BicOPT:Biochips Data Clustering algorithm

Faouzi Mhamdi[1,2] faouzi.mhamdi@ensi.rnu.tn

Ahmed Zammali[1] zammaliahmad@gmail.com,

[1]Laboratory of Technologies of Information and Communication and Electrical Engineering (LaTICE)
National Superior School of Engineers of Tunis (ENSIT), University of Tunis, Tunisia
[2]Hihger Institute of Applied Language and Computer Science of Beja, University of Jendouba, Tunisia

Abstract

Biochips present a new technology that allows to analyze the level of expression of genes, among the techniques that are applicable on this technology is the biclustering. The main objective of the latter is to extract groups of genes taking into account the coherence between all the conditions that characterize them. There are a variety of biclustering algorithms that have already been proposed in the field of biochips. Each of these algorithms differs from the others by a set of characteristics. In this paper, we focus on the BicFinder algorithm, where we propose to make improvements in order to make it faster. In the first place, we will present a fast variant of this algorithm. Then we will present our version of algorithm named BicOPT followed by a set of experiments applied to real data.

Keywords biochips, biclustering, BicFinder, BicOPT, Evaluation functions, experimental study.

1 Introduction

In a data matrix, we can find links between the set of rows or between the set of columns, or between the set of rows and columns simultaneously. A technique called clustering only allows us to detect the first and second cases. So, this technique remains too simplistic to determine the third case. Another more interesting technique, called simultaneous classification, cross-classification or block classification. It is also referred to as a biclustering [8,9] hence the objective of this approach is to extract the groups of rows while taking into account the consistency with all the columns. This technique can be used in several fields, among which we mention that of Bio-chips.

The input file for a biclustering algorithm of biochip data is a data matrix, where the rows are filled by the names of the genes and the columns are the conditions. So, let a data matrix M where n is the number of rows and m is the number of columns. A bigroupe B is a set of pairs (I, J), with I is a subset of rows of M and J is a subset of M columns, all of these subassemblies have a sub matrix called bigroupe.

The aim of the clustering algorithms is to produce a coherent, stable and homogeneous bigroup. The homogeneity criteria vary from one algorithm to another. Generally, the biclustering problem is NP-difficult. We then used heuristic algorithms to construct biclusters close to the optimal. The problem of biclustering can be formulated as following [2]:

$$f\left(B_{opt}\right) = \max f(B) \qquad (1)$$

with

- B∈BC(M)
- f is an objective function measuring the quality *i.e.,* the degree of coherence, of a group of bigroupes.
- BC(M) : is the set of all groups of possible bigroupes associated with M

Madeira et Oliveira [9] propose to classify the algorithms of biclustering according to the approaches used for their construction. These approaches are classified according to five categories [7]: IRCCC (Iterative Row and Column Clustering Combination), DC (Divide and Conquer), GIS (Greedy Iterative Search), EBE (Exhaustive Bicluster Enumeration) et DPI (Distribution Parameter Identification). BicOPT is based on the BicFinder algorithm following the Greedy Iterative Search approach of a polynomial complexity O(n^4m). So, in this paper we will present in the first place the BicFinder algorithm. In the second place, we will detail our BicOPT contributions and we will pass to the illustrations of the experimental study of our approach, we will end with a conclusion.

2 BicFinder

BicFinder is a systematic greedy algorithm, its polynomial complexity is equal to O(n^5m), based on the construction of an acyclic directed graph (DAG). BicFinder allows to extract and produce a set of bigroupes close to what a biologist can do by looking for the maximum homogeneous zones. The stage of generation of bigroupes passes through 4 essential steps first of all the discretization of matrix M in M' (see equation 1), then the construction of DAG from M', then the extraction by applying the function ACSI (see equation 2) and validation using the ASR function (see equation 3).

Algorithm 1. BicFinder [1]

```
1: Input: M, α, β ; Output: B
2: Discretize M using Equation 7 to obtain M'// Step of discretization
3: Build the DAG associated with M'// Construction Step
4: B = Ø // Extraction step
5: For any $n_i$ in the DAG do
6: $I'_i$=Ø; $J'_i$=Ø; // Bi = ($I'_i$ , $J'_i$)
7: Sort arcs of $n_i$ in decreasing order according to the number of true
8: For any edge ($n_i$,$n^k$) do
9: Ic=$I'_i$ U {$g_i$,gk}; Jc=$J'_i$ ∪ {cl,cl+1 with T(M′[i, l] = M′[k, l]) = true};
10: If $ACSI_i$(Ic, Jc) >= α then $B_i$ = (Ic, Jc)
11: End
12: B = B U Bi
13: End
14: For any bigroupe Bi = (I′i , J′i) in B do // Selection step
15: If ASR(I′i , J′i) < β then B = B\Bi
16: End
17: Return B
```

Group extraction processes are subdivided into four main steps (see Figure 1).

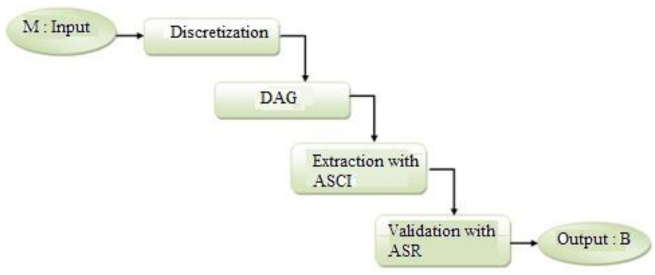


Figure 1.BicFinder algorithm process

2.1 Discretization

To compute ACSI, we must first discretize the initial matrix M (I, J), I = {1, 2, ..., n} and J = {1, 2, ..., m} Matrix M '(see equation 7).

$$M'[i,l]\begin{cases}1\ if\ M[i,l] < M[i,l+1]\\-1\ if\ M[i,l] > M[i,l+1]\\0\ if\ M[i,l] = M[i,l+1]\end{cases} \quad (2)$$

With i$\in[1,n]$ and l$\in[1..m-1]$

The discretization allows us to know the shape of the gene expression profile (which can be either monotonically increasing or monotonically decreasing ...).

2.2 Construction of DAG

Our graph is associated with the matrix M ', where each node n_i has a gene g_i. Two nodes n_i and n_j are connected by an arc if and only if (i> j). CSI $_{i,j}$ is assigned for each arc (n_i, n_j).

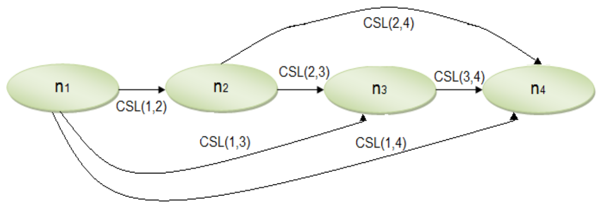


Figure 2. Example of DAG

2.3 Extraction: ACSI

Is a extraction function based on Concordance Index (CI) [12]. To calculate ACSI, the CSI function must be calculated for each arc of the graph (Dag) (see equation 3).

$$CSI(i,j,k) = \frac{\sum_{i=1}^{m-1} T(M'[i,l] = M'[j,l] = M'[k,l])}{\text{MaxCSL}_i} \quad (3)$$

with i$\in$ $[1..n-2]$, j$\in[2..n-1]$, k$\in[3..n]$, 1$\in[1..m-1]$and $i<j<k$

$$ACSI_i(I',J') = 2 * \frac{\sum_{j\in I;j\geq i+1}\sum_{k\in I;k\geq i+1} CSI(i,j,k)}{|I''|(|I''|-1)} \quad (4)$$

Our bigroup starts with an initial arc (MaxCSL $_{i,j}$) and at each iteration we add an arc if and only if $ACSI_i$ (I ', J')> = α otherwise we pass to the next arc.

2.4 Evaluation: ASR

The last step used is the evaluation of bigroupes generated by applying ASR function.

$$ASR(I',J') = 2\max\left\{\frac{\sum_{i\in I'}\sum_{j\in I';j\geq i+1} pij}{|I'|(|I'|-1)}, \frac{\sum_{k\in J'}\sum_{l\in J';l\geq k+1} pij}{|J'|(|J'|-1)}\right\} \quad (5)$$

with
$$p_{ij} = 1 - \frac{6\sum_{k=1}^{m}(r_k^i\,(x_k^i) - r_k^j\,(x_k^j))^2}{m(m^2\ -1)} \quad (6)$$

A bigroup is valid if its ASR> = β.

2.5 Clustering: K-medoids

After the presentation of the algorithm and the explanation of its operating principle, we describe, in this section, the BicFinder process using an illustrative example.
So, we fix the parameter α which controls the extraction and addition of the arc and the parameter β which controls the validation of bigroupes. Let the parameters α = 0.75, β = 0.5.

Table I. Data Matrix M

	C0	C1	C2	C3	C4	C5
g0	13	7	5	20	10	-5
g1	15	10	20	30	-2	15
g2	15	9	8	20	10	10
g3	3	8	10	9	15	4
g4	13	15	17	8	3	1
g5	20	8	12	25	27	1
g6	13	15	17	8	3	1

Table II. Matrix M' after discretization

	C0	C1	C2	C3	C4
g0	-1	-1	1	-1	-1
g1	-1	1	1	-1	1
g2	-1	-1	1	-1	0
g3	1	1	-1	1	-1
g4	1	1	-1	-1	-1
g5	-1	1	1	1	-1
g6	1	1	-1	-1	-1

The DAG is constructed from the matrix M '. The arcs are sorted in decreasing order relative to the weight associated with each edge (with the weight equal to the sum of true).

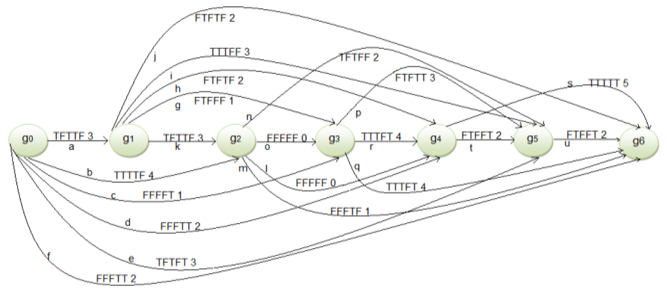


Figure 3. DAG associated with the matrix M '

For the first node g0 we have CSL (g0) = {(b), (a), (e), (d), (f), (c)}. So we take the first two arcs "b" and "a"

$ACSIg0\ (b,a) = \frac{\text{CSI}(0,1,2)}{2(2-1)/2} = \frac{3/4}{1} = 0.75$ We have ACSIg0 (b, a) = α so we add the arc "e"

$$ACSIg0\ (b,a,e) = \frac{\text{CSI}(0,1,2) + \text{CSI}(0,1,5) + \text{CSI}(0,2,5)}{3(3-1)/2} = \frac{\frac{3}{4}+\frac{2}{4}+\frac{2}{4}}{3} = 0.58 < \alpha$$

$$ACSIg0\ (b,a,d) = \frac{\text{CSI}(0,1,2) + \text{CSI}(0,1,4) + \text{CSI}(0,2,4)}{3(3-1)/2} = \frac{\frac{3}{4}+\frac{1}{4}+\frac{1}{4}}{3} = 0.41 < \alpha$$

$$ACSIg0\ (b,a,f) = \frac{\text{CSI}(0,1,2) + \text{CSI}(0,1,6) + \text{CSI}(0,2,6)}{3(3-1)/2} = \frac{\frac{3}{4}+\frac{1}{4}+\frac{1}{4}}{3} = 0.41 < \alpha$$

$$ACSIg0\ (b,a,c) = \frac{\text{CSI}(0,1,2) + \text{CSI}(0,1,3) + \text{CSI}(0,2,3)}{3(3-1)/2} = \frac{\frac{3}{4}}{3} = 0.25 < \alpha$$

We apply the same processes on the rest of the nodes and we obtain as a result: B= {($\{g0, g1, g2\};\ \{c'0, c'1, c'2, c'3, c'4\}$) ; ($\{g3, g4, g6\};\ \{c'0, c'1, c'2, c'3, c'4, c'5\}$)}. Only the bigroups who have a score ASR >= β Will be selected. $ASR(\{g0, g1, g2\};\ \{c'0, c'1, c'2, c'3, c'4\}) >$ β and $ASR(\{g3, g4, g6\};\ \{c'0, c'1, c'2, c'3, c'4, c'5\})<$ β. Finally, we obtain: B= {($\{g0, g1, g2\};\ \{c'0, c'1, c'2, c'3, c'4\}$)}.

3 BicOPT

The BicFinder algorithm has shown better performance compared to other bicluster algorithms [1]. The results obtained prompted us to study and improve this algorithm.

3.1 Optimization

The BicFinder algorithm resulted in better performance compared to other bicluster algorithms [1]. These results present a motivation for us to study and improve this algorithm.

3.1.1 Main Program

The temporal complexity of the extraction step is O (n^5,m) [1], which is rather complex. The second and third equations show that for a single node the minimum complexity time for the extraction step is O (n^2m) but we need to browse the whole data file so we have as a time of minimal complexity O (n^3m). Our main algorithm is divided into five steps (see algorithm 2):

- Discretization
- Construction of DAG
- Extraction of bigroupes
- Evaluation of bigroupes
- Results Visualization

Algorithm 2. Main program

```
1: F : File // Initial file
2: M : Integer // Number of columns
3: N : Integer // Number of rows
4: α, β,Ɣ : Integer // parameter
5: Mat : matrix // Matrix after discretization
6: T : Table // table of DAG
7: Begin
8: Mat =Discretization(F) ;
9:T=DagTree(Mat) ;
10:B=Extraction(T,Mat) ;
11:B=Evaluation(B);
12: Visualization(B);
13: End
```

3.1.2 Function "Extraction"

The extraction function uses equation 3. We have already mentioned that this equation has a minimum complexity time which is equal to O (n^3m), so we tried to implement this equation with a time of complexity less than O (n^5m) (See algorithm 3.).

Algorithm 3. Extraction(T ,Mat)

```
1: A,B,C,Arc : table
2: tab : table //Contains the detected arcs for a bicluster
3: nbrLine: integer //Number of rows for a bicluster
4: Begin
5: For i :=0 to n-2 do // browse all the lines of the input file
6:     Arc[]:= extractionEntier(T[i],',') ; // Arc[] : Table contains all the arcs of a selected node
7:     nbrLine:=0 ;
8:     Tab[0]:=arc[0] ;
9:     int j:=1 ;
10:    While (j< arc.length et x<=Ɣ) //Ɣ Is the maximum number of rows in a bigroup
11:        nbrline++ ;
12:        Tab[nbrLine]:=arc[j] ;
13:        Eval:=0
14:        For k := 0 to nbrLine do // the maximum number of rows in a bigroup = n
15:            For z := k+1 to nbrLine do
16:                A[]:=extractionEntier(Tab[k]) ;
17:                B[]:=extractionEntier(Tab[z]) ;
18:                C[]:=extractionEntier(Tab[0]) ;
19:                Eval:=eval+csi(A[1],A[2],B[2])/c[0] ; // Calculate the CSI function and CSI complexity = O(m)
20:            End
21:        End
22:        If (eval/((nbrLine+1)*nbrline)/2<seuil) nbrLine-- ; // If our evaluation variable below the threshold so the last added arc will be deleted
23:    End
24:    Row:=returnRow(tab) ;    //O(n²)
25:    colomn :=returnColomn(row,Mat) ; //O(nm)
26:    B:=B+{row,colomn} ;
27: End // Extraction is of the order of complexity of O(n⁴m)
28: Return B;
29: End.
```

The complexity time is calculated from the for loop nested of the extraction part so we have O (n^4m + n^3 + n^2m) therefore:
O (n^4m + n^3 + n^2m) = O (n^2 (n^2m + n + m)) = O (n^2 (n (1 + nm) + m))→ 1 is negligible with respect to nm, (N^2 (n^2m + nm)) = O (n^2 (nm (1 + n))) → we also have 1 is negligible with respect to n, so that O (n^4m) is obtained as a time of final complexity.

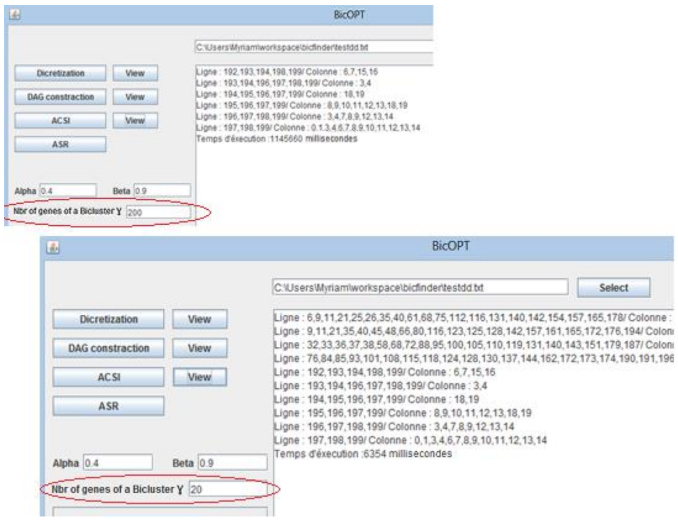
3.2 Improvement

Among the improvements made to our application, we mention: the addition of gamma parameter which allows us to limit the number of rows of a bigroup. In addition, the creation of a graphical interface that serves to display the results obtained and also to plot the gene expression curves of each group.

3.2.1 Size of biclusters

The columns of our bigroupe is calculated from the lines obtained from the ACSI function, so in the case of decreasing

values of the threshold α, the number of rows increases and in return the number of columns can decrease until no column is obtained.
Hence the risk of losing this bigroup altogether. We used another parameter ɣ that allows us to limit the number of rows of a bigroup.



C:\Users\Myriam\workspace\bicfinder\testdd.txt

Alpha 0.4 Beta 0.9
Nbr of genes of a Bicluster ɣ 200

```
Ligne : 192,193,194,198,199/ Colonne : 6,7,15,16
Ligne : 193,194,196,197,198,199/ Colonne : 3,4
Ligne : 194,195,196,197,199/ Colonne : 18,19
Ligne : 195,196,197,199/ Colonne : 8,9,10,11,12,13,18,19
Ligne : 196,197,198,199/ Colonne : 3,4,7,8,9,12,13,14
Ligne : 197,198,199/ Colonne : 0,1,3,4,6,7,8,9,10,11,12,13,14
Temps d'exécution :1145660 millisecondes
```

C:\Users\Myriam\workspace\bicfinder\testdd.txt

Alpha 0.4 Beta 0.9
Nbr of genes of a Bicluster ɣ 20

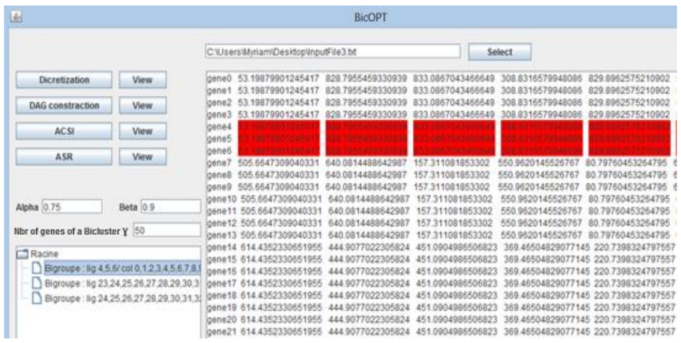
```
Ligne : 6,9,11,21,25,26,35,40,61,68,75,112,116,131,140,142,154,157,165,178/ Colonne :
Ligne : 9,11,21,35,40,45,48,66,80,116,123,125,128,142,157,161,165,172,176,194/ Colon
Ligne : 32,33,36,37,38,58,68,72,88,95,100,105,110,119,131,140,143,151,179,187/ Colon
Ligne : 76,84,85,93,101,108,115,118,124,128,130,137,144,162,172,173,174,190,191,196
Ligne : 192,193,194,198,199/ Colonne : 6,7,15,16
Ligne : 193,194,196,197,198,199/ Colonne : 3,4
Ligne : 194,195,196,197,199/ Colonne : 18,19
Ligne : 195,196,197,199/ Colonne : 8,9,10,11,12,13,18,19
Ligne : 196,197,198,199/ Colonne : 3,4,7,8,9,12,13,14
Ligne : 197,198,199/ Colonne : 0,1,3,4,6,7,8,9,10,11,12,13,14
Temps d'exécution :6354 millisecondes
```

Figure 4. Results obtained with gamma

For a first test, the value of ɣ is equal to n (where n is the number of rows of the data file), we obtained six biclusters with an execution time equal to 1145660 ms. We changed the value of parameter ɣ (number of genes generated) to 20 where we obtained ten biclusters with a run time of 6354 ms. So, the addition of this parameter allows the user to win in the execution time and in the number of biclusters obtained.

3.2.2 Display of biclusters

Unlike Command-row interface (CLI) systems that require commands to be stored, GUI systems offer a relatively intuitive approach. Even users without significant training can easily learn the system and use it to achieve their goals.



C:\Users\Myriam\Desktop\InputFile3.txt

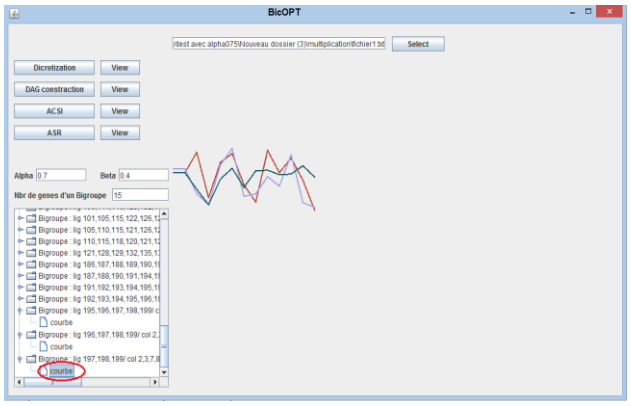
Alpha 0.75 Beta 0.9
Nbr of genes of a Bicluster ɣ 50

Bigroupe : lig 4,5,6/ col 0,1,2,3,4,5,6,7,8,9
Bigroupe : lig 23,24,25,26,27,28,29,30,3
Bigroupe : lig 24,25,26,27,28,29,30,31,3

```
gene0  53.19879901245417  828.7955459330939  833.0867043466649  308.8316579948086  829.8962575210902
gene1  53.19879901245417  828.7955459330939  833.0867043466649  308.8316579948086  829.8962575210902
gene2  53.19879901245417  828.7955459330939  833.0867043466649  308.8316579948086  829.8962575210902
gene3  53.19879901245417  828.7955459330939  833.0867043466649  308.8316579948086  829.8962575210902
gene4  [illegible]
gene5  [illegible]
gene6  [illegible]
gene7  505.6647309040331  640.0814488642987  157.311081853302  550.9620145526767  80.79760453264795
gene8  505.6647309040331  640.0814488642987  157.311081853302  550.9620145526767  80.79760453264795
gene9  505.6647309040331  640.0814488642987  157.311081853302  550.9620145526767  80.79760453264795
gene10 505.6647309040331  640.0814488642987  157.311081853302  550.9620145526767  80.79760453264795
gene11 505.6647309040331  640.0814488642987  157.311081853302  550.9620145526767  80.79760453264795
gene12 505.6647309040331  640.0814488642987  157.311081853302  550.9620145526767  80.79760453264795
gene13 505.6647309040331  640.0814488642987  157.311081853302  550.9620145526767  80.79760453264795
gene14 614.4352330651955  444.9077022305824  451.0904986506823  369.4650482907714  5 220.7398324797557
gene15 614.4352330651955  444.9077022305824  451.0904986506823  369.46504829077145  220.7398324797557
gene16 614.4352330651955  444.9077022305824  451.0904986506823  369.46504829077145  220.7398324797557
gene17 614.4352330651955  444.9077022305824  451.0904986506823  369.46504829077145  220.7398324797557
gene18 614.4352330651955  444.9077022305824  451.0904986506823  369.46504829077145  220.7398324797557
gene19 614.4352330651955  444.9077022305824  451.0904986506823  369.46504829077145  220.7398324797557
gene20 614.4352330651955  444.9077022305824  451.0904986506823  369.46504829077145  220.7398324797557
gene21 614.4352330651955  444.9077022305824  451.0904986506823  369.46504829077145  220.7398324797557
```

Figure 5. Display of biclusters

The design of the interface of our system offers the user several advantages, of which we quote that it allows to follow and to visualize all the steps of extraction of bigroupes and also allows him to determine the position of bigroupe in the matrix of initial data.



\test avec alpha075\Nouveau dossier (3)\multiplication\fichier1.txt

Alpha 0.7 Beta 0.4
Nbr de genes d'un Bigroupe 15

Bigroupe : lig 101,105,115,122,126,1
Bigroupe : lig 105,110,115,121,126,1
Bigroupe : lig 110,115,118,120,121,1
Bigroupe : lig 121,128,129,132,135,1
Bigroupe : lig 186,187,188,189,190,1
Bigroupe : lig 187,188,190,191,194,1
Bigroupe : lig 191,192,193,194,195,1
Bigroupe : lig 192,193,194,195,196,1
Bigroupe : lig 195,196,197,198,199/ c
Bigroupe : lig 196,197,198,199/ col 2,
Bigroupe : lig 197,198,199/ col 2,3,7,8

Figure 6. Gene expression curve

To measure the coherence of dictated biclusters, BicOPT makes it possible to visualize the curves created as a function of the level of expression of genes (see FIG. 6).
If the curves are similar, we can deduce that our bigroup has a strong coherence.

4 Results

The data file used by our program must be structured in the following format:

Table III. Sample data file

$Gene_0$	15	…	100
………	…	…	…
$Gene_i$	12.4	…	-15
……..	…	…	…
$Gene_n$	10.5	…	125

The first column of the file must contain the name of the genes. The columns are separated with spaces. Each row contains the gene name followed by a set of gene expressions.
However, to test BicOPT's ability to extract different types of biclusters, we used a set of real files : Human B-cell Lymphoma dataset [1] with the size of (4026 rows, 96

[1] Available on http://arep.med.harvard.edu/biclustering/

columns) and Saccharomyces Cerevisiae dataset[2] with the size of (2993 rows, 173 columns).
In order to compare the results obtained by BicOPT with the other algorithms we used the BicAT toolbox [3], this tool contains a set of bicluster algorithms (BiMax [11], CC [6], ISA [5], OPSM [4], xMotives [10]). The BicOPT algorithm and all other algorithms are run on an Intel Core I3 2.2 GHz machine and 4 Gb of RAM.

4.1 Human B-cell Lymphoma dataset

The BicOPT settings are set to 0.8 for Alpha and 0.4 for Beta. The test execution time of our algorithm lasted 137.15 minutes.

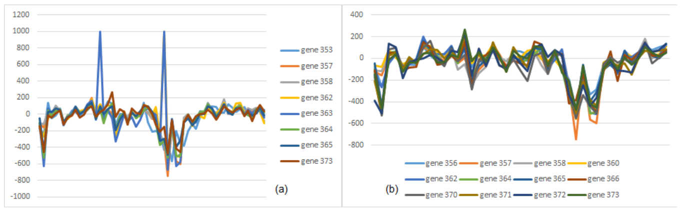


Figure 7. Expression profiles of two biclusters

The first bigroup of size (8.57), which is presented by the curve (a) (see Figure 7) and the second bigroup of size (12, 49), is presented by the curve (b) Figure 7). We have associated a curve for each gene, the latter is created as a function of the level of gene expression.
The curves are grouped together in the same frame in order to form an expression profile of a bicluster. According to curve (a) we have a strong gene expression profile with the presence of some noises. In the second curve (b) we also observe a strong gene expression profile but without the presence of noises. The presence of noise in the data file is invaluable in the groups detected by BicOPT.

4.2 Saccharomyces Cerevisiae dataset

After testing a set of simulations, the BicOPT parameters were set to 0.85 for Alpha and 0.7 for Beta. The test execution time of our algorithm is 54.5 minutes. To evaluate the biological relevance of bigroupes detected by our algorithm, we used two web tools, *GOTermFinder*[3] and *FuncAssociate*[4], to calculate the p-value [13]. More than the p-value is lower more than the bigroup genes are consistent.

4.2.1 GOTermFinder :

GOTermFinder is a well-known web-tool which allows to check the quality of each detected group and to search for the significant terms of gene ontology shared by the selected gene groups. To identify the characteristics that the genes can have in common, we selected three random groups in a random way (see Table 7).

Table IV.The most significant GO terms for three bigroupes.

Biclusters	Biological Process	Molecular Function	Cell Component
20 Genes x 101 Conditions	Cytoplasmic Translation (95.0%, 3.1E-28)	Structural component of the ribosome (95.0%, 3.82E-27)	Cytosolic Ribosome (95.0%, 8.66E-29)
	Biosynthesis process of peptides (95.0%, 2.22E16)	Structural molecular activity (95.0%, 1.31E-23)	Ribosomal subunit (95.0%, 5.85E-26)
18 Genes x 91 Conditions	Metabolic process of ncRNA (88.9%, 1.33E-14)	SnoRNA binding (27.8%, 2.65E-08)	Nucleolus (83.3%, 1.70E-16)
	Treatment of ncRNA (83.3%, 2.41E-14)	RNA binding (55.6%, 0.00017)	Preribosome (72.2%, 3.74E-16)
18 Genes x 88 Conditions	Cytoplasmic translation (83.3%, 3.66E-20)	Structural Component Of the ribosome (83.3%, 1.61E-19)	Cytosolic Ribosome (83.3%, 1.19e-20)
	Translation (83.3%, 5.70E-11)	Structural molecular activity (83.3%, 9.18E-17)	Ribosomal subunit (83.3%, 1.90e-18)

We used the GOTermFinder tool to describe the most significant shared terms, respectively, for the biological process, molecular function, and cellular component (see Table 4). For the first bigroup of size (20,101) we have for example Cytoplasmic Translation (95.0%, 3.1E-28), so this bigroup is involved in Cyptoplasmic translation with a frequency of 95.0% (among 20 genes in the first bigroup 19

[2] Available on http://people.ee.ethz.ch/~sop/bimax/SupplementaryMaterial/Datasets/BiologicalValidation/data/saccharomyces/yeast_GOEnrichment,Gasch2000,2944x173.txt

[3] Available on http://www.yeastgenome.org/cgi-bin/GO/goTermFinder.pl
[4] Available on http://llama.mshri.on.ca/funcassociate/

belong To this process) and with a p-value equal to 3.1E-28 (very significant value).

4.2.2 FuncAssociate

FuncAssociate is a web application that helps to discover properties enriched in lists of genes or proteins that emerge from the experiment on a large scale [13]. The basic idea is to select 20 biclusters and then to determine whether the set of genes discovered by biclustering algorithms shows a significant enrichment compared to an annotation of genetic ontology (GO) or not (see Figure 8).

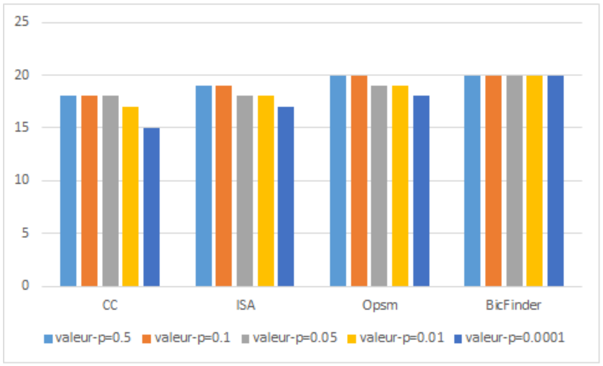


Figure 8. Percentages of Biclusters enriched by GO annotations

For values associated with parameter p, BicOPT surpassed the other algorithms with a percentage of 100% followed by Opsm with a percentage 90% for p = 0.0001. The other algorithms also perform reasonably well. The experiments applied to the real data set used prove that our proposed algorithm can identify bigroups with high biological relevance.

5 Conclusion

In this paper, we propose the BicOPT algorithm which presents a new optimized version of the BicFinder algorithm. The complexity time of our algorithm is equal to O (n^4m), which is less than that of BicFinder. BicOPT allows the extraction and the production of a set of biclusters based on the construction of an acyclic oriented graph (DAG). We added a new GAMMA parameter to limit the gene numbers of each generated bigroup. BicOPT has a graphical interface allowing to manage well the obtained bigroupes We realized different tests on real databases to evaluate the performance of BicOPT. In the realization of this study, we used two web tools GOTermFinder, FuncAssociate and a BicAT application. The experimental study of our approach to biclustering have good results.

References

[1] W. Ayadi, M. Elloumi, and J. K. Hao, “BicFinder : A biclustering algorithm for microarray data analysis,” Knowl. Inf. Syst., Vol. 30, N°2 : (2012), p341–358.

[2] W. Ayadi and M. Elloumi. Algorithms in Computational Molecular Biology: Techniques, Approaches and Applications, chapter Biclustering of Microarray Data.Wiley Book Series on Bioinformatics : Computational Techniques and Engineering. Wiley-Blackwell, John Wiley & Sons Ltd., New Jersey, USA (Publish), (2011), p651-664.

[3] S. Barkow, S. Bleuler, A. Prelic, P. Zimmermann, and E. Zitzler. Bicat: a biclustering analysis toolbox. Bioinformatics, Vol. 22, N°10 : (2006), p1282–1283.

[4] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data: the order-preserving submatrix problem. In RECOMB ’02: Proceedings of the sixth annual international conference on Computational biology, p49–57, New York, NY, USA, 2002. ACM.

[5] S. Bergmann, J. Ihmels, and N. Barkai. Defining transcription modules using large-scale gene expression data. Bioinformatics, Vol. 20, N°13 : (2004), p1993–2003.

[6] G. F. Berriz, J. E. Beaver, C. Cenik, M. Tasan, and F. P. Roth, “Next generation software for functional trend analysis,” vol. 25, N°22 : (2009), p3043–3044.

[7] M. (Riadi) Charrad, G. (Riadi) Saporta, Y. Lechevallier, and M. Ben Ahmed, “Le bi-partitionnement : Etat de l’art sur les approches et les algorithmes,”Ecol’IA'08 : (2008).

[8] Y. Cheng, G.M. Church, Biclustering of Expression Data, in Proc. International Conference on Intelligent Systems for Molecular Biology : (2000), p93-103.

[9] S. C. Madeira, A. L. Oliveira, Biclustering Algorithms for Biological Data Analysis: A Survey, IEEE Transactions on Computational

Biology and Bioinformatics , Vol.1, N°1 : (2004), p24-45.

[10] S. K. T. M. Murali, Extracting conserved gene expression motifs from gene expression data, Pac. Symp. Biocomput, Vol.8 : (2003), p77-88.

[11] A. Prelic, S. Bleuler, P. Zimmermann, P. Buhlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics, Vol. 22, N°9 : (2006), p1122–1129

[12] Y.S. Son and J. Baek. A modified correlation coefficient based similarity measure for clustering time-course gene expression data. Pattern Recognition Letters, Vol. 29, N°3 : (2008), p232–242.

[13] Wasserstein, Ronald L.; Lazar, Nicole A. "The ASA's Statement on p-Values: Context, Process, and Purpose". The American Statistician. Vol. 70, N°2 : (2016), p129–133.