A memetic algorithm for discovering negative correlation biclusters of DNA microarray data

Wassim Ayadi[a,b], Jin-Kao Hao[b,*]

[a] *LaTICE, School of Science and Technology of Tunis, University of Tunis, 1008 Tunis, Tunisia*

[b] *LERIA, University of Angers, 2 Boulevard Lavoisier, 49045 Angers, France*

(Accepted to *Neurocompting*, 15 May 2014)

Abstract

Most biclustering algorithms for microarrays data analysis focus on positive correlations of genes. However, recent studies demonstrate that groups of biologically significant genes can show negative correlations as well. So, discovering negatively correlated patterns from microarrays data represents a real need. In this paper, we propose a Memetic Biclustering Algorithm (MBA) which is able to detect negatively correlated biclusters. The performance of the method is evaluated on two well-known microarray datasets (*Yeast cell cycle* and *Saccharomyces cerevisiae*), showing that MBA is able to obtain statistically and biologically significant biclusters.

Keywords: Biclustering; Microarrays data; Negative correlations; Memetic algorithm.

1. Introduction

DNA microarray technology permits to measure simultaneously the expression levels of thousands of genes under diverse experimental conditions. This technology typically generates large amounts of raw data that need to be analyzed to draw useful information for specific biological studies and medical applications. In this context, biclustering of DNA microarray data is a particularly interesting approach since it allows the simultaneous identification of groups of genes that show highly correlated expression patterns through

[*] Corresponding author.

groups of experimental conditions (samples) (Bar-Joseph, 2004; Madeira and Oliveira, 2004; Hanczar and Nadif, 2011; Han and Yan, 2012).

Gene expressions from DNA microarrays are usually represented by an $n \times m$ data matrix $M(I, J)$ where n and m are respectively the number of measured genes and the number of conditions (or time points). Each cell $M[i, j]$ ($i \in I=\{1, 2, \ldots, n\}$, $j \in J=\{1, 2, \ldots, m\}$) represents the expression level of the i^{th} gene under the j^{th} condition. A bicluster is a subset of genes associated with a subset of conditions, i.e., a couple (I', J') such that $I' \subseteq I$ and $J' \subseteq J$.

Given a data matrix $M(I, J)$, the biclustering problem consists in extracting from $M(I, J)$ a group of coherent and significant biclusters of large size. In its general form, the biclustering problem is NP-hard (Cheng and Church, 2000; Madeira and Oliveira, 2004).

Existing biclustering algorithms can be grouped into two large classes (Ayadi et al., 2014): Those that adopt a systematic search approach and those that adopt a stochastic search framework, also called heuristic or metaheuristic approach. Representative systematic search algorithms include greedy algorithms (Ayadi et al., 2012b; Ben-Dor et al., 2002; Cheng and Church, 2000; Cheng et al., 2008; Liu and Wang, 2007; Teng and Chan, 2008), divide-and-conquer algorithms (Hartigan, 1972; Prelic et al., 2006) and enumeration algorithms (Ayadi et al., 2009, 2012a; Liu and Wang, 2003; Tanay et al., 2002). Stochastic search algorithms include neighborhood-based algorithms (Ayadi et al., 2012c; Bryan et al., 2006), GRASP (Das and Idicula, 2010; Dharan and Nair, 2009) and evolutionary algorithms (Bleuler et al., 2004; Divina and Aguilar-Ruiz., 2007; Gallo et al., 2009; Mitra and Banka, 2006). A recent review of various biclustering algorithms for biological data analysis is provided in (Valente-Freitas et al., 2013).

A majority of existing biclustering algorithms extract only positive correlated genes. However, recent studies show that a group of biologically significant genes can present negative correlations. Figure 1 shows an example of these correlations. Contrary to the case of a positive correlation where genes present similar patterns, in a negative correlation, genes present opposite patterns.

For example, in their study on the development of expression patterns for Arabidopsis thaliana, Schmid et al. (Schmid et al., 2005) found that two groups of genes show negative correlations from an early seed development stage to a late stage. In Zhao et al. (Zhao et al., 2008), the authors considered the negative correlated genes. They found that genes YLR367W

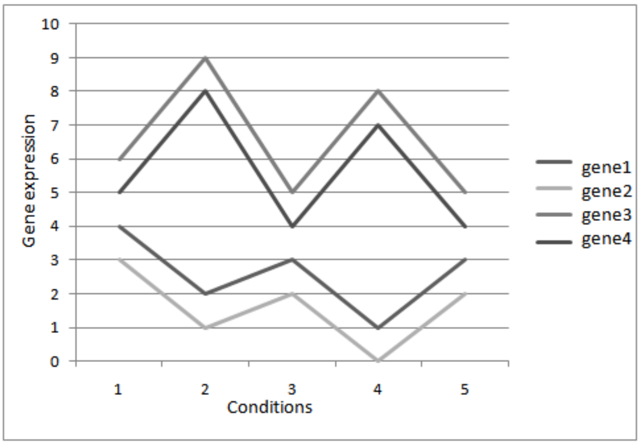


Figure 1: Genes 1 and 2 are negatively correlated with the genes 3 and 4

and YKR057W of the *Yeast* data share the same pattern, while they have a negative correlated pattern against gene YML009C under 8 conditions. These genes are grouped into the same bicluster because they are involved in protein translation and translocation.

In this paper, we address the issue of finding negative correlations based on local pattern of gene expression profiles. The key originality of our MBA method concerns the use of *positive and negative bicluster patterns* both in its search strategies and neighborhood definition. Bicluster pattern is a characteristic representation of a bicluster and is used to evaluate genes/conditions of biclusters. Positive bicluster pattern is used to improve the quality of a given initial positive bicluster, while the negative bicluster pattern is used to add negative correlation genes to the same bicluster. In the general case, if the absolute value of the correlation is considered, positive and negative correlation biclusters can be extracted without distinguishing the two types of correlations. However, the goal of our algorithm is to build the negative correlation biclusters.

The remainder of the paper is organized as follows: In section 2, we present the *Average Spearman's Rho* (ASR) evaluation function. In section 3, we describe the proposed MBA algorithm. In section 4, experimental studies of MBA on real DNA microarray datasets are presented. Moreover, we illustrate a biological validation of some extracted biclusters via two webtools, *FuncAssociate* (Berriz et al., 2003) and *GOTermFinder*[1]. Conclusions are given in the last section.

2. The ASR evaluation function

Many evaluation functions exist for bicluster evaluation such as Euclidean distance, Pearson correlation and Mean Squared Residue (MSR). Among these measures, MSR is the most popular evaluation function (Cheng and Church, 2000). It has been used by several biclustering algorithms (Angiulli et al., 2008; Bleuler et al., 2004; Cheng et al., 2008; Dharan and Nair, 2009; Mitra and Banka, 2006; Yang et al., 2003; Zhang et al., 2004). However, MSR is deficient to assess correctly the quality of certain types of biclusters like multiplicative models (Aguilar-Ruiz, 2005; Cheng et al., 2008; Pontes et al., 2007; Teng and Chan, 2008).

In (Ayadi et al., 2009), the authors have proposed another evaluation function, called *Average Spearman's Rho* (ASR). Let (I', J') be a bicluster in a data matrix $M(I, J)$, the ASR evaluation function is then defined by:

$$ASR(I', J') = 2 * max \left\{ \frac{\sum_{i \in I'} \sum_{j \in I'; j \geq i+1} \rho_{ij}}{|I'|(|I'|-1)}, \frac{\sum_{k \in J'} \sum_{l \in J'; l \geq k+1} \rho_{kl}}{|J'|(|J'|-1)} \right\} \tag{1}$$

where ρ_{ij} $(i \neq j)$ is the spearman's rank correlation (Lehmann and D'Abrera, 1998) associated with the row indexes i and j in the bicluster (I', J'), ρ_{kl} $(k \neq l)$ is the spearman's rank correlation associated with the column indices k and l in the bicluster (I', J') and ASR(I', J') $\in$[-1..1].

A high (resp. low) ASR value, close to 1 (resp. close to -1), indicates that the genes/conditions of the bicluster are positively (resp. negatively) correlated. ASR can thus be used to measure effectively both positive and negative correlations.

In the next section, we describe the proposed memetic biclustering algorithm MBA which uses the ASR measure.

[1] http://db.yeastgenome.org/cgi-bin/GO/goTermFinder

3. The MBA algorithm

3.1. Memetic algorithm

A memetic algorithm (MA) is typically based on the population-based search and neighborhood-based local search (Moscato, 1999). The basic rationale behind a MA is to combine these two different search methods in order to take advantage of their complementary search strategies. Indeed, it is generally believed that the population-based search framework offers more facilities for exploration while neighborhood search provides more capabilities for exploitation. If they are combined in a suitable way, the resulting hybrid method can then offer a good balance between exploitation and exploration, assuring a high search performance (Hao, 2012).

Mitra and Banka (Mitra and Banka, 2006) present a *Multi-Objective Evolutionary Algorithm* (MOEA) based on Pareto dominance. The authors try to find biclusters with maximum size and homogeneity by using a multi-objective genetic algorithm called *Non-dominated Sorting Genetic Algorithm* (NSGA-II) (Coello et al., 2002) in combination with a local search procedure. Gallo *et al.* (Gallo et al., 2009) illustrate another MOEA algorithm combined with a local search strategy. They extract biclusters with multiple criteria like maximum rows, columns, homogeneity and row variance.

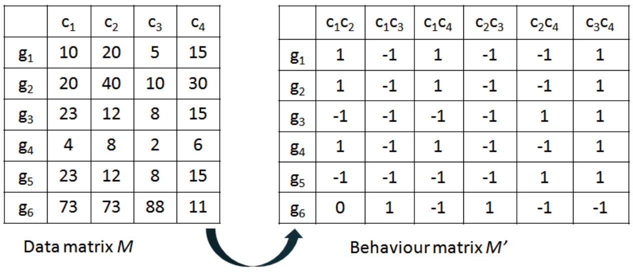
3.2. Preprocessing of gene expression matrix

Our algorithm applies a preprocessing step to transform the input data matrix M to a *Behavior Matrix* M'. This preprocessing step aims to highlight the trajectory patterns of genes. Indeed, according to (Schmid et al., 2005; Zhao et al., 2008), a group of genes is considered to be biologically significant if they present negative correlations. Within the transformed matrix M', each row represents the trajectory pattern of a gene across all the combined conditions while each column represents the trajectory pattern of all the genes under a pair of particular conditions in the data matrix M. The whole matrix M' provides thus useful information for the identification of relevant correlation biclusters.

Formally, the behavior matrix M' is constructed progressively by merging pairs of columns (conditions) from the input data matrix M. Since M has n rows and m columns, there is $m(m-1)/2$ distinct combinations between columns, represented by J''. So, M' has n rows and $m(m-1)/2$ columns. M' is defined as follows:

$$M'[i,l] = \begin{cases} 1 & \text{if } M[i,k] < M[i,q] \\ -1 & \text{if } M[i,k] > M[i,q] \\ 0 & \text{if } M[i,k] = M[i,q] \end{cases} \tag{2}$$

with $i \in [1..n]$, $l \in [1..J'']$, $k \in [1..m-1]$, $q \in [2..m]$ and $q \geq k+1$.



	c_1	c_2	c_3	c_4
g_1	10	20	5	15
g_2	20	40	10	30
g_3	23	12	8	15
g_4	4	8	2	6
g_5	23	12	8	15
g_6	73	73	88	11

Data matrix *M*

	c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
g_1	1	-1	1	-1	-1	1
g_2	1	-1	1	-1	-1	1
g_3	-1	-1	-1	-1	1	1
g_4	1	-1	1	-1	-1	1
g_5	-1	-1	-1	-1	1	1
g_6	0	1	-1	1	-1	-1

Behaviour matrix *M'*

Figure 2: Input data matrix M and its behaviour matrix M'

Figure 2 shows an illustrative example. We can observe, by considering each row of M', the trajectory (or behavior) pattern of each gene through all the combined conditions, i.e., up (1), down (-1) and no change (0). This figure also shows the trajectory of all rows (genes) over combined columns (combined conditions). Similarly, the combinations of all the paired conditions give useful information since a bicluster may be composed of a subset of non contiguous conditions. Our MBA algorithm uses M' to define its search space as well as its neighborhood that is critical for the search process.

3.3. General procedure of MBA

The key originality of MBA concerns the use of *positive and negative bicluster patterns* both in its search strategies and neighborhood definition. The bicluster pattern is a characteristic representation of a bicluster. It can be used to evaluate genes/conditions of biclusters. The positive bicluster pattern is used to improve the quality of the positive bicluster B, and the negative bicluster pattern is used to add negative correlation genes to the same bicluster B. This representation is defined by the behavior matrix of the bicluster, i.e., the trajectory patterns of the genes under all combined conditions of the bicluster for the positive correlations and the inverted patterns for the negative correlations.

Starting from a population of initial biclusters, MBA first uses the combination operator to obtain two offspring biclusters. The local search procedure is then applied to the best offspring bicluster to improve its quality by following a pattern-based neighborhood. By using the bicluster patterns, we define a set of rules which allow us to qualify the goodness (or badness) of a gene and condition. Using these rules, MBA updates its population with the improved new bicluster. This procedure stops when a fixed number of generations is reached.

The general MBA procedure is given in Algorithm 1. We describe in the following sections its main ingredients.

3.4. Population

The population of our algorithm is created by using the behavior matrix M' obtained from the preprocessing step described previously. More precisely, given a bicluster $B = (I', J')$, we encode the bicluster by its behavior matrix $s = (I', K)$ which is the sub-matrix of M' including only the set of genes in I' and all the combinations of paired conditions in J' (see example of Figure 3). It is clear that s has the same rows as B, its number K of columns is equal to $|J'|(|J'|-1)$. The population can be generated by any means. For instance, this can be done randomly with a risk of starting with biclusters of bad quality. A more interesting strategy is to employ a fast greedy algorithm to obtain rapidly a bicluster of reasonable quality. We use this strategy in this work and adopt two well-known algorithms: one is presented by Cheng and Church (Cheng and Church, 2000) and the other is called OPSM which is introduced in (Ben-Dor et al., 2002). As explained above, each bicluster of the population is encoded into its behavior matrix.

In the rest of this paper, the behavior matrix s of a bicluster of the population is called a *solution*.

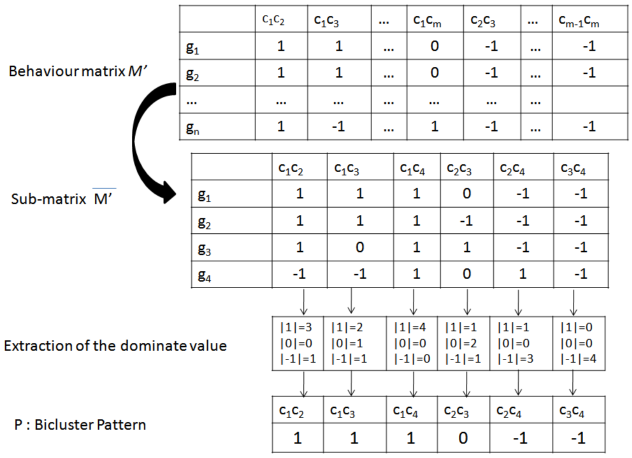
3.5. Crossover

Combination (or crossover) aims to create new promising candidate solutions by combining existing solutions. In this context, our crossover operator is applied to two randomly chosen parents to create two new offspring biclusters by considering separately the genes (rows) and conditions (columns). Let Bi_1 and Bi_2 be two parents:

$$Bi_1\text{: } g_1\ g_2\ \ldots\ g_n\ //\ c_1\ c_2\ \ldots\ c_p$$
$$Bi_2\text{: } g_1\ g_2\ \ldots\ g'_m\ //\ c'_1\ c'_2\ \ldots\ c'_q$$

Algorithm 1 General MBA Procedure

```
Input: Initial data matrix $M$, Population of biclusters, quality thresholds: $\alpha$, $\beta$, $threshold\_ASR$; Maximum number of iterations $Y, Z$
Output: Population of biclusters
Create the Behaviour Matrix $M'$ from $M$    /* Sect. 3.2 */
repeat
    Choose randomly two positive biclusters as parents from Population
    Apply the crossover operator to obtain two offspring biclusters  /* See Sect. 3.5 */
    Choose the best offspring bicluster $b$ that has the best $ASR(b)$  /* See Sect. 2 */
    repeat
        Create the behaviour sub-Matrix $\bar{M}'$ for $b$  /* Sect. 3.6 */
        $s_0 \leftarrow \bar{M}'$    /* Set the initial solution */
        Construct the positive bicluster pattern $P$ from $s_0$
        $s_1 \leftarrow s_0 \oplus mv_g^+(\alpha)$ /* Apply the row (gene) move operator by using positive pattern $P$, see Sect. 3.6 */
        Construct the negative bicluster pattern $\bar{P}$ from $s_1$
        $s_2 \leftarrow s_1 \oplus mv_g^-(\alpha)$ /* Apply the row (gene) move operator by using negative $\bar{P}$, see Sect. 3.6 */
        $s_3 \leftarrow s_2 \oplus mv_c(\beta)$ /* Apply the column (condition) move operator by using both $P$ and $\bar{P}$ */
        Reconstruct bicluster $B$ from $s_3$
    until ($|ASR(B)| \geq Threshold\_ASR$ or we reach the maximum number of iterations $Y$)
    Insert the bicluster $B$ in the population if $|ASR(B)| \geq Threshold\_ASR$
until (we reach the maximum number of iterations $Z$)
Return $Population$
```



Behaviour matrix M'

	c_1c_2	c_1c_3	...	c_1c_m	c_2c_3	...	$c_{m-1}c_m$
g_1	1	1	...	0	-1	...	-1
g_2	1	1	...	0	-1	...	-1
...	...	...	...	...	...	...	
g_n	1	-1	...	1	-1	...	-1

Sub-matrix $\overline{M'}$

	c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
g_1	1	1	1	0	-1	-1
g_2	1	1	1	-1	-1	-1
g_3	1	0	1	1	-1	-1
g_4	-1	-1	1	0	1	-1

Extraction of the dominate value

|1|=3 |0|=0 |-1|=1	|1|=2 |0|=1 |-1|=1	|1|=4 |0|=0 |-1|=0	|1|=1 |0|=2 |-1|=1	|1|=1 |0|=0 |-1|=3	|1|=0 |0|=0 |-1|=4

P : Bicluster Pattern

c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
1	1	1	0	-1	-1

Figure 3: Construction of bicluster pattern P

where $g_n \leq g'_m$ and $c_p \leq c'_q$

First, we generate two random integers Ψ_1 and Ψ'_1 corresponding respectively to the crossover point in the first part (i.e. genes) and second part (i.e., conditions) in Bi_1 such as $g_1 \leq \Psi_1 \leq g_n$ and $c_1 \leq \Psi'_1 \leq c_p$. Second, we generate $\Psi_2 = g'_i$ and $\Psi'_2 = c'_j$ as the crossover points respectively in gene part and condition part in Bi_2 where $g'_{i-1} \leq \Psi_2 \leq g'_i$ and $c'_{j-1} \leq \Psi'_2 \leq c'_j$.

For example, let us consider the following parents:

$$Bi_1\text{: } 2\ 3\ 6\ 7\ 9\ 11\ 12\ 16\ 20\ //\ 0\ 5\ 9\ 10$$
$$Bi_2\text{: } 0\ 3\ 4\ 8\ 10\ 14\ 21\ 25\ 26\ 28\ 30\ //\ 2\ 4\ 6\ 8\ 12$$

Suppose that $\Psi_1 = 11$ and $\Psi'_1 = 5$ therefore $\Psi_2 = 14$ and $\Psi'_2 = 6$
Thus the obtained offspring biclusters are:

$$C_1\text{: } 2\ 3\ 6\ 7\ 9\ 11\ 14\ 21\ 25\ 26\ 28\ 30\ //\ 0\ 5\ 6\ 8\ 12$$
$$C_2\text{: } 0\ 3\ 4\ 8\ 10\ 12\ 16\ 20\ //\ 2\ 4\ 9\ 10$$

Third, we evaluate the two obtained biclusters with the ASR evaluation function and we keep only the best one for the next step.

3.6. Local improvement of offspring biclusters

The goal of local improvement is to improve the quality of an offspring bicluster as far as possible. For this purpose, the proposed local improvement procedure takes the selected offspring bicluster as its input (current solution) and then iteratively replaces the current solution by another solution taken from a given neighborhood.

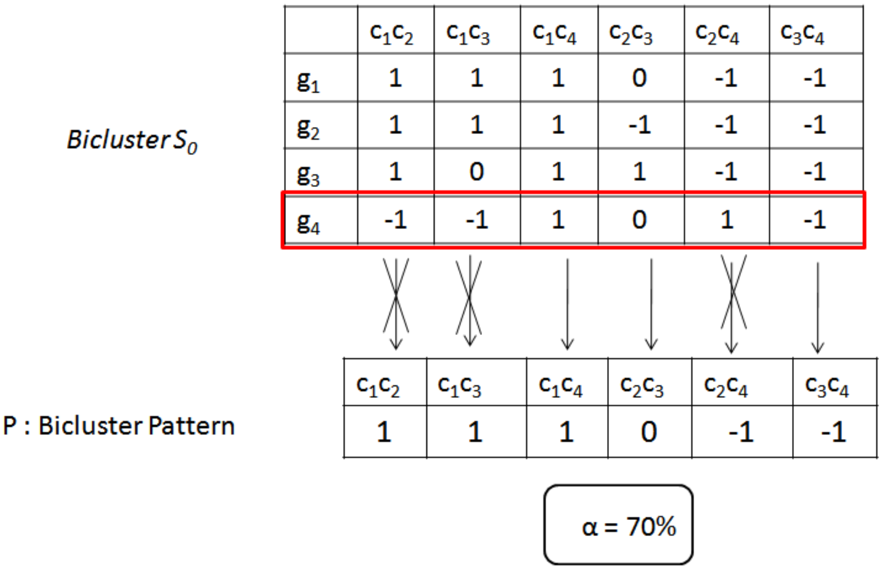
The neighborhood can generally be defined by a *move* operator. Given a solution, let mv be the move operator that can be applied to the solution. Then each application of mv transforms s_0 into a new solution s_1. This is typically denoted by $s_1 \leftarrow s_0 \oplus mv$.

In this paper, we devise three specially designed move operators that transform a given solution. Two of them operates on rows (genes) while the last one operates on columns (combinations of pairwise conditions) of a given solution. These operators are based on the general drop/add operation which removes some elements and adds other new elements in the given solution. The critical issue here is the criterion that is employed to determine the elements to be removed and added. In our case, this decision is based on the positive and negative patterns.

Our first move operator, denoted by mv_g^+, performs changes by removing a number of rows (genes) of the bicluster and adding other genes with a positive pattern in order to obtain more coherent biclusters. Let $s = (I', K)$ be a solution, we first extract from the behavior matrix M' the associated sub-matrix $\bar{M}'$. Let R and C denote respectively the index set of rows and columns of $\bar{M}'$. From $\bar{M}'$ we build the positive bicluster pattern P of s which is defined by a vector indexed by C. $P[j], j \in C$, takes the dominating value $k \in \{1, 0, -1\}$ such that k has the highest appearance in the column i of $\bar{M}'$.

Now for each gene $g_i, i \in R$ of the solution s, we define the quality of g_i as the percentage of concordances between the behavior pattern of g and the positive behavior pattern P of bicluster s. Let α be a fixed quality threshold of genes. Let D denote the set of bad genes of s such that their quality do not reach the quality threshold fixed by α. Let G denote the set of good genes missing from s such that their quality surpasses the quality threshold α. Then our first move operator mv_g removes from s all the bad genes of D and adds the positive gene from G.

Figure 4 shows an example where one bad gene (g_4) is deleted and Figure 5 shows one good gene (g_{10}^+) is added. g_4 is bad because its behavior pattern has a low concordance with the bicluster behavior pattern (only 50% which is inferior to the quality threshold $\alpha = 70\%$). Similarly, g_{10}^+ is good because its quality (83%) is higher than α. This replacement increases thus the positive coherence of the resulting bicluster. In the general case, the number of deleted gene may differ from the number of added genes. Notice that this



Bicluster S_0

	c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
g_1	1	1	1	0	-1	-1
g_2	1	1	1	-1	-1	-1
g_3	1	0	1	1	-1	-1
g_4	-1	-1	1	0	1	-1

P : Bicluster Pattern

c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
1	1	1	0	-1	-1

$\alpha = 70\%$

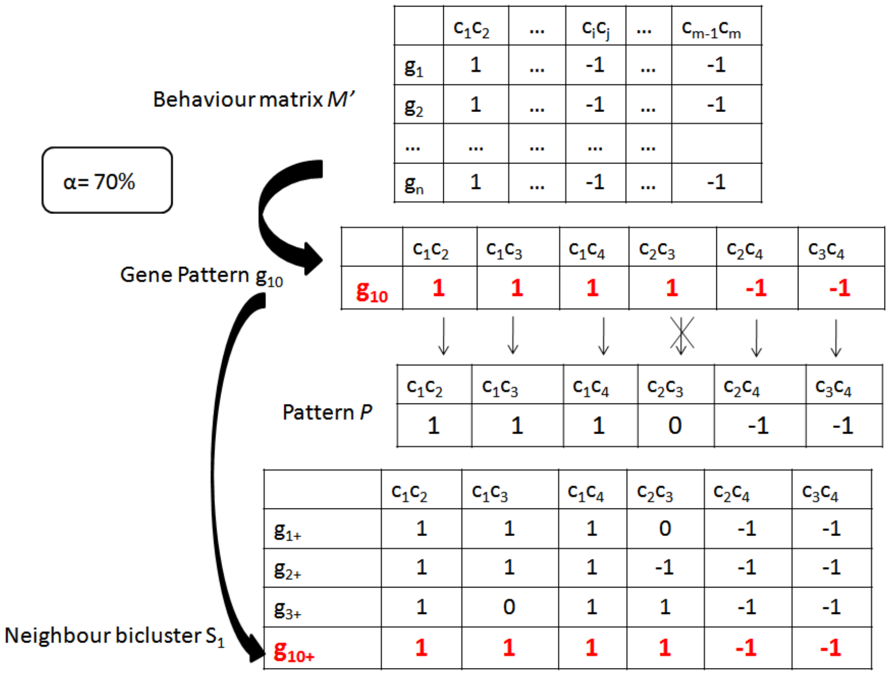
Figure 4: Positive row move operator mv_g^+: A bad gene (g_4) is deleted since its quality (50%) is inferior to $\alpha = 70\%$

move operator does not change the columns of the solution.

Our second move operator, denoted by mv_g^-, performs changes by adding other genes with a negative pattern in order to obtain biclusters with negative correlations. Similar to the first move operator, mv_g^- uses the negative bicluster pattern ($\bar{P}$) which is constructed by inverting all the values of the positive bicluster pattern P. mv_g^- uses a quality threshold α for each row. The quality of each row is defined as the percentage of concordances between the row pattern and the value of this row in the bicluster pattern.

Then, when our second move operator mv_g^- adds a good gene from the current bicluster, we select a gene under the same subset of conditions from the ”behavior matrix” M' which has a dominating value relative to $\bar{P}$ higher than a fixed threshold α. Notice that this move operator does not change the conditions of the solution (see example of Figure 6). At the end, we obtain a bicluster with two parts: a part of genes (that are marked with ”+”) which are relative to the positive pattern P, and the other part (that are marked with ”-”) which are relative to the negative pattern $\bar{P}$.

Our third move operator, denoted by mv_c, removes a number of columns (combined conditions) and adds other columns in order to obtain more co-



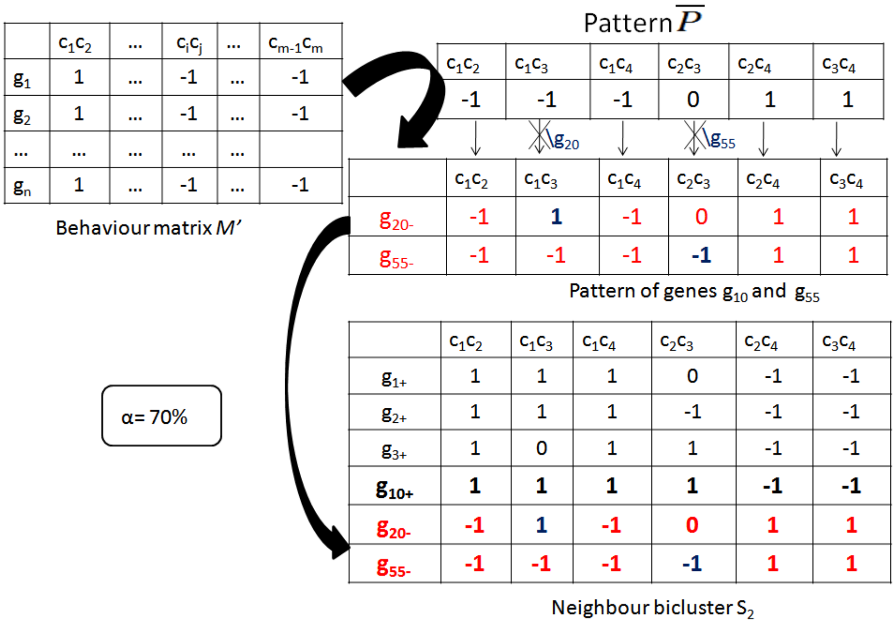
	c_1c_2	...	c_ic_j	...	$c_{m-1}c_m$
g_1	1	...	-1	...	-1
g_2	1	...	-1	...	-1
...	...	...	...	...	
g_n	1	...	-1	...	-1

	c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
g_{10}	**1**	**1**	**1**	**1**	**-1**	**-1**

c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
1	1	1	0	-1	-1

	c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
g_{1+}	1	1	1	0	-1	-1
g_{2+}	1	1	1	-1	-1	-1
g_{3+}	1	0	1	1	-1	-1
g_{10+}	**1**	**1**	**1**	**1**	**-1**	**-1**

Figure 5: Positive row move operator mv_g^+: A good g_{10} is selected and added which has a quality (83%) superior to $\alpha = 70\%$



	c_1c_2	...	c_ic_j	...	$c_{m-1}c_m$
g_1	1	...	-1	...	-1
g_2	1	...	-1	...	-1
...	...	...	...	...	
g_n	1	...	-1	...	-1

Behaviour matrix *M'*

Pattern $\overline{P}$

c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
-1	-1	-1	0	1	1

	c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
g_{20-}	-1	**1**	-1	0	1	1
g_{55-}	-1	-1	-1	**-1**	1	1

Pattern of genes g_{10} and g_{55}

α= 70%

	c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
g_{1+}	1	1	1	0	-1	-1
g_{2+}	1	1	1	-1	-1	-1
g_{3+}	1	0	1	1	-1	-1
g_{10+}	**1**	**1**	**1**	**1**	**-1**	**-1**
g_{20-}	**-1**	**1**	**-1**	**0**	**1**	**1**
g_{55-}	**-1**	**-1**	**-1**	**-1**	**1**	**1**

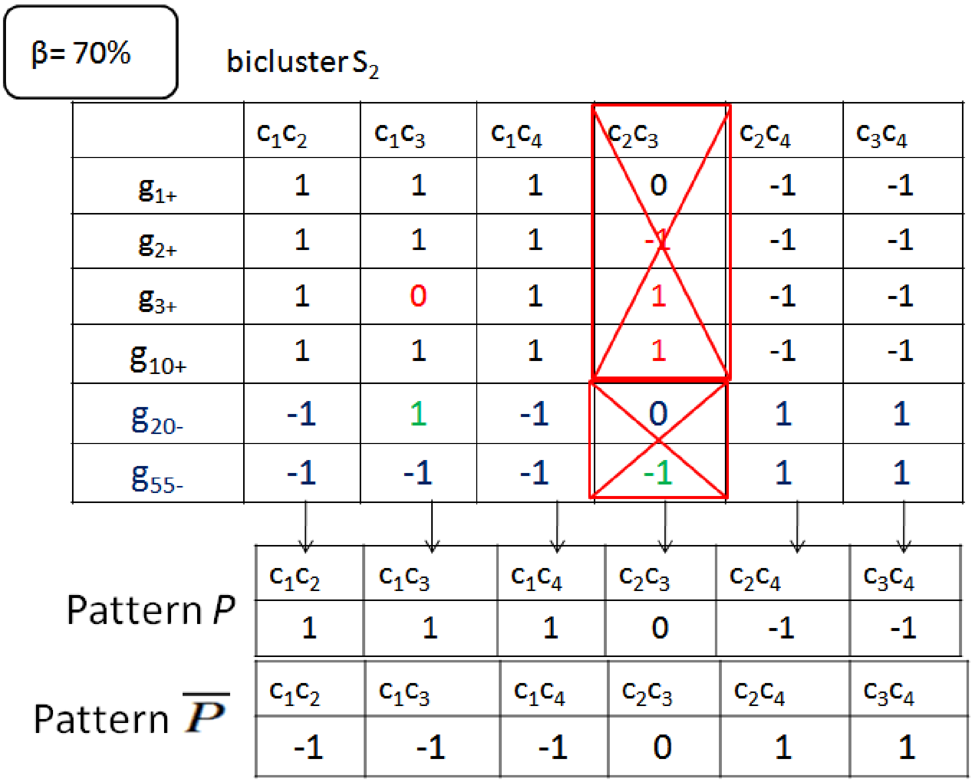
Neighbour bicluster S_2

Figure 6: Negative row move operator mv_g^-: Good genes g_{20} and g_{55} are selected and added which have a quality (83%) superior to $\alpha = 70\%$

herent biclusters by using P and $\bar{P}$. Similar to the two other move operators, mv_c uses a quality threshold β for each column. The quality of each column is divided into two parts: One part for positive genes and another part for negative genes, and each part is evaluated separately for each type of pattern. The quality of the positive part is defined as the percentage of concordances between the positive column pattern (marked with positive genes) and the value of this column in the bicluster pattern P. The quality of the negative part is defined as the percentage of concordances between the negative column pattern (marked with negative genes) and the value of this column in the bicluster pattern $\bar{P}$.

Then, when our third move operator mv_c detects a bad condition with P or $\bar{P}$ from the current bicluster, we test if the dominating value of each condition of the current bicluster has the same value with the corresponding value in the bicluster pattern P for the positive genes and $\bar{P}$ for the negative genes. If they are different (for P or $\bar{P}$), this condition is considered as bad (and removed from the current bicluster) (see example of Figure 7). To add a good condition to the current bicluster, we select a condition under the same subset of positive genes from the "behavior matrix" M' which has a dominating value higher than a fixed threshold β and the same subset of negative genes from the "behavior matrix" M' which has a dominating value higher than a fixed threshold β (see example of Figure 8). Notice that this move operator does not change the rows of the solution. In the general case, the number of deleted columns may differ from the number of added columns at each application of this move operator.

For a given solution, our MBA algorithm applies these three move operators to reach a local optimum s (with a ASR value higher than the fixed $threshold\_ASR$ threshold). This local optimum solution s is composed of a group of genes and columns, each column representing the positive and negative trajectory pattern of two conditions across the group of genes. Among the combinations of conditions in s, some conditions may be combined with only a few other conditions. These conditions are in fact insignificant conditions for the extracted bicluster. For this reason, during the decoding process (transforming s into a bicluster B), we retain only conditions which are combined with at least 50% other selected conditions. For instance, if we have $s = \{(g_1, g_2, g_3, g_4); (c_1c_2, c_1c_3, c_1c_4, c_2c_3)\}$, condition c_4 will not be kept in the final bicluster because it is not combined at least with 50% of the other conditions, i.e., c_2 and c_3. The bicluster obtained is thus $B = \{(g_1, g_2, g_3, g_4); (c_1, c_2, c_3)\}$.



β= 70%

bicluster S_2

	c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
g_{1+}	1	1	1	0	-1	-1
g_{2+}	1	1	1	-1	-1	-1
g_{3+}	1	0	1	1	-1	-1
g_{10+}	1	1	1	1	-1	-1
g_{20-}	-1	1	-1	0	1	1
g_{55-}	-1	-1	-1	-1	1	1

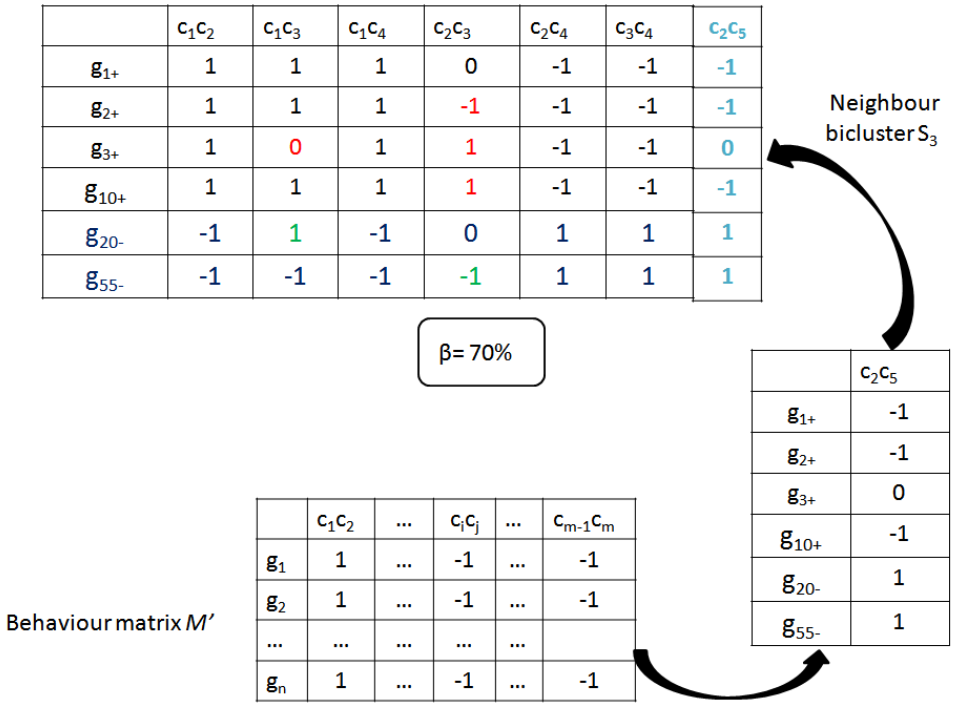
Pattern P

c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
1	1	1	0	-1	-1

Pattern $\overline{P}$

c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
-1	-1	-1	0	1	1

Figure 7: Column move operator: Column c_2c_3 has a quality inferior to the threshold $\beta = 70\%$ for P and $\bar{P}$ and thus removed



	c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4	c_2c_5
g_{1+}	1	1	1	0	-1	-1	-1
g_{2+}	1	1	1	-1	-1	-1	-1
g_{3+}	1	0	1	1	-1	-1	0
g_{10+}	1	1	1	1	-1	-1	-1
g_{20-}	-1	1	-1	0	1	1	1
g_{55-}	-1	-1	-1	-1	1	1	1

	c_2c_5
g_{1+}	-1
g_{2+}	-1
g_{3+}	0
g_{10+}	-1
g_{20-}	1
g_{55-}	1

	c_1c_2	...	c_ic_j	...	$c_{m-1}c_m$
g_1	1	...	-1	...	-1
g_2	1	...	-1	...	-1
...	...	...	...	...	
g_n	1	...	-1	...	-1

Figure 8: Column move operator: c_2c_5 with a quality superior to $\beta = 70\%$ is selected and added

3.7. Update of the population

This step decides whether a new solution should become a member of the population and which existing solution of the population should be replaced. To maintain an appropriate diversity of the population, we compute the value of the Jaccard index which is used to measure the overlap between the offspring bicluster B (improved by local search) and the two parents. We insert the offspring in the population if $|ASR(B)| \geq Threshold\_ASR$ and we delete the solution of the population that has a high overlap with the offspring. This replacement rule prevents the search process from a premature convergence, and helps the algorithm to continually discover new promising search areas.

3.8. Time complexity of the algorithm

For one generation, the algorithm has to compute the selection operator in $O(1)$ time, the crossover operator in $O(n+m)$ time where n and m represent the total number of genes and conditions, and the local improvement operator in $O(Yn^2m)$ time where Y is the maximum number of local improvement iterations. Hence in the worst case, for one generation, the algorithm requires $O(Yn^2m)$ time.

4. Experimental Studies

In this section, we assess the MBA algorithm on two DNA microarray data: Saccharomyces cerevisiae and Yeast cell-cycle. We evaluate our proposed method against the results of some prominent biclustering algorithms used by the community, namely, CC (Cheng and Church, 2000), OPSM (Ben-Dor et al., 2002), ISA (Bergmann et al., 2004) and Bimax (Prelic et al., 2006). For these reference methods, we use *Biclustering Analysis Toolbox* (BicAT) which is a recent software platform for clustering-based data analysis that integrates all these biclustering algorithms (Barkow et al., 2006). We also compare our method with two additional methods (Samba (Tanay et al., 2002) and RMSBE (Liu and Wang, 2007)).

For the experiments, we empirically fix α, β and $threshold\_ASR$ of the MBA algorithm as follows. We experiment a number of combinations (typically several tens) and for each combination, we compute the p-values of the obtained biclusters. We pick the combination with the lowest p-value for the final experiment. For CC, OPSM, ISA and Bimax, the default values used in (Liu and Wang, 2007) are adopted for the Yeast Cell-Cycle dataset. For all the other experiments, we report the results of the compared algorithms from their original papers. The MBA algorithm was coded in Java and run on a Intel Core 2 Duo T6400 PC with 2.0GHz CPU and 3.5Gb RAM.

Figure 9 shows two selected biclusters obtained by the proposed algorithm for Saccharomyces cerevisiae dataset and Yeast Cell-Cycle dataset respectively. From the figure, we can observe that the negative correlated genes are well captured by our algorithm.

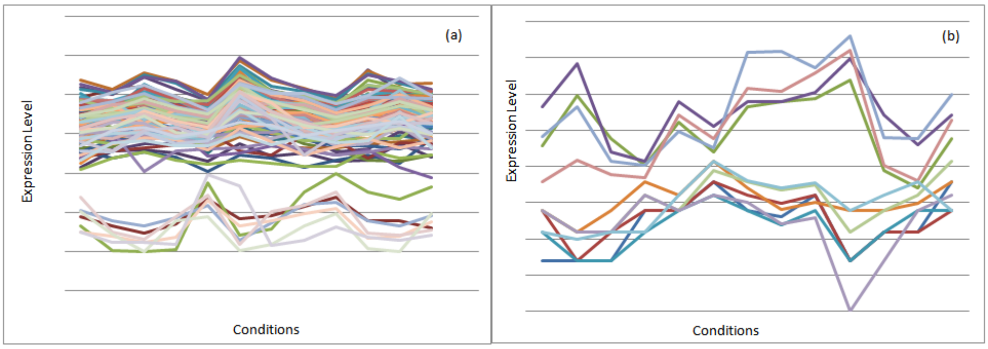


Figure 9: Two biclusters contain negatively correlated genes which show opposite patterns. These biclusters were obtained from Saccharomyces cerevisiae dataset (a) and Yeast Cell-Cycle dataset (b)

4.1. Saccharomyces cerevisiae dataset

The Saccharomyces cerevisiae dataset (available at http://www.tik.ethz.ch/sop/bimax/) (Gasch et al., 2000) contains the expression levels of 2993 genes under 173 experimental conditions. For this experiment, the parameters of MBA are experimentally set as follows: $\alpha = 0.7$, $\beta = 0.7$, $threshold\_ASR$=0.65, Y=50 and Z=100.

The results of MBA are compared against the reported scores of RMSBE, *Bimax*, OPSM, ISA, Samba and CC from (Liu and Wang, 2007; Prelic et al., 2006). In order to evaluate the statistical significance, we determine whether the set of genes contained in the bicluster shows significant enrichment with respect to a specific *Gene Ontology* (GO). We use the webtool *FuncAssociate* (available at http://llama.mshri.on.ca/funcassociate/) (Berriz et al., 2003) for this purpose. *FuncAssociate* computes the adjusted significance scores for each bicluster, i.e, adjusted p-values (p=5%, 1%, 0.5%, 0.1% and 0.001%) which is the one-sided p-value of the association between attribute and query resulting from Fisher's Exact Test. The best biclusters have an adjusted p-value less than 0.001%.

Figure 10 presents different significant scores p for each algorithm over the percentage of total extracted biclusters. On the one hand, MBA and RMSBE outperform other algorithms. MBA (resp. RMSBE) results show that 97% (resp. 98%) of discovered biclusters are statistically significant with $p < 0.001\%$. On the other hand, apart from CC, the other algorithms have reasonably good performance. In particular, OPSM is the best of the other compared algorithms: 87% of its biclusters has $p < 0.001\%$. CC underperforms because it is unable to find coherent biclusters and its lack of robustness against noise.

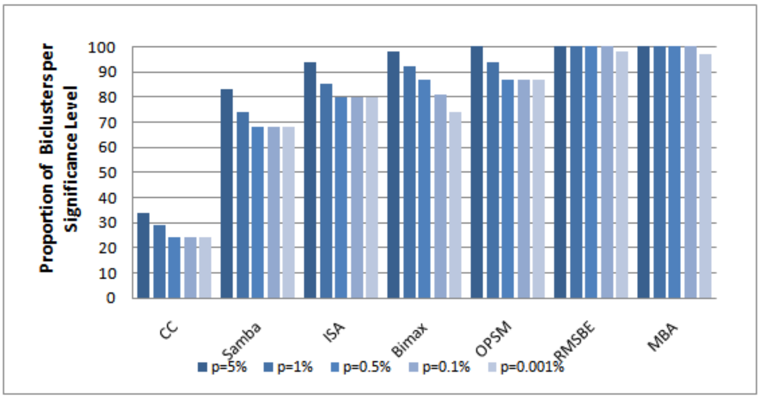


Figure 10: Proportions of biclusters significantly enriched by GO on Saccharomyces cerevisiae dataset.

Yeast Cell-Cycle dataset

The *Yeast cell-cycle* dataset (available at http ://arep.med.harvard.edu/ biclustering/) is described in (Tavazoie et al., 1999). This dataset is processed in (Cheng and Church, 2000) and publicly available from (Cheng and Church, 2006). It contains the expression profiles of more than 6000 yeast genes measured at 17 conditions over two complete cell cycles. In our experiments we use 2884 genes selected by (Cheng and Church, 2000).

For this dataset, two criteria are used. First, we evaluate the statistical relevance of the extracted biclusters by computing the adjusted p-value like as for the Saccharomyces cerevisiae dataset. Second, we identify the biological annotations for the obtained biclusters. For this experiment, the parameters of MBA are set as follows: α=0.5, β=0.6, $threshold\_ASR$=0.6, Y=50 and Z=100.

Statistical relevance: To evaluate the statistical relevance of MBA, we use again the p-values and apply the web-tool *FuncAssociate* (Berriz et al., 2003). The results of MBA are compared against CC, ISA, *Bimax* and OPSM. Figure 11 shows, for each significant score p (p=5%, 1%, 0.5%, 0.1% and 0.001%) and for each compared algorithm, the percentage of the statistically significant biclusters extracted by the algorithm with the indicated p-value. We observe that MBA outperforms the other algorithms on this dataset. 93% of discovered biclusters of MBA are statistically significant with $p < 0.001\%$. On the other hand, the best of the compared algorithm (*Bimax*) has only a percentage of 64% for $p < 0.001\%$.

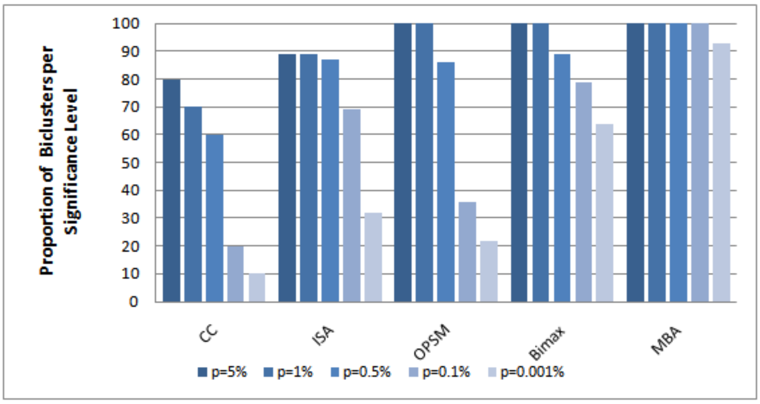


Figure 11: Proportions of biclusters significantly enriched by GO on Yeast Cell-Cycle dataset.

Analysis of biological annotation enrichment of biclusters: To evaluate the biological significance of the obtained biclusters in terms of the associated biological processes, molecular functions and cellular components respectively, we use the *Gene Ontology* (GO) term finder *GOTermFinder* (available at http://db.yeastgenome.org/cgi-bin/GO/goTermFinder). Indeed, the GO project provides a controlled vocabulary to describe gene and gene product attributes in any organism, and it is a collaborative effort to address the need for consistent descriptions of gene products in different databases (cited from www.geneontology.org). *GOTermFinder* can find the significant shared GO terms for genes within the same bicluster.

Table 1: Most significant shared GO terms (process, function, component) for two biclusters on yeast cell-cycle dataset.

Biclusters	Biological Process	Molecular function	Cellular component
54 genes × 6 conditions	maturation of SSU-rRNA (52.7%, 4.5e-37) ribosome biogenesis (38.4%, 22.5e-13) maturation of SSU-rRNA from tricistronic rRNA transcript (SSU-rRNA, 5.8S rRNA, LSU-rRNA) (9.3%, 2.32e-11)	structural molecule activity (42.6%, 9.54e-35)	cytosolic part (23.6%, 6.86e-56) ribosome (31.5%, 8.41e-33)
11 genes × 13 conditions	DNA strand elongation (6.54% , 9.35e-21) DNA strand elongation during DNA replication (7.9% , 8.43e-06)	structure specific DNA binding (3.9% , 0.000183) structural constituent of ribosome (17.43%, 0.00563)	nuclear replication fork (5.7% , 5.34e-19) non-membrane-bounded organelle (56.9%, 6.73e-13)

We present the significant shared GO terms (or parent of GO terms) used to describe two selected sets of genes extracted by MBA with 54 genes × 6 conditions and 11 genes × 13 conditions respectively. We report the GO most significant terms shared by these biclusters in terms of biological process, molecular function and cellular component. The values within parentheses after each GO term in Table 1, such as (52.7%, 4.5e-37) in the first bicluster, indicate the cluster frequency and the statistical significance. The cluster frequency (52.7%) shows that out of 54 genes in the first bicluster 29 belong to this process, and the statistical significance is provided by a p-value of 4.5e-37 (highly significant).

5. Conclusion

In this paper, we have proposed a novel memetic algorithm, called MBA, for discovering negative correlated genes of microarrays data. MBA operates on a set of candidate biclusters and uses these biclusters to create new solutions by applying variation operators such as combinations and local improvements. By using a behavior matrix representation of solutions, the local improvement is guided by a positive and negative pattern-based neighborhood which is defined by three move operators. These operators change respectively the rows and columns of the current solution according to the type of pattern information related to each row and each column of the current solution as well as the initial matrix. The performances of the MBA algorithm is assessed on two well-known DNA microarray datasets. Computational experiments show highly competitive results of MBA in comparison with other popular biclustering algorithms by providing statistically and biologically significant biclusters. MBA is a computationally effective method and can also be used to improve biclusters obtained by other methods by adding negative correlation.

Acknowledgment

We are grateful to the reviewers for their useful comments. The work was partially supported by the ”Pays de la Loire” Region (France) within the Griote project (2013-2018).

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

Angiulli, F., Cesario, E., Pizzuti, C., 2008. Random walk biclustering for microarray data. Journal of Information Sciences, 1479–1497.

Ayadi, W., Elloumi, M., , Hao, J. K., 2012a. Bimine+: An efficient algorithm for discovering relevant biclusters of dna microarray data. Knowledge Based Systems Journal 15 (4), 224–34.

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

Ayadi, W., Elloumi, M., Hao, J. K., 2012b. Bicfinder: a biclustering algorithm for microarray data analysis. Knowledge and Information Systems: An International Journal 30, 341–358.

Ayadi, W., Elloumi, M., Hao, J. K., 2012c. Pattern-driven neighborhood search for biclustering of microarray data. BMC Bioinformatics 13(S-7): S11.

Ayadi, W., Elloumi, M., Hao, J. K., 2014. Microarray Image and Data Analysis: Theory and Practice, L. Rueda (Ed.). CRC Press Taylor & Francis, Ch. Systematic and Stochastic biclustering algorithms for microarray data analysis, pp. 321–345.

Bar-Joseph, Z., 2004. Analyzing time series gene expression data. Bioinformatics 20(16), 2493–2503.

Barkow, S., Bleuler, S., Prelic, A., Zimmermann, P., Zitzler, E., 2006. Bicat: a biclustering analysis toolbox. Bioinformatics 22(10), 1282–1283.

Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z., 2002. Discovering local structure in gene expression data: the order-preserving submatrix problem. In: Proceedings of the sixth annual international conference on Computational biology. ACM, New York, NY, USA, pp. 49–57.

Bergmann, S., Ihmels, J., Barkai., N., 2004. Defining transcription modules using large-scale gene expression data. Bioinformatics 20(13), 1993–2003.

Berriz, G., King, O., Bryant, B., Sander, C., Roth, F., 2003. Characterizing gene sets with funcassociate. Bioinformatics 19 (18), 2502–2504.

Bleuler, S., Prelic, A., Zitzler, E., 2004. An ea framework for biclustering of gene expression data. In: Proceedings of Congress on Evolutionary Computation. pp. 166–173.

Bryan, K., Cunningham, P., Bolshakova, N., 2006. Application of simulated annealing to the biclustering of gene expression data. In: IEEE Transactions on Information Technology on Biomedicine, 10(3). pp. 519–525.

Cheng, K., Law, N., Siu, W., Liew, A., 2008. Identification of coherent patterns in gene expression data using an efficient biclustering algorithm and parallel coordinate visualization. BMC Bioinformatics 9(210), 1282–1283.

Cheng, Y., Church, G., 2006. Biclustering of expression data. Technical report, (supplementary information).

Cheng, Y., Church, G. M., 2000. Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology. AAAI Press, pp. 93–103.

Coello, C. A. C., Lamont, G. B., Veldhuizen, D. A. V., 2002. Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation). 2 edition, Secaucus, NJ, USA.

Das, S., Idicula, S., 2010. Application of reactive grasp to the biclustering of gene expression data. In: Proceedings of the International Symposium on Biocomputing. ACM, New York, NY, USA, pp. 1–8.

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

Divina, F., Aguilar-Ruiz., J., 2007. A multi-objective approach to discover biclusters in microarray data. In: Proceedings of the 9th annual conference on Genetic and evolutionary computation. ACM, New York, NY, USA, pp. 385–392.

Gallo, C., Carballido, J., Ponzoni, I., 2009. Microarray biclustering: A novel memetic approach based on the pisa platform. In: Proceedings of the 7th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics. Springer-Verlag, Berlin, Heidelberg, pp. 44–55.

Gasch, A., Spellman, P., Kao, C., Carmel-Harel, O., Eisen, M., Storz, G., Botstein, D., Brown, P., 2000. Genomic expression programs in the response of yeast cells to environmental changes. Molecular Biology of the Cell, 11(12), 4241–4257.

Han, L., Yan, H., 2012. Hybrid method for the analysis of time series gene expression data. Knowledge-Based Systems, 35, 14-20, 2012 November 2012, Pages 1420

Hanczar, B., Nadif, M., 2011. Using the bagging approach for biclustering of gene expression data. Neurocomputing 74 (10), 1595–1605.

Hao, J. K., 2012. Memetic algorithms in discrete optimization. In: Handbook of Memetic Algorithms. Studies in Computational Intelligence 379, Chapter 6. Springer, pp. 73–94.

Hartigan, J. A., 1972. Direct clustering of a data matrix. Journal of the American Statistical Association 67(337), 123–129.

Lehmann, E., D'Abrera, H., 1998. Nonparametrics: Statistical Methods Based on Ranks. Englewood Cliffs, NJ: Prentice-Hall, pp. 292–323.

Liu, J., Wang, W., 2003. Op-cluster: Clustering by tendency in high dimensional space. IEEE International Conference on Data Mining, 187–194.

Liu, X., Wang, L., 2007. Computing the maximum similarity bi-clusters of gene expression data. Bioinformatics 23(1), 50–56.

Madeira, S. C., Oliveira, A. L., 2004. Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics 1 (1), 24–45.

Mitra, S., Banka, H., 2006. Multi-objective evolutionary biclustering of gene expression data. Journal of Pattern Recognition, 2464–2477.

Moscato P., 1999. A gentle introduction to memetic algorithms. In David W. Corne D.W., Dorigo M., Glover F. (eds.), New Ideas in Optimization, McGraw-Hill Ltd., UK Maidenhead, UK, England, pp. 219-234.

Pontes, B., Divina, F., Giráldez, R., Aguilar-Ruiz, J., 2007. Virtual error: A new measure for evolutionary biclustering. In: Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics. pp. 217–226.

Prelic, A., Bleuler, S., Zimmermann, P., 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.

Schmid, M., Davison, T., Henz, S., Pape, U., Demar, M., Vingron, M., Scholkopf, B., Weigel, D., Lohmann, J., 2005. A gene expression map of arabidopsis thaliana development. Nature Genetics 37, 501–06.

Tanay, A., Sharan, R., Shamir, R., 2002. Discovering statistically significant biclusters in gene expression data. Bioinformatics 18, S136–S144.

Tavazoie, S., Hughes, J. D., Campbell, M. J., Cho, R. J., Church, G. M., 1999. Systematic determination of genetic network architecture. Nature Genetics 22, 281–285.

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

Valente-Freitas, A., Ayadi, W., Elloumi, M., Oliveira, J. L., Hao, J. K., 2013. Survey on biclustering of gene expression data. In: Biological Knowledge Discovery Handbook: Preprocessing, Mining and Postprocessing of Biological Data. Wiley Book Series on Bioinformatics, pp. 591–608.

Yang, J., Wang, H., Wang, W., Yu, P., 2003. Enhanced biclustering on expression data. In: Proceedings of the 3rd IEEE Symposium on BioInformatics and BioEngineering. IEEE Computer Society, Washington, DC, USA, pp. 321–327.

Zhang, Z., Teo, A., Ooi, B., Tan, K., 2004. Mining deterministic biclusters in gene expression data. IEEE International Symposium on Bioinformatic and Bioengineering, 283–290.

Zhao, Y., Yu, J. X., Wang, G., Chen, L., Wang, B., Yu, G., 2008. Maximal subspace coregulated gene clustering. IEEE Transactions on Knowledge and Data Engineering 20 (1), 83–98.