}, \frac{\sum_{k \in J''} \sum_{l = k+1}^{k+1} \rho_{kl} - |J''|}{|J''||J''| - 1} \right\}
\] (3.7)

where \( \rho_{ij} \) (resp. \( \rho_{kl} \)) is the Spearman’s rank correlation associated with the row indices \( i \) and \( j \) in the bicluster \((J', J'')\) [12]. The values of the \( E_{ASR} \) function belong also to \([-1,1]\), hence, a high (resp. low) \( E_{ASR} \) value, i.e., close to 1 (resp. close to -1), indicates that the bicluster is strongly (resp. weakly) coherent. On the other hand, like Spearman’s rank correlation, the \( E_{ASR} \) is less sensitive to the presence of noise in data [2]. There are other intra-biclusters evaluation function like the **Average Correspondance Similarity Index** (ACSI) [2].

### 4.2. Inter-biclusters evaluation functions

An inter-biclusters evaluation function is a function that measures the quality of a group of biclusters, i.e., it assesses the accuracy of an algorithm to recover true implanted biclusters in a data matrix. There are several inter-biclusters evaluation functions. In what follows, we present some of them:

Let \( M_1 \) and \( M_2 \) be two groups of biclusters defined as follows:

\[
M_1 = \{ B_{1}^{(1)}, B_{2}^{(1)}, ..., B_{K_1}^{(1)} \}, \text{ where } B_{1}^{(1)} = (G_{1}^{(1)}, C_{1}^{(1)}), G_{1} \text{ and } C_{1} \text{ are respectively the } p^{th} \text{ gene and condition, } 1 \leq l \leq K_1: \text{ Set of true implanted biclusters in a data matrix } M, \\
M_2 = \{ B_{1}^{(2)}, B_{2}^{(2)}, ..., B_{K_2}^{(2)} \}, \text{ where } B_{m}^{(2)} = (G_{m}^{(2)}, C_{m}^{(2)}), G_{m} \text{ and } C_{m} \text{ are respectively the } m^{th} \text{ gene and condition, } 1 \leq m \leq K_2: \text{ Set of the biclusters extracted by a biclustering algorithm.}
\]

- The **Prelic** index is defined as follows:

\[
I_{Prelic}(M_1, M_2) = \frac{1}{K_1} \sum_{i=1}^{K_1} \max_{j} S_{Prelic}(B_{i}^{(1)}, B_{j}^{(2)})
\] (3.8)

where \( S_{Prelic} \) is based on the **Jaccard index** for two sets and defined as follows:

\[
S_{Prelic}(B_{i}, B_{j}) = \frac{|G_{i} \cap G_{j}|}{|G_{i} \cup G_{j}|}
\] (3.9)

This index compares two solutions based on categorization of genes. However, it compares only genes sets.

- The **Liu and Wang index** is defined as follows:
Where

\[ I_{\text{Liu&Wang}}(M_1, M_2) = \frac{1}{K_1} \sum_{i=1}^{K_1} \max_j S_{\text{Liu&Wang}}(B^{(1)}_i, B^{(2)}_j) \]  

(3.10)

\[ S_{\text{Liu&Wang}}(B_i, B_j) = \frac{|G_i \cap G_j| + |C_i \cap C_j|}{|G_i \cup G_j| + |C_i \cup C_j|} \]  

(3.11)

It compares two solutions by considering both genes and conditions.

• The \( wtjaccard \) index is defined as follows:

\[ I_{\text{wtjaccard}}(M_1, M_2) = \frac{\sum_{i=1}^{K_1} |B^{(1)}_i| \cdot \max_j S_{\text{jaccard}}(B^{(1)}_i, B^{(2)}_j)}{\sum_{i=1}^{K_1} |B^{(1)}_i|} \]  

(3.12)

Where

\[ S_{\text{jaccard}}(B_i, B_j) = \frac{|C_i \cap B_j| + |G_i \cap G_j|}{|C_i| + |B_j| - |C_i \cap C_j|} \]  

(3.13)

• The \( Dice \) index is defined as follows:

\[ I_{\text{Dice}}(M_1, M_2) = \frac{1}{K_1} \sum_{i=1}^{K_1} \max_j S_{\text{Dice}}(B^{(1)}_i, B^{(2)}_j) \]  

(3.14)

Where:

\[ S_{\text{Dice}}(B_i, B_j) = \frac{2 \cdot |C_i \cap C_j|}{|C_i| + |C_j|} \]  

(3.15)

Which is proposed in [13] and called F-measure in biclustering cases to computes the overall relevance of two bicluster solutions.

• The Santamaria index is defined as follows:

\[ I_{\text{wtDice}}(M_1, M_2) = \frac{\sum_{i=1}^{K_1} |B^{(1)}_i| \cdot \max_j S_{\text{Dice}}(B^{(1)}_i, B^{(2)}_j)}{\sum_{i=1}^{K_1} |B^{(1)}_i|} \]  

(3.16)

The Santamaria index is the most conservative index among above others indices and used for biclustering case [14, 13]. In fact, while the Prelic index compares only object sets and the LW index compares object sets and feature sets independently, the Santamaria index compares two solutions using pairs of genes and conditions.

For gene expression case, the \( Gene \ Match \ Score \) (GMS) function doesn’t take into account column match. It is given by:

\[ E_{\text{GMS}}(B_1, B_2) = \frac{1}{|B_1|} \sum_{(i, j) \in B_1 \times B_2} \max_{B_2} \frac{|I_i \cap I_j|}{|I_i \cup I_j|} \]  

(3.17)
Where $B_1$ and $B_2$ are two groups of biclusters and the pair $(I,J)$ represents the submatrix whose rows and columns are given by the set $I$ and $J$, respectively.

The Row and Column Match Scores (RCMS) assess the method’s accuracy to recover known biclusters and reveal true ones. Thereafter, more similar measures of match scores have been introduced [5, 15, 6]. For instance, the evaluation functions, herein called Row and Column Match Scores, $E_{RCMS1}$ and $E_{RCMS2}$, are proposed in [6] and [15], respectively and given by:

$$E_{RCMS1}(B_1,B_2) = \frac{1}{|B_1|} \sum_{(I,J) \in B_1} \max_{(I,J) \in B_2} \frac{|I \cap J| + |I \cup J|}{|I| + |J|}$$

$$E_{RCMS2}(B_1,B_2) = \frac{1}{|B_1|} \sum_{(I,J) \in B_1} \max_{(I,J) \in B_2} \frac{|I \cap J| + |I \cap J|}{|I| + |J|}$$

All these measures of match score are used to assess the accuracy of an algorithm to recover known biclusters and reveal true ones. Both $E_{RCMS1}$ and $E_{RCMS2}$ have the advantage of reflecting simultaneously, the match of the row and column dimensions between biclusters as opposed to $E_{GMS}$ that doesn’t take into account column match. They vary between 0 and 1 (the higher the better the accuracy). Let $B_{opt}$ denote the set of true implanted biclusters in the data matrix $M$ and $B$ the set of the output biclusters of a biclustering algorithm. Thus, $E_{GMS}(B_{opt},B)$ and $E_{RCMS1}(B_{opt},B)$ express how well each of the true biclusters are detected by the algorithm under consideration. $E_{RCMS2}(B_X,B_Y)$, where $B_X$ (resp. $B_Y$) denotes the set of biclusters detected by the algorithm $X$ (resp. Algorithm $Y$), has the particularity to allow the quantification of how well each bicluster identified by the algorithm $X$ is contained into some bicluster detected by the algorithm $Y$.

### 4.3 BICLUSTERING ALGORITHMS

As we mentioned earlier, the biclustering problem is NP-hard [3, 10]. Consequently, heuristic algorithms are typically used to approximate the problem by finding suboptimal solutions. We distinguish different approaches adopted by biclustering approaches [3].

#### 4.3.1 Iterative Row and Column Clustering Combination Approach

By adopting the Iterative Row and Column Clustering Combination Approach (IRCCC) approach, we apply clustering algorithms on both rows and columns separately and then combine the results to obtain biclusters [56]. The conceptually simpler way to perform biclustering using existing algorithms without searching novels algorithms. But, this approach consider approximatively same advantages and drawbacks that clustering algorithms used. Among the algorithms adopting this approach we mention Croki2 [58], Crobin [58], DCC [59], ITWC [61], CTWC [54] and Bi-SOM [60].

#### 4.3.2 Greedy Iterative Search Approach

By adopting the Greedy Iterative Search (GIS), first, we construct sub matrices of the data matrix by adding/removing a row/column to/from the current sub matrix that optimizes a certain function. Then, we reiterate this process until no other row/column can be added/removed to/from any sub matrix. This approach presents the same advantage and drawback as DC. They may make wrong decisions and loose good biclusters, but they have the potential to be very fast. Among the algorithms adopting this approach we mention Spectral [16], Quest [17], RandomWalkBiclustering [18], BicFinder [19], MSB [6], ISA [17, 20], OPSM [21] and SAMBA [17, 22].
4.3.3 Exhaustive Bicluster Enumeration Approach

By adopting the Exhaustive Bicluster Enumeration (EBE), we identify all the possible groups of biclusters in order to keep the best one, i.e., the one that optimizes a certain evaluation function. The advantage of this approach is that it is able to obtain the best solutions. Its drawback is that it is costly in computing time and memory space. Among the algorithms adopting this approach we mention BSGP [28, 29], OPC [30, 6], CPB [30], IT [31], e-Bmotif [29], BIMODULE [32], RAP [26], BBK [33] and MSB [6].

4.3.4 Distribution Parameter Identification Approach

By adopting the Distribution Parameter Identification (DPI) approach, we use a statistical model to identify the distribution parameters and generate the data by minimizing a certain criterion iteratively. These algorithms certainly find the best biclusters, if they exist, but have a very serious drawback. Due to their high complexity, they can only be executed by assuming restrictions on the size of the biclusters. Among the algorithms adopting this approach we mention QUBIC [38], PRMs [39], FABIA [40], BEM [41] and BCEM [42].

4.3.5 Divide and Conquer Approach

By adopting the Divide-and-Conquer (DC) approach, first, we start by a bicluster representing the whole data matrix then we partition this matrix in two submatrices to obtain two biclusters. Next, we reiterate recursively this process until we obtain a certain number of biclusters verifying a specific set of properties. The advantage of DC is that it is fast, its drawback is that it may ignore good biclusters by partitioning them before identifying them. DC algorithms have the significant advantage of being potentially very fast. However, they have the very significant drawback of being likely to miss good biclusters that may be split before they can be identified. Among the algorithms adopting this approach we mention OWS [48], TWS [49], BiBit [28] and BARTMAP [50] and GS [51].

5. BICLUSTERING VALIDATION

There are two types of biclusters validation;

(i) Statistical validation: It is used to validate synthetical data
(ii) Biological validation: It is used to validate biological data

5.1. Statistical validation

Statistical validation can be made by adopting one or many of the following indices:

• Separation: It reflects how well the biclusters are separated from each other. Separation between two biclusters

\[ Sep(A, B) = 1 - \frac{A \cap B}{A \cup B} \]  

\[ \text{(5.1)} \]

• Coverage: We distinguish three types of coverage, matrix coverage, genes coverage and conditions coverage:
• **Compactness**: It assesses cluster homogeneity, with intra-cluster variance [63].

• **Connectedness**: It assesses how well a given partitioning groups data items together with their nearest neighbours in the data space [63].

• **Coherence**: It expresses how well a bicluster is fitted to a specified model. The coherence is computed thanks to compactness and connectedness.

• **Significance**: It is computed thanks to $p$-value. Let $B$ be a bicluster, $p$-value is defined as follows [15]:

$$p\text{-value}_B = 1 - \phi\left(\frac{|1_B|/|B| - p}{\sqrt{\frac{p(1-p)}{|B|}}}\right)$$  \hspace{2cm} (5.5)

where $\phi$ is the standard normal distribution function, $|1_B|$ is the number of 1’s in the bicluster $B$ and $p = k(|I|*|J|)$ of 1’s in $M(I,J)$, $k$ is the number of 1’s in the binary matrix $M_b$. A bicluster $B$ is considered as potentially significant at a level of significance $\alpha$ if $p\text{-value}_B < \alpha$.

### 5.2. Biological validation

Biological validation can qualitatively evaluate the capacity of an algorithm to extract meaningful biclusters from a biological point of view. To assess biologically biclusters, we can use Gene Ontology (GO) annotation [64]. In GO, genes are assigned to three structured, controlled vocabularies, called ontologies: biological process, cellular components and molecular functions. The GO Consortium (GOC) [64] [65] is involved in the development and application of the GO.

In what follows, we briefly report some R tools related to GOC [66, 67]:

• **AnnotationDbi**: It provides user interface and database connection code for annotation data packages using SQLite data storage.

• **FunCluster**: It is a functional profiling and analysis of microarray expression data based on GO & KEGG.

• **GExMap**: It is an intuitive visual tool to perform a GO and to test to unveil genomic clusters, graphical interpretations and statistical results in pdf files.

• **GO.db annotation**: It provides detailed information about the latest version of the GOs and it is updated biannually.

• **GOsummaries**: It shows GO enrichment results in the context of experimental data.

• **GOstats**: It determines which GOs found in gene lists are statistically over/under-represented.

• **goTools**: It compares the GOs represented by the genes in the three gene lists (biological process, molecular function and cellular component).

• **topGO**: It provides tools for testing GO terms while accounting for the topology of the GO graph. Different test statistics and different methods for eliminating local similarities and dependencies between GO terms can be implemented and applied.

### 6. TOOLS

For clustering, we introduce some clustering webtools.

• **WLPT@DNA-Array**: It is a webtool of management and analysis of DNA microarrays by using weighted trees. It computes the appearance probability of a DNA microarray, to compare the
informational distances in the expression of genes between DNA microarrays, and determines the group of candidate genes related to a pathology. WLPT®DNA-Array is available at http://www.genopole-lille.fr/spip.

• Lattice Miner (LM) is a formal concept analysis webtool for the construction, visualization and manipulation of concept lattices. It allows the generation of formal concepts and association rules as well as the transformation of formal contexts via apposition, subposition, reduction and object/attribute generalization, and the manipulation of concept lattices via approximation, projection and selection. LM allows also the drawing of nested line diagrams. LM is available at http://sourceforge.net/projects/lattice-miner/

• Formal concept analysis based Association rule Miner (FAM) was designed and implemented considering user’s facility of information retrieval such as context editing, concept and lattice exploring, query submitting and showing the association rules in response to the query. FAM is available at http://bike.snu.ac.kr/

• SPECLUST is a webtool for hierarchical clustering of peptide mass spectra obtained from protease-digested proteins. Mass spectra are clustered according to the peptide masses they contain, such that mass spectra containing similar masses are clustered together. Hierarchical clustering of Mass Spectra (MS) with SPECLUST can in particular be useful for MS-screening of large proteomic data sets derived from 2D-gels. SPECLUST can also be used to identify masses shared by mass spectra. Masses present in the majority of the mass spectra in a data set are likely to be contaminant. With SPECLUST, MS/MS can be focused on non-contaminant shared masses in a cluster, facilitating investigations of protein isoforms. Within a cluster, shared and unique masses represent peptides from regions that are similar and different, respectively, between protein isoforms. Taken together, SPECLUST is a versatile tool for analysis of mass spectrometry data. SPECLUST is available at http://bioinfo.thep.lu.se/speclust.html.

• Mixture Modelling (Mixmod) webtool fits mixture models to a given data set with a density estimation, a clustering or a discriminant analysis purpose. A large variety of algorithms to estimate the mixture parameters are proposed (EM, CEM, SEM) and it is possible to combine them to lead to different strategies in order to get a sensible maximum of the likelihood (or complete-data likelihood) function. Mixmod is currently focused on multivariate Gaussian mixtures and fourteen different Gaussian models. It can be considered according to different assumptions on the component presented by the variance matrix eigenvalue decomposition. Moreover, different information criteria for choosing a parsimonious model (the number of mixture components, for instance), some of them favoring either a cluster analysis or a discriminant analysis view point, are included. Written in C++, Mixmod is interfaced with Scilab and Matlab. Mixmod, the statistical documentation and also the user guide are available at http://www.math.univ-lyon1.fr/mixmod/index.php.

There are also many R microarray biclustering tools. Table 1. presents a few examples on tools and here are some examples [68]:

<table>
<thead>
<tr>
<th>Tool</th>
<th>Biclustering algorithms</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lattice</td>
<td>Galois lattice</td>
<td>[17]</td>
</tr>
<tr>
<td>arules</td>
<td>rules</td>
<td>[71]</td>
</tr>
<tr>
<td>rootSolve, pracma</td>
<td>Newton Raphson</td>
<td>[71]</td>
</tr>
<tr>
<td>Package</td>
<td>List of datasets</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>aroma. Copy-number (cn) and aroma. for affymetrix anpuce</td>
<td>Spleen</td>
<td></td>
</tr>
<tr>
<td>Abd</td>
<td>Analysis of Biological Data (abd)</td>
<td></td>
</tr>
<tr>
<td>ICluster</td>
<td>Breast cancer, DNA cn, breast.chr17</td>
<td></td>
</tr>
</tbody>
</table>

### 7. DATASETS

Real data are also used because artificial data can only be used to test the effect of certain aspects such as noise level and overlap degree of the bicluster problems on different models/algorithms. We introduce AML/ALL (acute myeloid leukemia and acute lymphoblastic leukemia), Central Nervous System (CNS), lung cancer and colon cancer datasets. All these datasets can be obtained directly from http://sdmc.lit.org.sg/GEDatasets/

Table 2. Microarray datasets used to evaluate biclustering algorithms
### Table

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORCME</td>
<td>Gene expression</td>
</tr>
<tr>
<td>Adegenet</td>
<td>Genetic and genomic</td>
</tr>
<tr>
<td>SNPMClust</td>
<td>Dose-response microarray</td>
</tr>
<tr>
<td>DCGL</td>
<td>Differential co-expression and regulation analysis</td>
</tr>
<tr>
<td>Opmdata</td>
<td>OmniLog(R) Phenotype Microarray data (opmdata)</td>
</tr>
<tr>
<td>Knorm</td>
<td>Across multiple biologically interrelated experiments</td>
</tr>
<tr>
<td>Biclust</td>
<td>BicatYeast</td>
</tr>
<tr>
<td>DDHFm</td>
<td>Data-Driven Haar-Fisz for Microarrays (DDHFm)</td>
</tr>
<tr>
<td>integrativeMEdata</td>
<td>Categorical clinical factors, cancer microarray</td>
</tr>
<tr>
<td>Madsim</td>
<td>Flexible microarray data simulation model (madsim)</td>
</tr>
<tr>
<td>EMA</td>
<td>Easy Microarray data Analysis (EMA)</td>
</tr>
<tr>
<td>FBN</td>
<td>SNP microarray</td>
</tr>
<tr>
<td>BioConductor</td>
<td>Acute Lymphocytic Leukemia (ALL), arrayMissPattern.</td>
</tr>
<tr>
<td>Bioconductor annotation Data</td>
<td>GO.db, GO.dbconn, GOBANCESTOR, GOBPARENTS, GOBCHILDREN, GOCCANCESTOR, GOCCPARENTS, GOCCCHILDREN, GOCOFFSPRING, GOMAPCOUNTS,</td>
</tr>
<tr>
<td>Lemma</td>
<td>Laplace approximated EM Microarray Analysis (lemma)</td>
</tr>
<tr>
<td>Maanova</td>
<td>N-dye Micro 18-array affymetrix experiment</td>
</tr>
<tr>
<td>GeneARMA</td>
<td>Time-course microarray with periodic gene expression</td>
</tr>
<tr>
<td>iGenomicViewer</td>
<td>IGGVex</td>
</tr>
<tr>
<td>CLAG</td>
<td>Breast tumor cells</td>
</tr>
</tbody>
</table>

- **CNS dataset**: This dataset consists of 34 samples: 10 classic medulloblastomas, 10 malignant, 10 rhabdoids, and 4 normals.
- **Lung cancer dataset**: This dataset is composed of 32 samples which are about Malignant Pleural Mesothelioma (MPM, 16 samples) and ADenoCArcinoma (ADCA, 16 samples) of the lung.
- **Colon cancer dataset**: Murali and Kasif used a colon cancer dataset originated in to test XMOTIF. The matrix contains 40 colon tumor samples and 22 normal colon samples over about 6500 genes. Colon cancer dataset is available at http://www.weizmann.ac.il/physics.
- **FuncAssociate** allow to evaluate the discovered biclusters. FuncAssociate first uses Fisher's exact test to compute the hypergeometric functional score of a gene set, then uses the Westfall and Young procedure to compute the adjusted significant score of the gene set. The analysis is performed on the gene expression data of S. cerevisiae.
For biclustering, there are many microarray datasets, related to R package, used to evaluate biclustering algorithms. Table 2. presents a few examples on these datasets.

7. CONCLUSION

We have briefly reviewed clustering algorithms of microarray data. We have reported advantages and drawbacks of certain algorithms. Although clustering of microarray data has been the subject of a large research, no one of the existing clustering algorithms is perfect and the construction of biologically significant groups of clusters for large microarray data is still a problem. Biological validation of clustering algorithms of microarray data is one of the most important open issues. The biclustering of microarray data has been the subject of a large research. No one of the existing biclustering algorithms is perfect. The construction of biologically significant groups of biclusters for large microarray data is still a problem that requires a continuous work. Biological validation of biclusters of microarray data is one of the most important open issues. So far, there are no general guidelines in the literature on how to validate biologically extracted biclusters.

REFERENCES


Authors

Professor ELLOUMI Mourad : Full Professor in Computer Science Head of the BioInformatics Group (BIG) of The Laboratory of Technologies of Information and Communication, and Electrical Engineering (LaTICE), National High School of Engineers of Tunis (ENSIT), University of Tunis, Tunisia, and Professor at the Faculty of Economic Sciences and Management of Tunis (FSEGT), University of Tunis El Manar, Tunisia.

Mrs BEN SABER Haifa : Phd student on the BioInformatics Group (BIG) of The Laboratory of Technologies of Information and Communication, and Electrical Engineering (LaTICE), National High School of Engineers of Tunis (ENSIT), University of Tunis, Tunisia, and Assistant at the Time Université, Tunisia.