Evolutionary Algorithm Based on New Crossover for the Biclustering of Gene Expression Data

Ons Maâtouk[1,2], Wassim Ayadi[2,3], Hend Bouziri[1], and Beatrice Duval[2]

[1] LARODEC Laboratory, ISG Tunis, Université de Tunis, Tunisia
[2] LERIA, Université d'Angers, 2 Boulevard Lavoisier, 49045 Angers, France
[3] LaTICE Laboratory, ESSTT, Université de Tunis, Tunisia
mtk-ons@hotmail.fr, {wassim.ayadi,hend.bouziri}@gmail.com, bd@info.univ-angers.fr

Abstract. Microarray represents a recent multidisciplinary technology. It measures the expression levels of several genes under different biological conditions, which allows to generate multiple data. These data can be analyzed through biclustering method to determinate groups of genes presenting a similar behavior under specific groups of conditions.

This paper proposes a new evolutionary algorithm based on a new crossover method, dedicated to the biclustering of gene expression data. This proposed crossover method ensures the creation of new biclusters with better quality. To evaluate its performance, an experimental study was done on real microarray datasets. These experimentations show that our algorithm extracts high quality biclusters with highly correlated genes that are particularly involved in specific ontology structure.

Keywords: Biclustering, Evolutionary algorithm, Crossover method, Microarray data, Data mining.

1 Introduction

During recent years, microarray technology has reached a main role in biological and biomedical research [24]. This technology measures the expression levels of thousands of genes in different biological conditions. It allows to generate large amount of data [14]. The analysis of these data allows the extraction of biological knowledge in order to understand diseases [4]. Given the huge masses of data to be analyzed, the use of data mining techniques has become essential to extract the knowledge embedded in these masses of information. Among the clustering techniques, we can find the biclustering which has been used extensively to analyse gene expression data.

The biclustering is a data mining technique to discover high quality biclusters. These biclusters are illustrated by groups of genes presenting a similar behavior under specific groups of conditions. Formally, the biclustering problem [26] is to build a group of biclusters associated with a data matrix taking account a fitness

M. Comin et al. (Eds.): PRIB 2014, LNBI 8626, pp. 48–59, 2014.

function that measures the quality of a group of biclusters. Thus, it is highly combinatorial problem [26] and known to be NP-Hard [5].

Given the robustness to dynamic changes of the evolutionary approach and their ability to self-optimization, we adopt this approach to solve the biclustering problem. Most of the biclustering algorithms based on the evolutionary approach, like [1,8,9], use random crossover method. However, these methods do not guarantee to obtain a better quality child biclusters, that prompt us to seek a crossover method specific for the biclustering of gene expression data and allowing to have better quality biclusters.

In this work, we propose an ***E**volutionary **B**iclustering **A**lgorithm based on a new **Cross**over method* (EBACross). This new method is dedicated to the biclustering of gene expression data. EBACross uses a fast local search algorithm to generate an initial population with reasonable quality. A selection and a mutation operator are used. After an experimental study, we notice that our proposed algorithm can extract high quality biclusters with highly correlated genes which are particularly involved in specific ontology structure.

2 Description of the Proposed Biclustering Algorithm

In order to extract high-quality biclusters, we propose a new biclustering algorithm adopting the evolutionary approach. It can be summarized by 5 steps:

1. Generate the initial population P_{init}. This step is based on the Cheng and Church algorithm [5]. It is recognized for its reasonable results in a quick time and its almost total coverage of genes and conditions. It allows to start with reasonable quality biclusters covering almost all the data matrix.
2. Build the parent set P by selecting the best biclusters of the initial population P_{init}. The selection is based on four complementary fitness functions: size (f_1), *MSR* (f_2), average correlation (f_3) and coefficient of variation (f_3).
3. Create children biclusters by our proposed crossover operator that is dedicated for the biclustering of gene expression data. Based on a discretization method and the standard deviation function, this crossover combines the biclusters parents in pair giving priority to the biclusters that satisfy a maximum number of fitness functions.
4. In order to avoid overlapping biclusters and to increase the diversification of biclusters a mutation operator is used. This operator is applied to the biclusters resulting to the crossover. It is based on the average correlation function which allows to improve the coherence of gene biclusters.
5. Replace bicluster parents by those resulting to the mutation and repeat from step 2 until the reaching of the number of iterations.

2.1 Biclusters Encoding

To represent the biclusters, the majority of existing biclustering algorithms uses a fixed size binary string [17,19]. This string is built by two bit strings. The first

one represents the genes and the second represents the conditions. The string position of a gene (respectively a condition) takes 1 if the gene (respectively the condition) belongs to the bicluster, 0 otherwise. This method explores all genes and conditions. It leads to high consumption of time and memory space.

To remedy, we represent biclusters as string composed by an ordered gene and condition indices like in [7,22,1].

2.2 Selection

The selection method is applied on the initial population P_{init} to build the parent set P. This set includes the best biclusters of P_{init} according the fitness functions. To extract maximal high-quality biclusters of highly correlated genes, we can consider four main complementary fitness functions:

Size: Most of biclustering algorithm defined the size of a bicluster by its number of elements $|G| * |C|$ as in [11]. This function gives more chance to the number of genes to be maximized since the total number of genes is higher than the number of conditions. To be able to choose if we want to give more chance to the number of genes or to the number of conditions to be maximized, we define the size of biclusters by the following function where α and β are two constants.

$$f_1(Bic) = \alpha\frac{|G'|}{|G|} + \beta\frac{|C'|}{|C|} \tag{1}$$

Mean Squared Residue: Cheng and Church [5] proposed *Mean Squared Residue* (MSR) which measures the correlation of a bicluster. A high value of MSR indicates that the bicluster is weakly coherent while a low value of MSR indicates that it is highly coherent. It is defined as follows:

$$f_2(Bic) = \frac{1}{|G'|\,|C'|}\sum_{i\in G', j\in C'}(m_{ij} - m_{iC'} - m_{G'j} + m_{G'C'})^2 \tag{2}$$

where $m_{iC'}$ (respectively $m_{G'j}$) represents the expression level average of the i^{th} row (respectively the j^{th} column), $m_{G'C'}$ corresponds to the expression level average of the bicluster $Bic(G', C')$ and m_{ij} represents the expression level corresponding to the i^{th} row and the j^{th} column.

Average Correlation: Nepomuceno *et al.* [18] proposed the average correlation function to evaluate the correlation between genes in each biclusters. They indicate that the proposed function can find biclusters that cannot be found by the algorithms based on MSR. Due to these, algorithms might not find scaling patterns when the variance of gene value is high. The average correlation of the bicluster $Bic(G', C')$ is defined as follows:

$$f_3(Bic) = \frac{2}{|G'|.(|G'|-1)}\sum_{i=1}^{G'}\sum_{j=i+1}^{G'}\left|\frac{cov(g_i, g_j)}{\sigma_{g_i}\sigma_{g_j}}\right| \tag{3}$$

where $cov(g_i, g_j)$ represents the covariance of the rows corresponding to the gene g_i and the gene g_j and σ_{g_i} (respectively σ_{g_j}) corresponds to the standard deviations of the rows corresponding to the gene g_i (respectively the gene g_j).

This measure varies between 0 and 1. If the genes are highly correlated, $f_3(Bic) = 1$, 0 otherwise.

Coefficient of Variation: Statistically, the Coefficient of Variation (CV) is used to characterize the variability of the data in a sample by evaluating the percentage of variation relative to its average. The higher the value of the coefficient of variation is, the larger is the dispersion around the average. It allows to compare the variability of several samples that have different average or even which are not expressed in the same units.

By adopting it to the biclustering of microarray data, the coefficient of variation can be considered as a measure to evaluate the variability of genes of a bicluster under all its conditions. This measure is calculated separately for each bicluster and is defined as follows:

$$f_4(Bic) = \frac{\sigma_{Bic}}{m_{G'C'}} \tag{4}$$

where σ_{Bic} represents the standard deviation of the bicluster Bic and $m_{G'C'}$ corresponds to the average of all the expression levels of the bicluster Bic.

A bicluster with a high coefficient of variation is a bicluster whose the dispersion of expression levels is high.When a bicluster have a coefficient of variation equal to 0 then it has constant values.

So, the parent set P can be divided into four subsets:

- P_1: biclusters from P_{init}, with a value of f_1 higher than the threshold Th_1.
- P_2: biclusters from $P_{init} \setminus P_1$, with a value of f_2 lower than the threshold Th_2.
- P_3: biclusters from $P_{init} \setminus (P_1 \cup P_2)$, with a value of f_3 higher than the threshold Th_3.
- P_4: biclusters from $P_{init} \setminus (P_1 \cup P_2 \cup P_3)$, with a value of f_4 higher than the threshold Th_4.

2.3 Crossover

In order to obtain children biclusters with a better quality than their parent biclusters, we propose a new crossover method specific for the biclustering of gene expression data. Unlike the random crossover method used for the biclustering of gene expression data [1,22], our crossover considers the two parts of bicluster (genes and conditions part) simultaneously. It is essentially based on five steps :

Selection of the Biclusters to Combine: It consists to select the biclusters which satisfies more fitness functions to combine them together.

Let's consider the four bicluster parents : Bic_0, Bic_1, Bic_2 and Bic_3. Table 1 represents the satisfaction of the four parent biclusters to the different fitness

functions. Bic_1 satisfies all the fitness functions, Bic_2 satisfies three fitness functions while Bic_0 and Bic_3 satisfy only two. So, we start by combining the biclusters Bic_1 and Bic_2. Then, we combine the biclusters Bic_0 and Bic_3.

Table 1. Satisfaction of the biclusters to the different fitness functions

Biclusters	f_1	f_2	f_3	f_4
Bic_0	$< Th_1$	$< Th_2$	$> Th_3$	$< Th_4$
Bic_1	$= Th_1$	$< Th_2$	$> Th_3$	$> Th_4$
Bic_2	$< Th_1$	$< Th_2$	$> Th_3$	$> Th_4$
Bic_3	$< Th_1$	$> Th_2$	$< Th_3$	$= Th_4$

Creation of the Total Bicluster: This step consists to merge the sets of genes G_1 and G_2 (respectively the sets of conditions C_1 and C_2) of the two parent biclusters Bic_1 and Bic_2 into a single set G (respectively C). This allows to create a new bicluster Bic_{Tot}.

Let's consider the following bicluster parents :

$$Bic_1 = (|G_1| \times |C_1|)$$
$$Bic_2 = (|G_2| \times |C_2|)$$

where :
$G_1 = \{ g_1, g_2, \ldots, g_n \}$ and $G_2 = \{ g'_1, g'_2, \ldots, g'_m \}$ correspond respectively to the sets of genes of the two parent biclusters Bic_1 and Bic_2.
$C_1 = \{ c_1, c_2, \ldots, c_p \}$ and $C_2 = \{ c'_1, c'_2, \ldots, c'_q \}$ correspond respectively to the sets of conditions of the two parent biclusters Bic_1 and Bic_2.

The merge of these two biclusters gives a bicluster $Bic_{Tot} = (|G| \times |C|)$ where $G = G_1 \cup G_2$ corresponds to the set of genes and $C = C_1 \cup C_2$ corresponds to the set of conditions.

Discretization of the Total Bicluster: To cluster the conditions with similar expression levels for each gene, a discretization method is applied independently for each one. This requires the decomposition of the total bicluster into several vectors. Each vector represents the expression levels of a specific gene under all conditions of total bicluster. The discretization method is based on the *Standard Deviation* (SD) to determine whether conditions can belong to the same cluster. It is a statistical measure to evaluate the dispersion of a value around the average. This measure is defined as follows:

$$SD_C = \sqrt{\frac{1}{t-1} \sum_{i=1}^{t} (a_i - \bar{C})} \tag{5}$$

where $C = \{c_1, c_2, ..., c_t\}$ represents a set of t conditions, a_i represents the expression level of the i^{th} condition and $\bar{C}$ represents the average of the expression levels of the set C, for a specific gene.

The discretization method can be summarized by the following steps:

1. Sort the set C, according to their expression levels in ascending order, to construct a new conditions set $C' = \{c'_1, c'_2, ..., c'_t\}$. This part is important. It ensures a better clustering and optimal clusters for the next part.
2. Cluster the conditions of the set C' based on the standard deviation. First, calculate the standard deviation SD_C of the vector. Then, check whether the cluster Cl is empty and browse the conditions one by one.
3. If $Cl = \emptyset$, add the condition c'_j to Cl and return to the previous step. Else, add also the condition c'_j to Cl and calculate its standard deviation SD_{Cl}.
4. If $SD_{Cl} > SD_C$ or $SD_{Cl} > SD_{Old}$ (SD_{Old} standard deviation of Cl before adding the last condition c'_j), assign c'_j to the next cluster and return to the step 2. Otherwise, to be sure that the condition c'_j is closer to the conditions of Cl than the condition following c'_{j+1}, calculate the standard deviation SD_{Next} of c'_j and c'_{j+1}.
5. If ($SD_{Cl} <= SD_{Next}$), assign the condition c'_j to the next cluster and repeat all the steps. Otherwise, let the condition c'_j in this cluster Cl, return to the step 2 and repeat with the condition c'_{j+1}.

These steps are repeated for each vector. Once complete, return vectors in their original order. Then, bring them together to construct the discretized matrix $Disc$. The cell of this matrix D_{ij} indicates the index of the cluster to which the j^{th} condition belongs for the i^{th} gene.

Construction of the Variation Matrix: Based on the discretized matrix, we build a new matrix. This matrix shows the variation of genes between each pair of conditions. Columns represent genes and rows represent the pair of conditions $\{(c_1\text{-}c_2), (c_1\text{-}c_3), (c_1\text{-}c_3) ..., (c_{t-1}\text{-}c_t)\}$. The cells of the matrix V_{ij} can take only three values:

- If $D_{ia} > D_{ib} : V_{ij} =$ -1 with $a \leq t$, $b \leq t$ and $a < b$
  For the i^{th} gene, the index of the cluster to which belongs the condition a is higher than the cluster to which belongs condition b.
- If $D_{ia} = D_{ib} : V_{ij} = 0$ with $a \leq t$, $b \leq t$ and $a < b$
  For the i_{th} gene, both conditions a and b belong to the same cluster.
- If $D_{ia} < D_{ib} : V_{ij} = 1$ with $a \leq t$, $b \leq t$ and $a < b$
  For the i^{th} gene, the index of the cluster to which belongs the condition a is lower than the cluster to which belongs condition b.

Search of Children Bicluster: The last step allows to extract the biclusters by browsing variation matrix and selecting genes having the same index.

Let's consider the example in Table 2. First, check if there are other genes with the same index as the gene g_0 for the pair of conditions (c_0-c_1). Only the gene g_3 is found. Now, check if these two genes g_0 and g_3 have the same index for other pairs of conditions. In this example, the genes g_0 and g_3 have the same index for all pairs of conditions. So, the first child bicluster contains the genes g_0, g_3 and the conditions c_0, c_1, c_2, c_3 ($Child_1$: 0 3 // 0 1 2 3).

Then, do the same steps with the gene g_1. The index of the gene is different from all other genes for the conditions, as well as for the gene g_2. Therefore, back to the gene g_0 for the pair of conditions (c_0-c_2) and check if there are any other gene with the same index. Only the gene g_3 is found. It is the case of the first child. Thus, ignore it and go to the next gene. So on, until finding all children biclusters. To avoid overlapping biclusters, we used the *Jaccard index* [13]. This index measures the overlap between two biclusters in terms of genes and conditions.

Table 2. Example of variation matrix

	c_0-c_1	c_0-c_2	c_0-c_3	c_0-c_4	c_1-c_2	c_1-c_3	c_1-c_4	c_2-c_3	c_2-c_4	c_3-c_4
g_0	-1	-1	0	1	0	1	1	1	1	1
g_1	1	0	0	1	-1	-1	0	0	1	1
g_2	0	1	1	1	1	1	1	0	1	1
g_3	-1	-1	0	1	0	1	1	1	1	1

This crossover method allows to create children biclusters with a better quality. The use of standard deviation to discretize parent biclusters allows to group closest expression levels for each gene and to construct the variation matrix. This matrix indicates the variation of the expression levels between each pair of conditions, which allows to determinate the genes presenting a similar behavior and to extract biclusters with highly correlated genes.

2.4 Mutation

In order to ensure the diversification of biclusters and to improve their quality, a mutation method is applied. It tries to improve the coherence between the genes of the biclusters obtained from the crossover, using a correlation matrix. This genetic operator seeks the less coherent gene in the bicluster. Then, it replaces this less coherent gene by the most coherent gene which does not belong to the bicluster.

To construct the correlation matrix, we must calculate the correlation coefficient between each pair of genes. Then, depending on the value obtained, a value is assigned to the cell C_{ij} corresponding to the correlation between the gene g_i and the gene g_j. The cell C_{ij} can take only three values: C_{ij} = -1, if $i = j$. Otherwise, $C_{ij} = 0$ when $\left|\rho_{(g_i,g_j)}\right| < Th_{Corr}$ and $C_{ij} = 1$, when $\left|\rho_{(g_i,g_j)}\right| \geq Th_{Corr}$.

3 Experimental Results

In order to test the performance of our proposed algorithm and analyze its results, a series of experiments is performed on real gene expression datasets: *Yeast cell cycle* [25] and *Saccharomyces cerevisiae* [10]. The evaluation of biclustering

algorithms and its comparison are based on two complementary criteria: statistical criteria and biological criteria. We compare the results of EBACross with other sate-of-the-art biclustering algorithms ISA [3], BiMax [20], CC [5], OPSM [2], X-Motif [16] and the evolutionary algorithm EvoBic [1], H-MOBI [22], SEBI [8].

3.1 Statistical results

To measure the quality of resulting biclusters, we use the functions "size", "average correlation", "MSR" like in [1,18,22]. We calculate the "coverage" and we proceed as in [9,11,15]. This criterion is defined as being the total number of cells of the matrix M covered by the resulting biclusters.

Table 3. Comparing the fitness function values and the coverage of different biclustering algorithms for the *Yeast Cell Cycle* and *Saccharomyces cerevisiae* datasets

Yeast Cell Cycle

	EvoBic	H-MOBI	SEBI	BiMax	CC	ISA	OPSM	X-Motif	EBACross
Gene number	788,4	1610,8	—	24,0	39,62	76,3	437,94	1,2	38,08
Condition number	3,3	7,87	—	3	3,16	8,7	9,5	11,4	3,78
Average correlation	0,90	—	—	0,66	0,84	0,50	0,91	0,71	0,82
MSR	291	297	205,18	209,5	10,94	248,25	288,04	203,14	167,62
Genes coverage	99,58	—	13,61	79,09	61,79	73,44	83,98	52,86	66,85
Conditions coverage	70,59	—	15,25	64,71	100	100	100	100	100
Total coverage	44,21	—	—	46,48	10,75	38,94	18,67	25,36	49,53

Saccharomyces cerevisiae

	EvoBic	H-MOBI	SEBI	BiMax	CC	ISA	OPSM	X-Motif	EBACross
Gene number	17,8	—	—	32,8	81,11	76,27	95,58	1,12	41,46
Condition number	3	—	—	3	19,64	8,71	12,5	34,52	4,20
Average correlation	0,90	—	—	0,68	0,33	0,59	0,87	0,97	0,81
MSR	0,08	—	—	0,18	0,36	0,22	0,08	10^{-17}	0,25
Genes coverage	4,84	—	—	29,54	96,06	34,08	13,79	10,89	85,77
Conditions coverage	7,51	—	—	79,19	100	58,38	26,01	100	84,39
Total coverage	0,09	—	—	0,99	49,09	2,25	0,96	2,62	10,12

Table 3 presents the average value of the gene number, the condition number, the average correlation, the MSR and the coverage of the obtained biclusters for the *Yeast Cell Cycle* and *Saccharomyces cerevisiae* datasets.

We can show that most algorithms have relatively close results. For the *Yeast Cell Cycle*, the best MSR value is obtained by CC (MSR = 10,94) and the best average correlation value is obtained by OPSM (ρ = 0,91) while for the *Saccharomyces cerevisiae* dataset, the best MSR and average correlation value are obtained by X-Motif (MSR = 10^{-17} and ρ = 0,97). Although the results of our proposed algorithm are not the best, they have a satisfactory quality and are consistent. Indeed, we note an average correlation value equal to 0,82 (respectively 0,81) and a MSR value equal to 167,62 (respectively 0,25) for the *Yeast Cell Cycle* (respectively the *Saccharomyces cerevisiae*) dataset.

Concerning the percentage of cells in the initial matrix covered by the different biclustering, we can show that most algorithms have relatively close results.

However, our algorithm has the best percentage for the genes coverage, conditions coverage and total coverage. Indeed, for the *Yeast Cell Cycle* (respectively the *Saccharomyces cerevisiae*) dataset, the biclusters generated by our algorithm cover 66,85% (respectively 85,77%) of the genes, 100% (respectively 84,39%) of the conditions and 49,53% (respectively 10,12%) of the cells of the initial matrix.

3.2 Biological Results

The biological criteria allows to measure the quality of resulting biclusters, by checking whether the genes of a bicluster have common biological characteristics. For that, we calculate the p-value. The biclusters with a p-value p lower than 5% are considered as over-represented. The most obtained biclusters have a p-value close to 0, i.e., the most genes of this bicluster have common biological characteristics.

Given the large number of the obtained biclusters, We proceed as in [8,12,20] and we test only on the one hundred best biclusters.

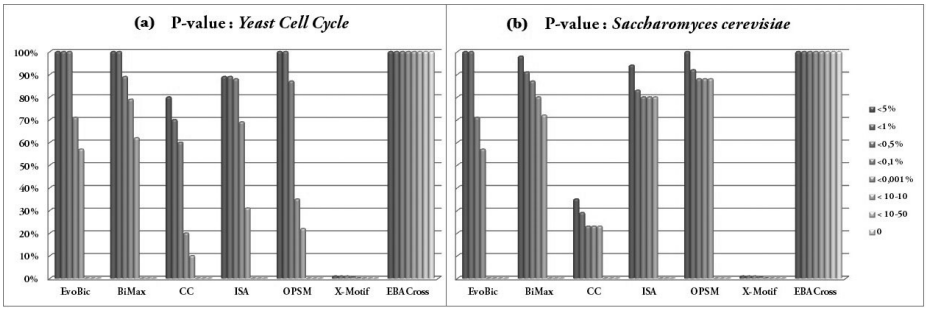

Fig. 1. P-value : *Yeast Cell Cycle* and (a) *Saccharomyces cerevisiae* dataset (b)

FIGURE 1 show the percentage of extracted biclusters for different adjusted p-value (p = 5%; 1%; 0,5%; 0,1%; 0,001%; 10^{-10}; 10^{-50} and 0), for the *Yeast Cell Cycle* and the *Saccharomyces cerevisiae* datasets.

We can show that the majority of the algorithms have rather low percentage. For the *Saccharomyces cerevisiae* dataset, 72%, 80% and 88% of the biclusters respectively extracted by Bimax, ISA and OPSM are statistically significant with a p-value $p < 0{,}001\%$. Only EBACross reaches a value less than 10^{-10}. Indeed, 100% of the biclusters extracted by our algorithm are statistically significant with a p-value equal to 0. However, for the *Yeast Cell Cycle* dataset, we notice a degradation in the results of the majority of algorithms while our algorithm maintains the quality of its results. Only 62%, 31% and 22% of the biclusters respectively extracted by Bimax, ISA and OPSM are statistically significant with a p-value $p < 0{,}001\%$ and 100% of the biclusters extracted by EBACross are statistically significant with a p-value equal to 0.

We evaluate also qualitatively the capacity of the algorithms to extract significant biclusters with a biological point of view. It requires the incorporation of biological knowledge. The biological signification of the obtained biclusters can be interpreted based on *Gene Ontology* (GO) [6] for the description of the roles of genes and their products [21]. There are three ontology structures describing the gene products: biological process, molecular function and cellular component.

Given the large number of the obtained biclusters, we proceed as in [19,23] and we present the most significant GO shared of three random biclusters extracted by our algorithm in Table 4 and Table 5, respectively for the *Yeast Cell Cycle* and *Saccharomyces cerevisiae* datasets. These tables include the gene number, condition number and the different shared GO terms for each ontology structures of the biclusters.

Table 4. GO terms of biclusters extracted by EBACross for *Yeast Cell Cycle* dataset

Biclusters	Cellular component	Molecular function	Biological process
3 genes; 9 conditions			*cellular process* (66,7%; 0)
4 genes; 7 conditions		*transferase activity* (75%; 0)	
13 genes; 5 conditions	*nucleolus* (86,7%; 9,4.10^{-32})		

Table 5. GO terms of biclusters extracted by EBACross for *Saccharomyces cerevisiae* dataset

Biclusters	Cellular component	Molecular function	Biological process
13 genes; 4 conditions			*regulation of mitotic cell cycle* (38,4%; 0)
4 genes; 5 conditions		*binding* (50%; 0)	
60 genes; 7 conditions	*intracellular* (31,67%; 5,2.10^{-6})		

We can show that the extracted biclusters are biologically relevant according to a single ontology structure and for this structure, we find only one GO term. For example, in Table 4, the genes of the first bicluster are particularly involved in the *cellular process* with a p-value $p = 0$, those of the second bicluster are particularly involved in the *transferase activity* function with a p-value $p = 0$, those of the third bicluster are particularly involved in the *nucleolus* component with a p-value $p = 9,4.10^{-6}$.

We can note that EBACross is efficient to extract signicant biclusters with specific GO term for all ontology structures (biological process, molecular function and cellular component).

4 Conclusion

In this paper, we introduce a new evolutionary algorithm. The selection operator allows to keep the best quality biclusters, based on four complementary functions. Then, a new crossover operator dedicated for the biclustering of gene expression data is used. This proposed crossover ensures the creation of new biclusters with better quality. Finally, based on the average correlation function a mutation operator seeks the least coherent gene in each bicluster to replace it with a more coherent gene.

To evaluate the performance of our algorithm, an experimental study was done on the real microarray datasets. We compare the results to other existing biclustering algorithms. These experimentations show that our algorithm EBACross allows to extract high quality biclusters with highly correlated genes. These biclusters are significant with specific GO term and their genes are particularly involved in specific ontology structure.

To refine the search mechanisms and improve the quality of the extracted biclusters in our future works, we plan to integrate biological knowledge in the research process by benefiting from the help of biologists.

References

1. Ayadi, W., Maatouk, O., Bouziri, H.: Evolutionary biclustering algorithm of gene expression data. In: The Proceedings of the 23th International Workshop on Database and Expert Systems Applications, DEXA 2012, pp. 206–210. IEEE, Vienna (2012)
2. Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering local structure in gene expression data: the order-preserving submatrix problem. In: RECOMB 2002: Proceedings of the Sixth Annual International Conference on Computational Biology, pp. 49–57 (2002)
3. Bergmann, S., Ihmels, J., Barkai, N.: Defining transcription modules using large-scale gene expression data. Bioinformatics 20(13), 1993–2003 (2004)
4. Berrer, D., Dubitzky, W., Draghici, S.: A practical approach to microarray data analysis, ch. 1, pp. 46–53. Kluwer Academic Publishers (2003)
5. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, pp. 93–103 (2000)
6. Gene Ontology Consortium. Gene ontology: tool for the unification of biology. Nature Genetics, 25, 25–29 (2000)
7. de Castro, P.A.D., de França, F.O., Ferreira, H.M., Von Zuben, F.J.: Applying biclustering to text mining: an immune-inspired approach. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 83–94. Springer, Heidelberg (2007)
8. Divina, F., Aguilar-Ruiz, J.S.: Biclustering of expression data with evolutionary computation. IEEE Transactions on Knowledge & Data Engineering 18(5), 590–602 (2006)
9. Divina, F., Aguilar-Ruiz, J.S.: A multi-objective approach to discover biclusters in microarray data. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, pp. 385–392 (2007)

10. Gasch, A.P., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D., Brown, P.O.: Genomic expression programs in the response of yeast cells to environmental changes. Molecular Biology of the Cell 11(12), 4241–4257 (2000)
11. Liu, J., Li, Z., Hu, X., Chen, Y.: Biclustering of microarray data with mospo based on crowding distance. BMC Bioinformatics 10(4), 9 (2009)
12. Liu, X., Wang, L.: Computing the maximum similarity bi-clusters of gene expression data. Bioinformatics 23(1), 50–56 (2007)
13. Madeira, S.C., Teixeira, M.C., Sá-Correia, I., Oliveira, A.L.: Identification of regulatory modules in time series gene expression data using a linear time biclustering algorithm. IEEE/ACM Transactions on Computational Biology and Bioinformatics 7(1), 153–165 (2010)
14. Martinez, R., Pasquier, N., Pasquier, C., Collard, M.: Analyse des groupes de gènes co-exprimés (AGGC): un outil automatique pour l'interprétation des expériences de biopuces. In: SFC 2006 Conference (2006)
15. Mitra, S., Banka, H.: Multi-objective evolutionary biclustering ofgene expression data. Pattern Recognition 39(12), 2464–2477 (2006)
16. Murali, T.M., Kasif, S.: Extracting conserved gene expression motifs from gene expression data. In: Pacific Symposium on Biocomputing, vol. 8, pp. 77–88 (2003)
17. Nepomuceno, J.A., Troncoso, A., Aguilar–Ruiz, J.S.: A hybrid metaheuristic for biclustering based on scatter search and genetic algorithms. In: Kadirkamanathan, V., Sanguinetti, G., Girolami, M., Niranjan, M., Noirel, J. (eds.) PRIB 2009. LNCS (LNBI), vol. 5780, pp. 199–210. Springer, Heidelberg (2009)
18. Nepomuceno, J.A., Troncoso, A., Aguilar-Ruiz, J.S.: Evolutionary metaheuristic for biclustering based on linear correlations among genes. In: SAC 2010, Switzerland, pp. 22–26 (2010)
19. Nepomuceno, J.A., Troncoso, A., Aguilar-Ruiz, J.S.: Biclustering of gene expression data by correlation-based scatter search. BioData Mining 4(3) (2011)
20. Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22, 1122–1129 (2006)
21. Robinson, P.N., Wollstein, A., Bohme, U., Beattie, B.: Ontologizing geneexpression microarray data: characterizing clusters with gene ontology. Bioinformatics 20(6), 979–981 (2004)
22. Seridi, K., Jourdan, L., Talbi, G.: Multi-objective evolutionary algorithm for biclustering in microarrays data. In: IEEE Congress of Evolutionary Computation, pp. 2593–2599 (2011)
23. Shyama, D., Sumam, M.I.: Application of greedy randomized adaptive search procedure to the biclustering of gene expression data. International Journal of Computer Applications 2(3), 0975–8887 (2010)
24. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics 18, 136–144 (2002)
25. Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., Church, G.M.: Systematic determination of genetic network architecture. Nature Genetics 22, 281–285 (1999)
26. Valente-Freitas, A., Ayadi, W., Elloumi, M., Oliveira, J.L., Hao, J.K.: A survey on biclustering of gene expression data. In: Biological Knowledge Discovery Handbook: Preprocessing, Mining, and Postprocessing of Biological Data, pp. 591–608 (2013)