International Journal of Machine Learning and Cybernetics (2018) 9:1879–1893
https://doi.org/10.1007/s13042-018-0794-9

ORIGINAL ARTICLE

A new FCA-based method for identifying biclusters in gene expression data

Amina Houari[1] · **Wassim Ayadi**[2] · **Sadok Ben Yahia**[1]

Received: 13 July 2015 / Accepted: 26 February 2018 / Published online: 7 March 2018

Abstract
Biclustering has been very relevant within the field of gene expression data analysis. In fact, its main thrust stands in its ability to identify groups of genes that behave in the same way under a subset of samples (conditions). However, the pioneering algorithms of the literature has shown some limits in terms of the quality of unveiled biclusters. In this paper, we introduce a new algorithm, called *BiFCA+*, for biclustering microarray data. *BiFCA+* heavily relies on the mathematical background of the formal concept analysis, in order to extract the set of biclusters. In addition, the *Bond* correlation measure is of use to filter out the overlapping biclusters. The extensive experiments, carried out on real-life datasets, shed light on *BiFCA+*'s ability to identify statistically and biologically significant biclusters.

Keywords Biclustering · Formal concept analysis · Data mining · Bioinformatics · DNA microarray data · *Bond* correlation measure

1 Introduction

A biological network is a linked collection of biological entities like genes, proteins, and metabolistes [34]. Analyzing information and extracting biologically relevant knowledge, from these entities, is one of the key issues of bioinformatics. For instance, DNA microarray technologies help to measure the expression levels of thousands of genes under experimental conditions [14]. To do so, gene expression data are arranged in a data matrix. In the latter, rows represent genes, columns represent samples (experimental conditions), and each entry of the matrix denotes the expression level of a gene under a certain experimental condition. In this respect, the discovery of transcriptional modules of genes that are co-regulated in a set of experiments is of paramount importance [14].

Interestingly enough, the clustering technique has been beneficial in many challenges in bioinformatics. In fact, it allows researchers to gather information such as cancer occurrences, specific tumor subtypes and cancer survival rates [67]. However, the use of clustering algorithms has two major drawbacks.

1. They consider the whole set of samples. This is despite the fact that genes may not be relevant to every sample. Instead, they can be relevant to only a subset of samples, which is a fundamental aspect for numerous problems in the biomedicine field [66]. Thus, clustering should be performed simultaneously on both genes and conditions.
2. Each gene can only be clustered into one group. Nevertheless, many genes can belong to several clusters depending on their influence in different biological processes [28].

In this respect, biclustering, which is a particular clustering type, palliates these drawbacks. Hence, biclustering aims to identify maximal sub-matrices (*aka biclusters*) where a subset of genes expresses highly correlated behaviors over a range of conditions [14]. Nevertheless,

✉ Wassim Ayadi
wassim.ayadi@fsegt.utm.tn

Amina Houari
amina.houari@fst.utm.tn

Sadok Ben Yahia
sadok.benyahia@fst.rnu.tn

[1] Faculty of Sciences of Tunis, University of Tunis El Manar, LIPAH-LR11ES14, 2092 Tunis, Tunisia

[2] National Higher Engineering School of Tunis, University of Tunis, LaTICE-LR11ES04, 1008 Tunis, Tunisia

the biclustering task is a highly combinatorial problem and is known to be an NP-hard one [18].

As it can be witnessed in the dedicated literature, the biclustering usage is widespread in the analysis of gene expression data. It was first introduced by the pioneering work of [18]. Subsequently, a lot of other algorithms have been proposed [14–16, 23, 25, 41, 54]. According to [25], the existing biclustering algorithms can be grouped into two main classes: systematic search algorithms and stochastic search algorithms (cf. Sect. 2.2).

Despite the large number of the aforementioned biclustering algorithms, most of them are based on greedy or stochastic approaches. However, they provide sub-higher-quality answers with restrictions on the structure, the coherency and the quality of biclusters [14, 33, 35]. Some attempts to palliate such drawbacks have relied on the pattern-mining approaches [38, 39, 49, 52]. Pattern-mining-based biclustering approaches aim to perform efficient and flexible searches with better solutions in terms of coherency and quality [30]. These advantages will bring these algorithms into the spotlight when it comes to biological data analysis [31, 32, 34, 38, 40, 49].

Among these pattern-mining-based algorithms are those relying on the formal concept analysis (FCA). Biclustering has multiple elements in common with the FCA. In fact, *a bicluster* can be seen as *a formal concept* that reflects the inherent relationship between objects and attributes [38]. Indeed, as for Prelic et al. [60], an *inclusion-maximal* bicluster is the maximal set of objects related to a maximal set of attributes. This definition perfectly matches with that of a *formal concept* in the FCA theory [68]. This close connection motivates the wide use of the FCA's large collection of mathematical results for the biclustering task. Indeed, the FCA is a key method used for the analysis of object-attribute relationships and for knowledge presentations [42, 45].

For these reasons, one might argue that the FCA can be considered as a type of biclustering methods for binary data. Various approaches have been interested in extracting biclusters using the FCA. However, these algorithms have the tendency to focus on one type of biclusters, extract overlapping ones or refrain from biological validation.

In this paper, we introduce a new FCA-based algorithm for biclustering DNA microarray data, called *BiFCA+*. This latter allows observing the profile of each gene through all pairs of conditions by discretizing the original microarray data. Interestingly enough, *BiFCA+* relies on the *Bond* correlation measure [55] to avoid the high overlap between extracted biclusters.

The main contributions of this paper are as follows:

- we propose a new discretization method of the DNA microarray data.
- We design an efficient FCA-based algorithm for extracting correlated genes.
- We filter out the obtained biclusters using the *Bond* correlation measure in order to remove high overlapping biclusters.
- We show the effectiveness of our method through extensive carried-out experiments on three real-life DNA microarray data. Indeed, we extract statistically and biologically significant biclusters, highlighting competitive results versus other popular biclustering algorithms.

The remainder of this paper is organized as follows: In the next section, we provide a background on the target task and we review some related work. Section 3 is dedicated to the description of the algorithm. In Sect. 4, we provide the results of the application of our algorithm on real-life microarray datasets. Finally, Sect. 5 concludes this paper and sketches issues of future work.

2 Background

In this section, we provide the basics on the biclustering field and we review the dedicated related work.

2.1 Biclustering: basic notions

In the following, we recall some basic definitions borrowed from the biclustering field.

Definition 1 (*Bicluster*) A bicluster is a subset of objects (genes) associated with a subset of attributes (conditions) in which these rows are co-expressed.

The bicluster associated with the matrix $M = (I, J)$ is a couple (A, B), such that $A \subseteq I$ and $B \subseteq J$, where (A, B) is maximal; *i.e.* there does not exist a bicluster (C, D) with $A \subseteq C$ or $B \subseteq D$.

This leads to the definition of biclustering.

Definition 2 (*Biclustering*) The biclustering problem focuses on the identification of the best biclusters of a given dataset. The best bicluster must fulfill a number of specific homogeneity and significance criteria (guaranteed through the use of a function to guide the search) [56].

In the following, we present the different types of biclusters.

Definition 3 (*Types of biclusters*) According to [14], a bicluster can be one of the following types (cf. Fig. 1) :

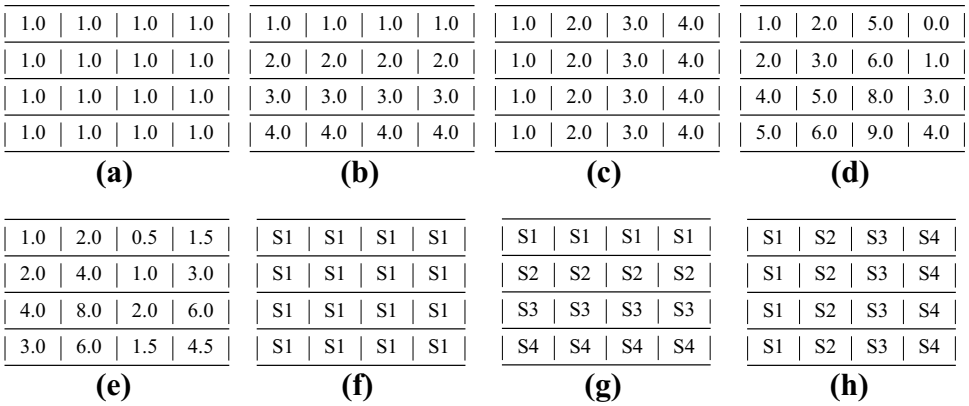

Fig. 1 Examples of different types of biclusters. **a** Constant bicluster, **b** constant rows, **c** constant columns, **d** coherent values (additive model), **e** coherent values (multiplicative model), **f** overall coherent evolution, **g** coherent evolution on rows, **h** coherent evolution on columns [14]

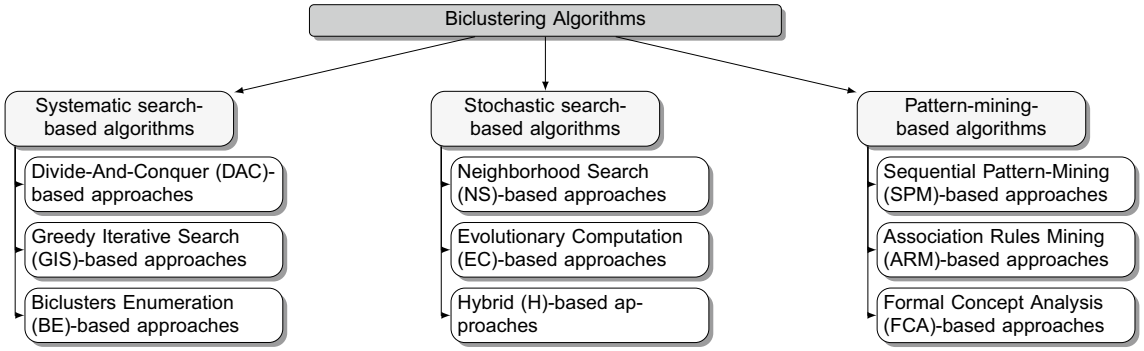

Fig. 2 Structured view on existing biclustering algorithms

– *Bicluster with constant values* It is a bicluster where all values are equal.

$$m_{ij} = c \tag{1}$$

– *Bicluster with constant values on rows and columns* There are two types of biclusters with constant values:

1. Constant values on rows:

$$m_{ij} = c + a_i \tag{2}$$

$$m_{ij} = c * a_i \tag{3}$$

2. Constant values on columns:

$$m_{ij} = c + b_j \tag{4}$$

$$m_{ij} = c * b_j \tag{5}$$

– *Bicluster with coherent values* There are two types of biclusters with coherent values. Those with an additive model and those with a multiplicative model defined respectively by:

1. Additive model:

$$m_{ij} = c + a_i + b_j \tag{6}$$

2. Multiplicative model:

$$m_{ij} = c * a_i * b_j \tag{7}$$

– *Bicluster with coherent evolutions* It is a bicluster where the coherent evolutions are observed across the rows and/or columns of the data matrix.

In the following, we scrutinize the pioneering work that has addressed the extraction of biclusters from gene expression data.

2.2 Related work

The costly computation complexity of extracting maximal sub-matrices of genes and conditions such that the genes express highly correlated behaviors over a range of conditions has been a main impediment to the wide-scale use of gene expression analysis community. A recent review of various biclustering algorithms for gene expression data is provided in [25], where existing biclustering algorithms were grouped into two main streams to which we add the

third stream. At a glance, as depicted in Fig. 2, the dedicated literature witnessed three main streams for addressing the biclustering task. These streams are detailed in the following.

(i) The systematic search-based stream includes the following approaches:

1. *The divide-and-conquer (DAC)-based approach* Generally, this approach repeatedly splits the problem into smaller ones with similar structures to the original problem, until these sub-problems become smaller enough to be straightforwardly solved. The solutions to the sub-problems are then combined to create a solution to the original problem respectively [25]. The algorithms adopting this approach were given in [60, 63].
2. *The Greedy Iterative Search (GIS)-based approach* In this approach, a solution is constructed in a step-by-step way using a given quality criterion. The decisions made at each step are based on information at hand without worrying about the impact of these decisions in the future. Moreover, once a decision is made, it will become irreversible and will never be reconsidered [25]. The algorithms adopting this approach were given in [9, 17, 69].
3. *The Biclusters Enumeration (BE)-based approach* As indicated by its name, an enumeration algorithm enumerates all the solutions for the original problem. The enumeration process is generally represented by a search tree [25]. The algorithms adopting this approach were given in [4, 7, 37, 61].

(ii) The stochastic search-based stream includes the following approaches:

1. *The Neighborhood Search (NS)-based approach* It starts with an initial solution and then moves iteratively to a neighboring solution thanks to the neighborhood exploitation strategy. The algorithms adopting this approach were given in [5, 20].
2. *The Evolutionary Computation (EC)-based approach* This approach is based on the natural evolutionary process such as population, reproduction, mutation, recombination, and selection. The algorithms adopting this approach were given in [21, 22].
3. *The Hybrid (H)-based approach* The latter tries to combine the neighborhood search and the evolutionary approaches. The algorithms adopting this approach were given in [26, 50].

(iii) The pattern-mining-based stream includes:

1. *Sequential Pattern-Mining (SPM)-based approaches* SPM is used in order to extract order-preserving biclusters. A bicluster is order-preserving if there is a permutation of its columns under which the sequence of values in every row increases. In this context, SPM is applied; and the biclusters are extracted from the frequent sequences as well as their supporting transactions. The algorithms adopting this approach were given in [29, 32].
2. *Association Rules Mining (ARM)-based approaches* ARM can be used to compose biclusters. To perform this task, they divide the problem into two sub-problems:

   (1) Finding all association rules that represent biclusters' samples/genes. In fact, they consider items of both the premise and conclusion of an association rule.

   (2) Extracting the supporting transactions of these items. The authors in [51] provided a review of various biological applications of association rule mining.
3. *Formal Concept Analysis (FCA)-based approaches* The FCA can be viewed as a kind of biclustering for binary data. It provides pattern (*bicluster*) extraction from a binary relation, namely, a *formal concept*. In its gene expression data applications, the concept's extent represents the maximal sets of genes related to a maximal set of samples (concept's intent). The algorithms adopting this approach were given in [38, 39].

In this work, we are particularly interested in the FCA-based biclustering algorithms. In this respect, several approaches have been interested in extracting biclusters using the FCA. In [52], the authors proposed a new approach, called FIST, for extracting the bases of extended association rules and conceptual biclusters, using frequent closed itemsets [57]. Nevertheless, they failed to detail their discretization method, and treated the matrix as though it had been already binary. This was done despite the fact that microarray data were not initially coded in a binary format.

Furthermore, their approach did not entail any biological validation of the extracted biclusters.

Whereas, Pensa et al. [59] relied on a single threshold, where expression values greater than this threshold were represented by 1, otherwise by 0. Most discretization techniques commonly applied to gene expression data use absolute expression values. However, the main drawback of this technique is how to find the best method to set the threshold value.

Kaytoue et al. [39] utilized the scaling of numerical data and considered that formal concepts were the groups of genes whose expression values were in the same intervals for a subset of conditions.

Added to that, Kaytoue el al. [38] referred to the algorithm presented in [39], using the triadic concept analysis [43, 64] in order to extract biclusters with similar values. Both of the latter approaches only paid attention to the extraction of one type of biclusters, i.e. biclusters with similar values. In addition, they did not offer any biological validation for the obtained biclusters.

The above mentioned biclustering algorithms have the tendency to focus on one type of biclusters, extract overlapping ones or refrain from biological validation. Thus, in the remainder, we introduce a new FCA-based approach for the extraction of biclusters from gene expression data.

3 *BiFCA+*: the proposed biclustering algorithm

In this section, we first recall some basic definitions borrowed from the FCA field. Second, we provide a detailed description of the proposed algorithm, followed by an illustrative example.

3.1 FCA basic settings

We start this subsection by presenting the notion of a formal context.

Definition 4 (*Formal context*) A formal context is a triplet $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$, where $\mathcal{O}$ represents a finite set of objects, $\mathcal{I}$ is a finite set of items (or attributes) and $\mathcal{R}$ is a binary (incidence) relation (i.e., $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{I}$). Each couple $(o, i) \in \mathcal{R}$ expresses that the object $o \in \mathcal{O}$ contains the item $i \in \mathcal{I}$. Table 1 sketches an example of a formal context.

Table 1 Example of a formal context

	I_1	I_2	I_3	I_4	I_5
1	×		×	×	
2		×	×		×
3	×	×	×		×
4		×			×
5	×	×	×		×

It is worth mentioning that the link between the power-sets $\mathcal{P}(\mathcal{I})$ and $\mathcal{P}(\mathcal{O})$, associated respectively to the set of items $\mathcal{I}$ and the set of objects $\mathcal{O}$, is defined as follows:

Definition 5 (*Galois connection*) Let $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$ be a formal context. The application ψ is defined from the power-set of objects [i.e., $\mathcal{P}(\mathcal{O})$] to the power-set of items [i.e., $\mathcal{P}(\mathcal{I})$]. It associates to a set of objects O the set of items $i \in \mathcal{I}$ that are common to all objects $o \in O$:

$$\psi : \mathcal{P}(\mathcal{O}) \rightarrow \mathcal{P}(\mathcal{I})$$
$$O \mapsto \psi(O) = \{i \in \mathcal{I} | \forall o \in O, (o, i) \in \mathcal{R}\}$$

In a dual way, the application ϕ is defined from the power-set of items [i.e., $\mathcal{P}(\mathcal{I})$] to the power-set of objects [i.e., $\mathcal{P}(\mathcal{O})$]. It associates to a set of items I the set of objects $o \in \mathcal{O}$ that contains all items $i \in I$:

$$\phi : \mathcal{P}(\mathcal{I}) \rightarrow \mathcal{P}(\mathcal{O})$$
$$I \mapsto \phi(I) = \{o \in \mathcal{O} | \forall i \in I, (o, i) \in \mathcal{R}\}$$

The coupled applications (ψ, ϕ) form a Galois connection between the power-set of $\mathcal{O}$ and that of $\mathcal{I}$ [8, 27].

The following definition introduces the closure operators associated to the Galois connection.

Definition 6 (**Galois closure operators**) Let us consider the power-sets $\mathcal{P}(\mathcal{I})$ and $\mathcal{P}(\mathcal{O})$, with the inclusion relation $\subseteq$, i.e. the partially ordered sets $(\mathcal{P}(\mathcal{I}), \subseteq)$ and $(\mathcal{P}(\mathcal{O}), \subseteq)$. The operators $\gamma = \phi o \psi$ from $(\mathcal{P}(\mathcal{I}), \subseteq)$ to $(\mathcal{P}(\mathcal{I}), \subseteq)$ and $\omega = \psi o \phi$ from $(\mathcal{P}(\mathcal{O}), \subseteq)$ to $(\mathcal{P}(\mathcal{O}), \subseteq)$ are closure operators of the Galois connection [8, 27]. They define closure systems on $(\mathcal{P}(\mathcal{I}), \subseteq)$ and $(\mathcal{P}(\mathcal{O}), \subseteq)$, respectively. The operator γ generates closed subsets of items, while the operator ω generates closed subsets of objects.

This leads to the definition of a formal concept.

Definition 7 (*Formal concept*) A pair $\langle A, B \rangle \in \mathcal{O} \times \mathcal{I}$ of mutually corresponding subsets, i.e. $A = \psi(B)$ and $B = \phi(A)$, is called a *formal concept*, where A is called *extent* and B is called *intent*.

In its gene expression data application, we consider a *formal concept* as a *bicluster*, where the concept's *extent* represents *genes* while the concept's *intent* represents the *experimental conditions*.

Example 1 Let us consider the formal context given by Table 1. We have: $\mathcal{O} = \{1, 2, 3, 4, 5\}$ and $\mathcal{I} = \{I_1, I_2, I_3, I_4, I_5\}$. From this formal context, we can extract $\langle 135, I_1I_3\rangle$[1] as a *formal concept*.

In the following, we define the support of an itemset.

Definition 8 (*Support of an itemset*) Let $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$ be a formal context. We distinguish two kinds of support that can be associated to a non-empty itemset I:

- *Conjunctive support:* $Supp(\wedge I) = |\{o \in \mathcal{O} | (\forall i \in I, (o, i) \in \mathcal{R})\}|$. $Supp(\wedge I)$, seen as a conjunction of items (i.e. $i_1 \wedge i_2 \wedge \cdots \wedge i_n$), is the number of objects containing all items of I.
- *Disjunctive support:* $Supp(\vee I) = |\{o \in \mathcal{O} | (\exists i \in I, (o, i) \in \mathcal{R})\}|$. $Supp(\vee I)$, seen as a disjunction of items (i.e. $i_1 \vee i_2 \vee \cdots \vee i_n$), is the number of transactions containing at least one item of I.

This leads us to present the *Bond* correlation measure.

Definition 9 (*Bond correlation measure*) The *Bond* correlation measure [55] computes the ratio between the conjunctive support and the disjunctive one. Thus, the *Bond* correlation measure of two concept's intents B_1 and B_2 is defined as follows:

$$Bond(B_1, B_2) = \frac{Supp(\wedge B_1, B_2)}{Supp(\vee B_1, B_2)} \tag{8}$$

Consequently, we can redefine the *Bond* correlation measure as follows:

$$Bond(B_1, B_2) = \frac{|B_1 \bigcap B_2|}{|B_1 \bigcup B_2|} \tag{9}$$

In the remainder of this section, we thoroughly describe our proposed method.

3.2 *BiFCA+* algorithm

The *BiFCA+* biclustering algorithm is an FCA-based algorithm that identifies biclusters from gene expression data. As illustrated by Fig. 3, it operates in three main phases. The first one is the *discretization phase*. Starting from a numerical dataset, the basic idea is to build a formal context where genes stand for objects and conditions for the attributes.

[1] We use a separator-free abbreviated form for the sets; e.g., $\{I_1I_2I_3\}$ stands for the set of items $\{I_1, I_2, I_3\}$.

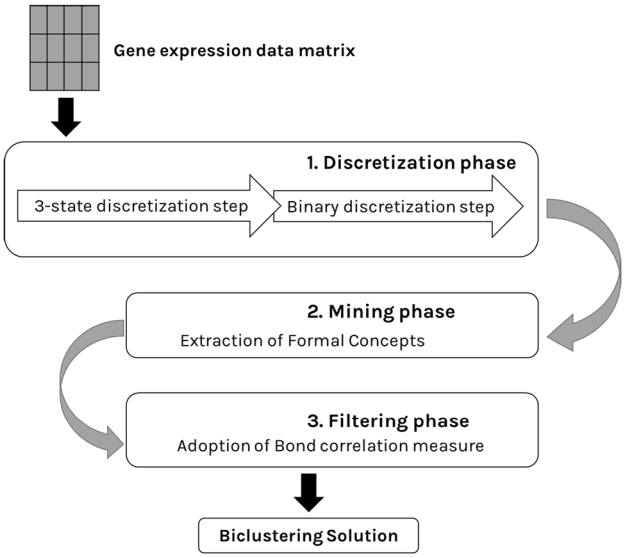

Fig. 3 *BiFCA+*'s at a glance

Subsequently, it starts the *mining phase*. The latter allows extracting formal concepts that represent the correlated biclusters. Finally, we have to perform the *filtering phase*. This latter is performed in order to remove the biclusters that have a high overlap. Given the biclusters obtained from the previous phase, we compute the similarity measure between each pair of biclusters. The latter is defined as the ratio between the conjunctive support of two biclusters and their disjunctive support. We only retain the biclusters having the *Bond* correlation measure that does not exceed a given threshold *minBond*. The pseudo-code of *BiFCA+* is shown in Algorithm 1. *BiFCA+* takes as an input a data matrix M_1 and a minimum correlation threshold *minBond*. *BiFCA+* enables the determination, from the data matrix M_1, of the set of the obtained biclusters β.

The phases of *BiFCA+* are thoroughly described in the following subsections.

Algorithm 1: BiFCA+

Data: Gene expression data matrix M_1, *minBond*.
Result: Set of biclusters β.

```
begin
    β := ∅ ;
    /* First step */
    Discretize M1 using Equation 10 to obtain M2 ; // 3-state data matrix.
    Discretize M2 using Equation 11 to obtain M3 ; // Binary data matrix.
    /* Second step */
    Extract set of formal concepts FCs; //
    /* Third step */
    for each two biclusters FCi = ⟨Ai, Bi⟩ and FCj = ⟨Aj, Bj⟩ do
        if Bond(Bi,Bj) > minBond then
            β = β ⋃ {FCi or FCj}; // Bicluster with highest number of samples.
        else
            β = β ⋃ {FCi and FCj}; // Consider FCi and FCj as biclusters.
    return β;
```

3.2.1 Pre-processing of gene expression data matrix

Our method applies a pre-processing phase to transform the original data matrix M_1 into a binary one. This phase is split into two steps:

1. First, we discretize the original data into a 3-state data matrix M_2. This step aims to unveil the trajectory patterns of genes. According to [48, 58], in the DNA microarray data analysis, we add genes into a bicluster whenever their trajectory patterns of expression levels are similar across a set of samples.

   Interestingly enough, our proposed discretization phase keeps track of the profile shape[2] over conditions and preserves the similarity information of the trajectory patterns of the expression levels.

   Before delving through the mining process, we must at first discretize the initial data matrix. The discretization process outputs the 3-state data matrix. It consists in combining in pairs, for each gene, all the adjacent conditions. Indeed, the 3-state data matrix gives an idea about the profile. Furthermore, it gives a global view of the profile of all conditions.

   In our case, each column of the 3-state data matrix carries the meaning of the variation of genes between a pair of M_1 conditions. It offers useful information for the identification of biclusters, i.e. up (1), down (-1) and no change (0).

   Formally, the matrix M_2 (3-state data matrix) is defined as follows :

$$M_2 = \begin{cases} 1 & \text{if } x_1 < x_2 \\ -1 & \text{if } x_1 > x_2 \\ 0 & \text{if } x_1 = x_2 \end{cases} \tag{10}$$

   with $x_1 = M_1[j, l]$; $x_2 = M_1[j, l+1]$; and $j \in [1 \dots n]$; $l \in [1 \dots m-1]$

2. For the second step of the pre-processing phase, we build the binary data matrix in order to extract formal concepts. In this respect, we compute the average number of repetitions for each column in the matrix M_2 (3-state data matrix). In other words, we have:

   (a) $|maxrepeat|$: This variable stands for the maximum number of occurrences by column.
   (b) $|minrepeat|$: This variable stands for minimum number of occurrences by column.
   (c) $|mediumrepeat|$: It stands for the medium number of occurrences by column.

[2] This may be either monotone increasing, monotone decreasing, up–down or down–up, etc.

Table 2 Example of gene expression data matrix (M_1)

	$c1$	$c2$	$c3$	$c4$	$c5$	$c6$
$g1$	10	20	5	15	0	18
$g2$	20	30	15	25	26	25
$g3$	23	12	8	15	20	50
$g4$	30	40	25	35	35	15
$g5$	13	13	18	25	30	55
$g6$	20	20	15	8	12	23

It is better to choose the mean value since the maximum will produce a huge number of high overlapping biclusters, whereas the minimum value generates biologically none-valid biclusters.

Formally, we define the binary matrix as follows:

$$M_3 = \begin{cases} 1 & \text{if } M_2[j, l] = average\ value \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where $j \in [1 \dots n]$ and $l \in [1 \dots m-1]$

After the discretization step, the dimensions of our data matrix (M_2) become equal to $n * (m-1)$.

Example 2 Let us consider the data matrix M_1 given by Table 2. For the first row, we have $M_{1_{1j}} = (10, 20, 5, 15, 0, 18)$ with $j \in [1 \dots 6]$. In the first step of the pre-processing phase we obtain the discretized first row; i.e. $M_{2_{1j}} = (1, -1, 1, -1, 1)$ with $j \in [1 \dots 5]$. In the second step, the first column becomes $M_{3_{i1}} = (0, 0, 0, 0, 1, 1)$ with $i \in [1 \dots 6]$.

3.2.2 Extracting formal concepts (biclusters)

The FCA can be viewed as a kind of biclustering for binary data. It provides patterns (*biclusters*) extraction from a binary context.

In this respect, after preparing the binary data matrix, we move to extract formal concepts (biclusters) from the binary matrix M_3.

The extraction of the formal concepts is carried out through the invocation of a slightly modified version of the efficient LCM algorithm [65]. The choice of this algorithm is argued by the fact that it has a linear complexity in the number of closed attributes and has been shown to be one of the best algorithms dedicated to such a task.

3.2.3 Computation of similarity measure (*Bond*)

The *BiFCA+* algorithm is already able to identify overlapping biclusters. In fact, for the filtering process, we only consider biclusters having a low overlap. Indeed, in the case of biclusters that have a high overlap, they have the same biological signification. The *Bond* correlation measure

achieves its minimum of 0 when the biclusters do not overlap at all and its maximum value 1 whenever they are identical.

In order to compute the similarity between the two biclusters (i.e. formal concepts) FC_1 and FC_2, with $FC_1 = \langle A_1, B_1 \rangle$ and $FC_2 = \langle A_2, B_2 \rangle$, where $A_i, i = 1, 2$, represents the extent, and B_i represents the intent, we use the *Bond* correlation measure. The latter assesses the overlap between two concept's intents (cf. Definition 9).

Finally, we only retain the obtained biclusters for which the *Bond* correlation measure does not exceed a given threshold. The set of such biclusters represents a solution to our problem.

In the following, we provide an illustrative example of the *BiFCA+* approach.

3.3 Illustrative example

Let us consider the data matrix given by Table 2. Each column represents all the gene expression levels from a single experiment, and each row represents the expression of a gene across all experiments.

3.3.1 Pre-processing phase of data matrix

The pre-processing phase goes as follows:

1. First, we transform the numerical data into the 3-state data matrix. This is done using Eq. 10. Table 3 provides the obtained results.
2. Second, we create the binary matrix, using the 3-state data matrix. Let us consider the 3-state data matrix given by Table 3. For the sake of building the binary data matrix, we compute the average number of repetitions for each column in the matrix M_2; e.g., for the column $\grave{C}1$ we have:
   (a) $|maxrepeat|$: It is equal to 3 and corresponds to the value 1.
   (b) $|minrepeat|$: It is equal to 1 and corresponds to the value -1.
   (c) $|mediumrepeat|$: It is equal to 2 and corresponds to the value 0. Therefore, the *average value* is 0.

   Subsequently, and using Eq. 11, we obtain the binary matrix sketched by Table 4.

3.3.2 Formal concept extraction phase

After preparing the binary data matrix, we move to extract formal concepts, i.e. the candidate biclusters, from the matrix M_3 (using the LCM algorithm [65]).

Table 3 3-state data matrix (M_2)

	$\grave{C}1$	$\grave{C}2$	$\grave{C}3$	$\grave{C}4$	$\grave{C}5$
$g1$	1	−1	1	−1	1
$g2$	1	−1	1	1	−1
$g3$	−1	−1	1	1	1
$g4$	1	−1	1	0	−1
$g5$	0	1	1	1	1
$g6$	0	−1	−1	1	1

Table 4 Binary data matrix (M_3)

	$\grave{C}1$	$\grave{C}2$	$\grave{C}3$	$\grave{C}4$	$\grave{C}5$
$g1$	0	0	0	1	0
$g2$	0	0	0	0	1
$g3$	0	0	0	0	0
$g4$	0	0	0	0	1
$g5$	1	1	0	0	0
$g6$	1	0	1	0	0

By using the binary data matrix given in Table 4, we obtain as a result the formal concepts shown in Table 5.

3.3.3 Filtering phase

In this phase, we only retain biclusters having a low overlap. This overlap is assessed through the *Bond* correlation measure. For example, with respect to the formal concepts given in Table 5, if we consider FC_3 and FC_4, we compute the *Bond* correlation measure:

$$Bond\,(B_3, B_4) = \frac{|\{\grave{C}1\grave{C}2\} \bigcap \{\grave{C}1\grave{C}3\}|}{|\{\grave{C}1\grave{C}2\} \bigcup \{\grave{C}1\grave{C}3\}|}$$

$$Bond\,(B_3, B_4) = \frac{1}{3} = 0.33$$

The *Bond* correlation measure threshold is equal to 0.5. Thus, we consider the formal concepts FC_3 and FC_4 as non overlapping biclusters. Nevertheless, by lowering the threshold value to 0.3, we only consider one bicluster, that is the one having the highest number of conditions.

4 Experimental results

In this section, we show the results of applying the *BiFCA+* approach on three well-known real-life datasets. The evaluation of biclustering algorithms and the comparison are based on two criteria: statistical and biological. We compare the results obtained by our algorithm versus the state-of-the-art

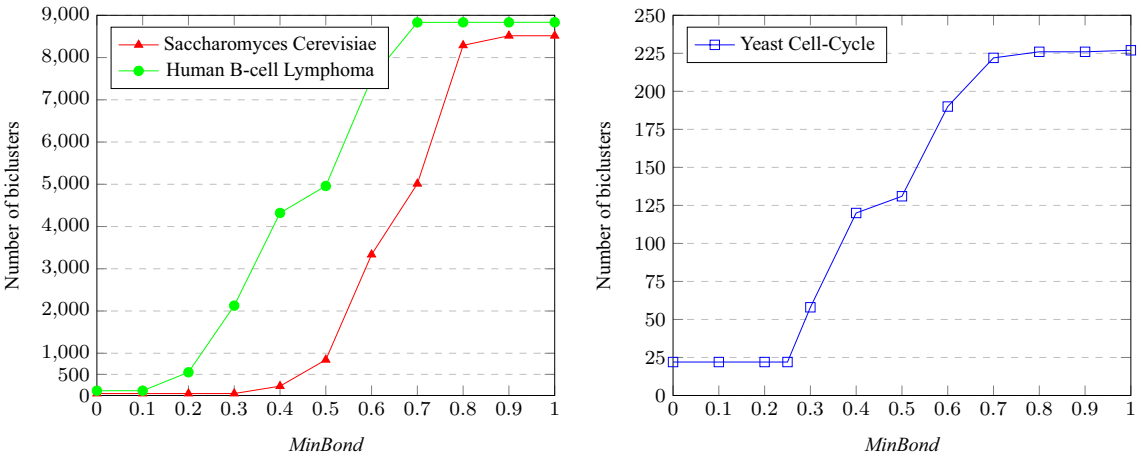

Fig. 4 *BiFCA+* biclusters number w.r.t. *minBond* variations

Table 5 Formal concepts extracted from binary context

FCs	Extent (genes)	Intent (conditions)
FC_1	$g5g6$	$\grave{C}1$
FC_2	$g2g4$	$\grave{C}5$
FC_3	$g5$	$\grave{C}1\grave{C}2$
FC_4	$g6$	$\grave{C}1\grave{C}3$
FC_5	$g1$	$\grave{C}4$

biclustering algorithms as well as the Trimax algorithm[3] [38], which also relies on the FCA. The experiments are carried out on the different datasets. According to the obtained experimental results, interesting reductions of the number of biclusters are obtained as far as the value of *minBond* is lowered. Representative results are plot by Fig. 4, where the *minBond* is set when there is a significant decrease in the number of obtained biclusters. For instance, in the yeast cell-cycle dataset, the *minBond* is set to 0.25.

4.1 Description of used datasets

In order to assess the performance of our proposed algorithm and analyze its results, we carry out a series of experimentations on the following real-life gene expression datasets:

- *Yeast cell-cycle dataset* The yeast cell-cycle[4] is a very popular dataset in the gene expression analysis community. In fact, it is one of the most studied organisms, and the functions of each gene are well known. We use the *Yeast Cell-Cycle dataset* described in [62], processed in [18] and publicly available from [19]. It contains 2884 genes and 17 samples. In the experimentations conducted on this dataset, *minBond* is experimentally set to 0.25.
- *Saccharomyces cerevisiae dataset* The Saccharomyces cerevisiae dataset[5] contains the expression levels of 2993 genes under 173 samples. We experimentally set the *minBond* to 0.3 for our experiments on this dataset.
- *Human B-cell Lymphoma dataset:* The Human B-cell Lymphoma dataset [1] contains the expression levels of 4026 genes under 96 samples.[6] In the experimentations we conduct on this dataset, *minBond* is fixed experimentally to 0.1.

4.2 Description of considered tests

In the following, we describe respectively the statistical and biological criteria.

4.2.1 Statistical criterion

To evaluate the statistical relevance of our algorithm, we heavily rely on the following criteria.

- *Coverage* [12, 46, 50] It represents the total number of cells in a microarray data matrix covered by the obtained biclusters. In the biclustering domain, validation using coverage is considered interesting since a large coverage of a dataset is very important in several applications that rely on biclusters [25]. In fact, the higher the num-

[3] Available at https://github.com/mehdi-kaytoue/trimax.

[4] Available at http://arep.med.harvard.edu/biclustering/.

[5] Available at http://www.tik.ethz.ch/sop/bimax/.

[6] Available at http://arep.med.harvard.edu/biclustering/.

ber of highlighted correlations, the greater the amount of extracted information. Consequently, the higher the coverage, the lower the overlapping in the biclusters.

In the literature, this test has been applied respectively on the Yeast Cell-Cycle and human B-cell lymphoma datasets.[7]

– *p* value We compute the percentage of biclusters having an adjusted *p* value, i.e. the proportion between the number of biclusters having an adjusted *p* value and the total number of obtained bicluters. We compute the adjusted *p* value [60], i.e. based on the exact value of Fisher test [24] to measure the quality of the obtained biclusters. In fact, the biclusters having a *p* value lower than 5% are considered as over-represented; in other words, the majority of genes of a bicluster have common biological characteristics. The best biclusters have an adjusted *p* value less than 0.001. This measure is computed thanks to the web tool ***FuncAssociate***[8][11]. This test is applied respectively on the Yeast Cell-Cycle and *Saccharomyces cerevisiae* datasets.

4.2.2 Biological criterion

The Gene Ontology (GO) project[9] is a collaborative effort to address the need for consistent descriptions of gene products in different databases. The project began as a collaboration between three model organism databases, among them the Saccharomyces Genome Database (SGD). This latter concerns our datasets (Yeast Cell-Cycle and *Saccharomyces cerevisiae*). The GO project provides controlled vocabulary of defined terms representing gene product properties. This covers three domains: (1) biological process, (2) molecular function and (3) cellular component.

In order to evaluate our biclusters biologically we make use of the ***GoTermFinder*** web tool[10] [13]. It searches for significant shared GO terms, used to describe the genes in a given list to help discovring what the genes may have in common. In fact, the biological criterion enables measuring the quality of the resulting biclusters, by checking whether the genes of a bicluster have common biological characteristics.

This test is applied respectively on the Yeast Cell-Cycle and *Saccharomyces cerevisiae* datasets.

[7] The human B-cell lymphoma dataset version that we have does not contain the names of genes to perform other tests.

[8] Available at http://llama.mshri.on.ca/funcassociate/

[9] http://geneontology.org/

[10] Available at http://db.yeastgenome.org/cgi-bin/GO/goTermFinder

Table 6 Human B-cell lymphoma coverage for different algorithms

Human B-cell lymphoma			
Algorithms	Total coverage (%)	Gene coverage (%)	Condition coverage (%)
BiMine [4]	8.93	26.15	100
BiMine+ [7]	21.19	46.26	100
BicFinder [6]	44.24	55.89	100
MOPSOB [47]	36.90	–	–
MOEA [50]	20.96	–	–
SEBI [22]	34.07	38.23	100
CC [18]	36.81	91.58	100
Trimax [38]	8.50	46.32	11.46
BiFCA+	**67.84**	**100**	**100**

Bold values stand for the best results

4.3 Study of statistical relevance

To evaluate our algorithm on real-life datasets, the following criteria are of use:

4.3.1 Coverage criterion

As in [6, 12, 46, 47], we use the coverage criterion defined as the total number of cells in a microarray data matrix covered by the obtained biclusters.

We compare the results of our algorithm versus those obtained by Trimax [38] and those reported by [3]. In the latter reference, the following algorithms were considered: CC [18], BiMine [4], BiMine+ [7], BicFinder [6], MOPSOB [47], MOEA [50] and SEBI [22].

Table 6 (resp. Table 7) presents the coverage of the obtained biclusters. At a glance, we remark that most of the algorithms have relatively close results. For the Human B-cell Lymphoma (respectively the Yeast Cell-Cycle) dataset, the biclusters extracted by our algorithm cover 100% (respectively 80.12%) of the genes, 100% (respectively 100%) of the conditions and 67.84% (respectively 57.07%) of the cells of the expression data matrix. Trimax is largely outperformed, since it only covers respectively 8.50 % of cells, 46.32 % of genes and 11.46 % of conditions for the Human B-cell Lymphoma. It is also worth mentioning that for Yeast Cell-Cycle, the CC algorithm obtains the best results since it masks groups that are extracted with random values. Thus, it prohibits the genes/ conditions that were previously discovered from being selected during the next search process. This type of mask leads to a high coverage. Furthermore, The Yeast dataset only contains positive integer values. Consequently, one can use the Mean Squared Residue (MSR) [18] to extract large biclusters. By contrast, the Human B-cell Lymphoma dataset contains integer values

Table 7 Yeast cell-cycle coverage for different algorithms

Yeast cell-cycle			
Algorithms	Total coverage (%)	Gene coverage (%)	Condition coverage (%)
BiMine [4]	13.36	32.84	100
BiMine+ [7]	51.76	68.65	100
BicFinder [6]	55.43	76.93	100
MOPSOB [47]	52.40	_	_
MOEA [50]	51.34	_	_
SEBI [22]	38.14	43.55	100
CC [18]	**81.47**	**97.12**	100
Trimax [38]	15.32	22.09	70.59
BiFCA+	57.07	80.12	**100**

Bold values stand for the best results

including negative ones. This means that the application of the MSR on this dataset does not result in the extraction of large biclusters.

This implies that our algorithm can generate biclusters with a high coverage of a data matrix. This outstanding coverage is due to the discretization phase as well as the extraction of biclusters without focusing on a specific type of biclusters.

4.3.2 P value criterion

To assess the quality of the extracted biclusters, we use the web tool ***FuncAssociate*** [11] in order to compute the adjusted significance scores for each bicluster (adjusted p value[11]). In fact, the best biclusters have an adjusted p value less than 0.001%. The results of our algorithm are compared versus those obtained by Trimax [38] as well as those concerning CC [18], ISA [10], OSPM [9] and Bimax [60], reported by [3].

The obtained results of the *Saccharomyces cerevisiae* and the Yeast Cell Cycle datasets for different adjusted p values (p = 5%; 1%; 0.5%; 0.1%; 0.001%), for each algorithm over the percentage of total biclusters, are respectively depicted in Figs. 5 and 6. For the *Saccharomyces cerevisiae* dataset (Fig. 5), the *BiFCA+* and Trimax results show that 100% of extracted biclusters are statistically significant with the adjusted p value equal to 0.001%. It is important to note that Bimax achieves its best results whenever $p < 0.1\%$, while CC, ISA and OSPM have a reasonable performance with $p < 0.5\%$.

Whereas, for the Yeast Cell Cycle (Fig. 6), 100% of the extracted biclusters by *BiFCA+* are statistically significant

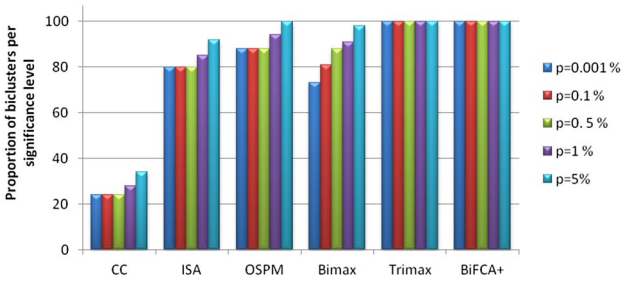

Fig. 5 Proportions of biclusters significantly enriched by GO annotations (*Saccharomyces cerevisiae* dataset)

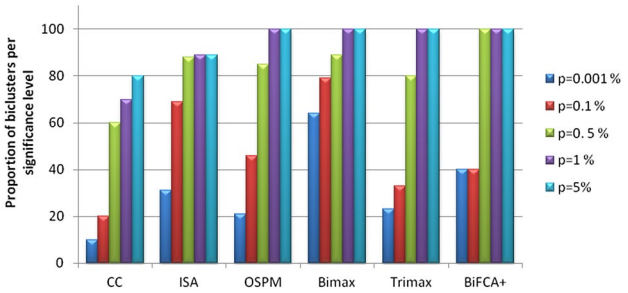

Fig. 6 Proportions of biclusters significantly enriched by GO annotations (Yeast Cell-Cycle dataset)

when $p < 0.5\%$, while only 80% of extracted biclusters by Trimax are statistically significant for the same p value. By contrast, Trimax achieves 100% of extracted biclusters when $p < 1\%$. Our results, then, sharply outperform those of Trimax; however, Bimax scored better when $p < 0.001\%$ and $p < 0.1\%$.

4.4 Biological results

The biological criterion allows measuring the quality of resulting biclusters, by checking whether the genes of a bicluster have common biological characteristics.

To evaluate the quality of the extracted biclusters and identify their biological annotations, we use ***GOTermFinder***, which is designed to search for the significant shared *Gene Ontology* (GO) terms of a group of genes. The GO is organized according to 3 axes: *biological process*, *molecular function* and *cellular component*.[12] We indicate in Tables 8 and 9 the biological annotations of two randomly selected biclusters in terms of the above cited axis, where we report the most significant GO terms. For instance, with the first bicluster extracted from the *Saccharomyces cerevisiae* dataset (Table 8), the list of genes is illustrated in Fig. 7. These genes concern the *Gene Ontology* term *"ribonucleoprotein*

[11] The adjusted significance scores assess genes in each bicluster, which indicates how well they match with the different GO categories.

[12] http://geneontology.org/

Table 8 Significant GO terms (process, function, component) for two biclusters, extracted from *Saccharomyces cerevisiae* data using *BiFCA+*

	Bicluster 1	Bicluster 2
Biological process	Ribosome biogenesis (5.8%, 2.17e-61)	Single-organism process (49.3%, 7.64e-43)
	ncRNA processing (5.8%, 8.44e-57)	Single-organism cellular process (43.0%, 4.61e-30)
	Sibonucleoprotein complex biogenesis (6.9%, 2.02e-55)	Single-organism metabolic process (25.5%, 7.13e-25)
Molecular function	Structural constituent of ribosome (3.1%, 4.03e-42)	Oxidoreductase activity (3.9%, 3.45e-14)
	Structural molecule activity (4.8%, 9.14e-33)	Transmembrane transporter activity (4.5%, 7.98e-14)
	RNA helicase activity (0.6%, 2.37e-13)	Substrate-specific transmembrane transporter activity (4.1%, 5.67e-12)
Cellular component	Ribonucleoprotein complex (10.7%, 1.39e-65)	Mitochondrial part (7.2%, 1.42e-19)
	Preribosome (2.4%, 3.28e-61)	Mitochondrion (16.2%, 1.51e-19)
	Cytosolic ribosome (2.5%, 4.67e-55)	Cell part (77.2%, 3.72e-15)

Table 9 Significant GO terms (process, function, component) for two biclusters, extracted from Yeast Cell-Cycle data using *BiFCA+*

	Bicluster 1	Bicluster 2
Biological process	Cytoplasmic translation (2.4%, 1.24e-06)	Single-organism process (49.3%, 0.09388)
	Single-organism process (49.3%, 1.82e-05)	
	Cell cycle process (8.4%, 5.39e-05)	
Molecular function	Structural molecule activity (4.8%, 0.00238)	N-methyltransferase activity (0.5 %, 0.2794)
	Structural constituent of ribosome (3.1%, 0.00310)	Transferase activity, transferring one-carbon groups (1.4%, 0.06158)
Cellular component	Cytosolic ribosome (2.5%, 7.85e-07)	Nuclear chromatin (1.8%, 0.00817)
	Non-membrane-bounded organelle (18.3%, 3.05e-06)	Cytosolic part (3.4 %, 0.01229)
	Intracellular non-membrane-bounded organelle (18.3%, 3.05e-06)	Cytosolic ribosome (2.5%, 0.05227)

Fig. 7 List of genes which concern the *Gene Ontology* term *"ribonucleoprotein complex"* (*Cellular component*) for the first bicluster (*Saccharomyces cerevisiae* dataset)

YBL072C YBL087C YBR048W YBR143C YBR189W YBR191W YBR247C YCL037C YCL054W YCR031C YCR057C YDL014W YDL031W YDL061C YDL148C YDL153C YDR012W YDR025W YDR064W YDR324C YDR382W YDR385W YDR398W YDR418W YDR447C YDR449C YDR450W YDR471W YDR500C YEL026W YEL054C YER006W YER025W YER074W YER117W YFL002C YFR031C-A YGL030W YGL076C YGL078C YGL099W YGL120C YGL147C YGL171W YGR128C YGR162W YGR214W YHL001W YHL033C YHR021C YHR052W YHR064C YHR066W YHR088W YHR089C YHR148W YHR169W YHR170W YHR196W YHR203C YIL052C YIR026C YJL033W YJL069C YJL109C YJL136C YJL138C YJL190C YJR094W-A YJR123W YJR145C YKL006W YKL081W YKL099C YKL156W YKR059W YKR060W YKR081C YLL008W YLL011W YLL045C YLR002C YLR048W YLR061W YLR175W YLR185W YLR186W YLR222C YLR244C YLR249W YLR264W YLR325C YLR340W YLR344W YLR367W YLR388W YLR409C YLR448W YMR128W YMR143W YMR146C YMR194W YMR230W YMR242C YNL002C YNL061W YNL067W YNL069C YNL075W YNL096C YNL301C YNR038W YNR053C YOL041C YOL077C YOL120C YOR056C YOR145C YOR234C YOR293W YOR312C YPL043W YPL079W YPL081W YPL126W YPL143W YPL198W YPL220W YPL266W YPR137W YPR144C

complex", in terms of *Cellular component* with a p value equal to $1.39e^{-65}$ (highly significant) and a background of 10.7%.

The results on these real-life datasets demonstrate that our proposed algorithm identifies biclusters with a high biological relevance.

4.5 Run time performs

Table 10 presents the comparison of the run time (in seconds) of our algorithm versus those respectively obtained by Trimax, BicFinder and BiMine. We note that for the Human B-cell Lymphoma dataset and *Saccharomyces cerevisiae*

Table 10 Execution time of the algorithms: *BiFCA+*, Trimax, BicFinder and BiMine

Algorithms	Execution time (s)		
	Yeast cell cycle	*Saccharomyces cerevisiae*	*Human B-cell lymphoma*
BiMine [4]	2 days	5 days	6 days
BicFinder [6]	300	29040	4680
Trimax [38]	**1.33**	250.83	63.96
BiFCA+	3.7	**180.53**	**8.10**

Bold values stand for the best results

datasets *BiFCA+* is the fastest, while BiMine is the costlier in execution time.

5 Conclusion

In this work, we have introduced the *BiFCA+* biclustering algorithm, a new FCA-based biclustering method for gene expression data.

Our approach consists in extracting formal concepts from a dataset after a discretization into a 3-state data matrix. A 3-state data matrix allows observing the profile of each gene through all pairs of conditions in the gene expression matrix. This latter discretization is used to extract formal concepts, a mathematical framework for deriving implicit relationships from a set of objects and their attributes. The resulting formal concepts represent biclusters. These biclusters are filtered with the help of the *Bond* correlation measure in order to remove the biclusters that have a high overlap.

The performances of the *BiFCA+* algorithm have been assessed on three real-life DNA microarray datasets. These experimentations show that *BiFCA+* enables extracting high quality biclusters. These biclusters have been evaluated with the *GO* annotations which checks the biological significance of biclusters. The obtained results confirm the *BiFCA+*'s ability to extract significant biclusters.

Future work will focus on the study of the extensions of the concepts of biclusters and formal concepts to those of triclusters and triconcepts [36]. Furthermore, in our forthcoming work, we will pay attention to the reduction of the obtained set of formal concepts [2] and the knowledge reduction of classical formal decision contexts [44]. Other avenues of future work also concern the extraction of biclusters by introducing biological knowledge during the extraction process. Moreover, we plan to use our method in other application domains such as text mining, target marketing and multimedia data processing. We also hope to enhance our experimentations by both extending our work to other correlation measures [53, 55] through classifying them into classes of measures sharing the same properties and using other statistical comparison criteria.

References

1. Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, Boldrick JC, Sabet H, Tran T, Yu X, Powell JI, Yang L, Marti GE, Moore T, Hudson JJ, Lu L, Lewis DB, Tibshirani R, Sherlock G, Chan WC, Greiner TC, Weisenburger DD, Armitage JO, Warnke R, Levy R, Wilson W, Grever MR, Byrd JC, Botstein D, Brown PO, Staudt LM (2000) Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. Nature 403(6769):503–511
2. Aswanikumar C, Srinivas S (2010) Concept lattice reduction using fuzzy k-means clustering. Expert Syst Appl 37(3):2696–2704. https://doi.org/10.1016/j.eswa.2009.09.026
3. Ayadi W (2011) Algorithmes systematiques et stochastiques de biregroupement pour l'analyse des donnees biopuces. Ph.D. thesis, University of Angers, France
4. Ayadi W, Elloumi M, Hao JK (2009) A biclustering algorithm based on a bicluster enumeration tree: application to DNA microarray data. BioData Mining 2:9
5. Ayadi W, Elloumi M, Hao JK (2010) Iterated local search for biclustering of microarray data. In: pattern recognition in bioinformatics–5th IAPR international conference, PRIB 2010, Nijmegen, The Netherlands, September 22-24, 2010. Proceedings, pp. 219–229
6. Ayadi W, Elloumi M, Hao JK (2012) Bicfinder: a biclustering algorithm for microarray data analysis. Knowl Inf Syst 30(2):341–358
7. Ayadi W, Elloumi M, Hao JK (2012) Bimine+: an efficient algorithm for discovering relevant biclusters of DNA microarray data. Knowl Based Syst 35:224–234
8. Barbut M, Monjardet B (1970) Ordre et classification: algèbre et combinatoire. Classiques Hachette. Hachette. https://books.google.fr/books?id=n3BpSgAACAAJ. Accessed Jan 2014
9. Ben-Dor A, Chor B, Karp RM, Yakhini Z (2003) Discovering local structure in gene expression data: the order-preserving submatrix problem. J Comput Biol 10(3/4):373–384
10. Bergmann S, Ihmels J, Barkai N (2004) Defining transcription modules using large-scale gene expression data. Bioinformatics 20(13):1993–2003
11. Berriz GF, King OD, Bryant B, Sander C, Roth FP (2003) Characterizing gene sets with funcassociate. Bioinformatics 19:2502–2504
12. Bleuler S, Prelic A, Zitzler E (2004) An EA framework for biclustering of gene expression data. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2004, 19-23 June 2004, Portland, OR, USA, pp. 166–173. https://doi.org/10.1109/CEC.2004.1330853
13. Boyle EI, Weng S, Gollub J, Jin H, Botstein D, Cherry JM, Sherlock G (2004) GO: : Termfinder-open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. Bioinformatics 20(18):3710–3715. https://doi.org/10.1093/bioinformatics/bth456
14. Madeira SC, Oliveira AL (2004) Biclustering algorithms for biological data analysis: a survey. IEEE Trans Comput Biol Bioinform 1:24–45
15. Cheng K, Law N, Chan Y, Siu W (2014) A joint framework for missing values estimation and biclusters detection in gene expression data. IJBRA 10(6):574–586. https://doi.org/10.1504/IJBRA.2014.065243

16. Cheng K, Law N, Siu W (2013) Use of biclustering for missing value imputation in gene expression data. Artif Intell Res 2(2):96–108. https://doi.org/10.5430/air.v2n2p96
17. Cheng KO, Law NF, Siu WC, Liew AWC (2008) Identification of coherent patterns in gene expression data using an efficient biclustering algorithm and parallel coordinate visualization. BMC Bioinform 9:210
18. Cheng Y, Church GM (2000) Biclustering of expression data. In: proc of ISMB, UC San Diego, California, pp 93–103
19. Cheng Y, Church GM (2006) Biclustering of expression data. Tech. rep., supplementary information
20. Das S, Idicula SM (2010) Application of cardinality based grasp to the biclustering of gene expression data. Int J Comput Appl 1:44–53
21. Divina F, Aguilar-Ruiz JS (2007) A multi-objective approach to discover biclusters in microarray data. In: genetic and evolutionary computation conference, GECCO 2007, proceedings, London, England, UK, July 7–11, 2007, pp 385–392. https://doi.org/10.1145/1276958.1277038
22. Divina F, AguilarRuiz JS (2006) Biclustering of expression data with evolutionary computation. IEEE Trans Knowl Data Eng 18(5):590–602
23. Eren K, Deveci M, Küçüktunç O, Çatalyürek ÜV (2013) A comparative analysis of biclustering algorithms for gene expression data. Brief Bioinform 14(3):279–292. https://doi.org/10.1093/bib/bbs032
24. Fisher RA (1922) On the interpretation of χ^2 from contingency tables, and the calculation of P. J R Stat Soc 85(1):87–94. https://doi.org/10.2307/2340521
25. Freitas A, Ayadi W, Elloumi M, Oliveira LJ, Hao JK (2013) Survey on biclustering of gene expression data. In: Elloumi M, Zomaya AY (eds) Biological knowledge discovery handbook: preprocessing, mining, and postprocessing of biological data. Wiley, Hoboken, New Jersey, pp 591–608
26. Gallo CA, Carballido JA, Ponzoni I (2009) Microarray biclustering: a novel memetic approach based on the pisa platform. In: Pizzuti C, Ritchie MD, Giacobini M (eds) Evolutionary computation, machine learning and data mining in bioinformatics. EvoBIO 2009. Springer, Berlin, Heidelberg, pp 44–55
27. Ganter B, Wille R (1999) Formal concept analysis–mathematical foundations. Springer
28. Gasch AP, Eisen MB (2002) Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. Genome Biol. https://doi.org/10.1186/gb-2002-3-11-research0059
29. Henriques R, Antunes C, Madeira SC (2013) Methods for the efficient discovery of large item-indexable sequential patterns. In: New frontiers in mining complex patterns–second international workshop, NFMCP 2013, Held in Conjunction with ECML-PKDD 2013, Prague, Czech Republic, September 27, 2013, Revised Selected Papers, pp 100–116. https://doi.org/10.1007/978-3-319-08407-7_7
30. Henriques R, Antunes C, Madeira SC (2015) A structured view on pattern mining-based biclustering. Pattern Recognit 48(12):3941–3958. https://doi.org/10.1016/j.patcog.2015.06.018
31. Henriques R, Madeira SC (2014) Bicpam: pattern-based biclustering for biomedical data analysis. Algorithm Mol Biol 9:27. https://doi.org/10.1186/s13015-014-0027-z
32. Henriques R, Madeira SC (2014) Bicspam: flexible biclustering using sequential patterns. BMC Bioinform 15:130. https://doi.org/10.1186/1471-2105-15-130
33. Henriques R, Madeira SC (2016) Bic2pam: constraint-guided biclustering for biological data analysis with domain knowledge. Algorithm Mol Biol 11:23. https://doi.org/10.1186/s13015-016-0085-5
34. Henriques R, Madeira SC (2016) Bicnet: flexible module discovery in large-scale biological networks using biclustering. Algorithm Mol Biol 11:14. https://doi.org/10.1186/s13015-016-0074-8
35. Hochreiter S, Bodenhofer U, Heusel M, Mayr A, Mitterecker A, Kasim A, Khamiakova T, Sanden SV, Lin D, Talloen W, Bijnens L, Göhlmann HWH, Shkedy Z, Clevert D (2010) FABIA: factor analysis for bicluster acquisition. Bioinformatics 26(12):1520–1527. https://doi.org/10.1093/bioinformatics/btq227
36. Ignatov DI, Gnatyshak DV, Kuznetsov SO, Mirkin BG (2015) Triadic formal concept analysis and triclustering: searching for optimal patterns. Mach Learning 101(1–3):271–302. https://doi.org/10.1007/s10994-015-5487-y
37. Ihmels J, Bergmann S, Barkai N (2004) Defining transcription modules using large-scale gene expression data. Bioinformatics 20:1993–2003
38. Kaytoue M, Kuznetsov SO, Macko J, Napoli A (2014) Biclustering meets triadic concept analysis. Ann Math Artif Intell 70(1–2):55–79. https://doi.org/10.1007/s10472-013-9379-1
39. Kaytoue M, Kuznetsov SO, Napoli A (2011) Biclustering numerical data in formal concept analysis. In: proc of ICFCA, Leuven, Belgium, pp 135–150
40. Kaytoue M, Kuznetsov SO, Napoli A, Duplessis S (2011) Mining gene expression data with pattern structures in formal concept analysis. Inf Sci 181(10):1989–2001. https://doi.org/10.1016/j.ins.2010.07.007
41. Király A, Laiho A, Abonyi J, Gyenesei A (2014) Novel techniques and an efficient algorithm for closed pattern mining. Expert Syst Appl 41(11):5105–5114. https://doi.org/10.1016/j.eswa.2014.02.029
42. Kumar CA (2012) Fuzzy clustering-based formal concept analysis for association rules mining. Appl Artif Intell 26(3):274–301
43. Lehmann F, Wille R (1995) A triadic approach to formal concept analysis. In: Conceptual structures: applications, implementation and theory, third international conference on conceptual structures, ICCS '95, Santa Cruz, California, USA, August 14–18, 1995, proceedings, pp 32–43. https://doi.org/10.1007/3-540-60161-9_27
44. Li J, Kumar CA, Mei C, Wang X (2017) Comparison of reduction in formal decision contexts. Int J Approx Reason 80:100–122. https://doi.org/10.1016/j.ijar.2016.08.007
45. Li X, Shao MW, Zhao XM (2016) Constructing lattice based on irreducible concepts. Int J Mach Learning Cybern. https://doi.org/10.1007/s13042-016-0587-y
46. Liu J, Li Z, Hu X, Chen Y (2009) Biclustering of microarray data with MOSPO based on crowding distance. BMC Bioinform. https://doi.org/10.1186/1471-2105-10-S4-S9
47. Liu J, Li Z, Liu F, Chen Y (2008) Multi-objective particle swarm optimization biclustering of microarray data. In: 2008 IEEE international conference on bioinformatics and biomedicine, BIBM 2008, 3–5 November 2008, Philadephia, Pennsylvania, USA, pp 363–366. https://doi.org/10.1109/BIBM.2008.17
48. Luan Y, Li H (2003) Clustering of time-course gene expression data using a mixed-effects model with b-splines. Bioinformatics 19(4):474–482
49. Martínez R, Pasquier N, Pasquier C (2008) Genminer: mining non-redundant association rules from integrated gene expression data and annotations. Bioinformatics 24(22):2643–2644. https://doi.org/10.1093/bioinformatics/btn490
50. Mitra S, Banka H (2006) Multi-objective evolutionary biclustering of gene expression data. Pattern Recognit 39:2464–2477
51. Mondal KC, Pasquier N (2014) Galois closure based association rule mining from biological data. In: Elloumi M, Zomaya AY (eds) Biological knowledge discovery handbook: preprocessing, mining, and postprocessing of biological data. Wiley, Hoboken, New Jersey, pp 761–802

52. Mondal KC, Pasquier N, Mukhopadhyay A, Maulik U, Bandyopadhyay S (2012) A new approach for association rule mining and bi-clustering using formal concept analysis. In: proc of machine learning and data mining in pattern recognition (MLDM), Berlin, Germany, pp 86–101
53. Mouakher A, Ben Yahia S (2016) Qualitycover: efficient binary relation coverage guided by induced knowledge quality. Inf Sci 355:58–73
54. Nepomuceno JA, Lora AT, Nepomuceno-Chamorro IA, Aguilar-Ruiz JS (2015) Integrating biological knowledge based on functional annotations for biclustering of gene expression data. Comput Method Progr Biomed 119(3):163–180. https://doi.org/10.1016/j.cmpb.2015.02.010
55. Omiecinski ER (2003) Alternative interest measures for mining associations in databases. IEEE Trans Knowl Data Eng 15:57–69
56. Orzechowski P (2013) Proximity measures and results validation in biclustering–a survey. In: Artificial intelligence and soft computing–12th international conference, ICAISC 2013, Zakopane, Poland, June 9–13, 2013, proceedings, part II, pp 206–217. https://doi.org/10.1007/978-3-642-38610-7_20
57. Pasquier N, Bastide Y, Taouil R, Lakhal L (1999) Discovering frequent closed itemsets for association rules. In: Beeri C, Buneman P (eds) ICDT. Springer, Berlin, Heidelberg, pp 398–416
58. Peddada S, Lobenhofer E, Li L, Afshari C, Weinberg C (2003) Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference. Bioinformatics 19:834–841
59. Pensa RG, Besson J, Boulicaut JF (2004) A methodology for biologically relevant pattern discovery from gene expression data. In: proc of discovery science, pp 230–241
60. Prelic A, Bleuler S, Zimmermann P, Wille A, Buhlmann P, Gruissem W, Hennig L, Thiele L, Zitzler E (2006) A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22(9):1122–1129
61. Tanay A, Sharan R, Shamir R (2002) Discovering statistically significant biclusters in gene expression data. Bioinformatics 18:S136–S144
62. Tavazoieand S, Hughes JD, Campbell MJ, Cho RJ, Church GM (1999) Systematic determination of genetic network architecture-genetics. Nat Genet 22:281–285
63. Teng L, Chan L (2008) Discovering biclusters by iteratively sorting with weighted correlation coefficient in gene expression data. J Signal Process Syst 50:267–280
64. Trabelsi C, Jelassi N, Ben Yahia S (2012) Scalable mining of frequent tri-concepts from folksonomies. In: Advances in knowledge discovery and data mining–16th Pacific-Asia conference, PAKDD 2012, Kuala Lumpur, Malaysia, May 29–June 1, 2012, proceedings, part II, pp 231–242. Springer-Verlag. https://doi.org/10.1007/978-3-642-30220-6_20
65. Uno T, Asai T, Uchida Y, Arimura H (2004) An efficient algorithm for enumerating closed patterns in transaction databases. In: Discovery science, 7th international conference, DS 2004, Padova, Italy, October 2–5, 2004, proceedings, pp 16–31. https://doi.org/10.1007/978-3-540-30214-8_2
66. Wang H, Wang W, Yang J, Yu PS (2002) Clustering by pattern similarity in large data sets. In: Proceedings of the 2002 ACM SIGMOD international conference on management of data, Madison, Wisconsin, June 3–6, 2002, pp 394–405. https://doi.org/10.1145/564691.564737
67. Wei J, Wang S, Yuan X (2010) Ensemble rough hypercuboid approach for classifying cancers. IEEE Trans Knowl Data Eng 22(3):381–391. https://doi.org/10.1109/TKDE.2009.114
68. Wille R (1982) Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival I (ed) Ordered Sets. Reidel, Dordrecht/Boston, pp 445–470
69. Zhang Y, Zha H, Chu CH (2005) A time-series biclustering algorithm for revealing co-regulated genes. Proc 5th Int Conf Inf Technol 1:32–37