BiBinConv$_{mean}$:A Novel Biclustering Algorithm for Binary Microarray Data

Haifa BEN SABER

Time Université
Laboratory of Technologies of Information and
Communication and Electrical Engineering (LaTICE)
National High School of Engineers of Tunis (ENSIT),
University of Tunis, Tunisia.

Mourad ELLOUMI

Laboratory of Technologies of
Information and Communication and
Electrical Engineering (LaTICE)
University of Tunis-El Manar, Tunisia.

***Abstract*— In this paper, we present a new algorithm called, BiBinConv$_{mean}$, for biclustering of binary microarray data. It is a novel alternative to extract biclusters from sparse binary datasets. Our algorithm is based on Iterative Row and Column Clustering Combination (IRCCC) and Divide and Conquer (DC) approaches, K-means initialization and the CroBin evaluation function [6]. Applied on binary synthetic datasets, our algorithm outperforms other biclustering algorithms for binary microarray data. Biclusters with different numbers of rows and columns can be detected, varying from many rows to few columns and few rows to many columns. Our algorithm allows the user to guide the search towards biclusters of specific dimensions.**

Keywords-component; Biclustering, binary data, microarray data, Iteratif Row Column Combinaison approach, Divide and Conquer approach, CroBin.

I. Introduction

A DNA Microarray is a glass slide covered with a product and DNA samples containing thousands of genes [8]. Biclustering of microarray data can be helpful to study, among others, the activity and the condition of the tissue via microarrays such as transcription factor binding, insertional mutagenesis and gene expression data. It can be helpfull also to find genes involved in tumor progression, identify the function of new genes, rank the tumors into homogenous groups and identify new therapeutic strategies.

Biclustering algorithms of binary microarray data enable to extract useful biclusters from binary data to provide information about the distribution of patterns and intrinsic correlations [16][15]. A number of biclustering algorithms of binary microarray data have been proposed in recent years, such as the Biclustering Bit-pattern (BiBit) [14][1], Cmnk [11], [9], BiMax [13], Bipartite Bron-Kerbosch (BBK) [12], Binary Matrix Factorization (BMF) [18], e-CCC Biclustering [5], e-BiMotif [5], BIMODULE[4], BIDENSE[4], CETree [4], DeBi [17] and Maximal Frequent Item Set [17]. Besides, there are also other approaches based on Gaussian or Latent Mixture Models, BEM and BCEM [3].

In the same context, different biclustering algorithms have been adapted to deal with biclustering of binary gene expression data. However, these changes lead to more complicated user input parameters. Besides, all the elements of every generated bicluster are set to zero in the input matrix, introducing noise.

It is an interesseting from the biological point of view [12] to search biclusters with small proportion of zeros especially when binary data matrix is obtained after normalization and binarization. However, most of biclustering algorithms of binary microarray data, inculding Cmnk and BiMax, fail to extract pertinent biclusters on sparse binary datasets. Indeed, if we apply one of these algorithms on a typical sparse binary microarray datasets (with thousands of columns), most of the extracted biclusters are made up only by 1's.

The rest of this paper is organized as follows. In section 2, we introduce some preliminaries. In section 3, we present BiBinConv$_{mean}$ our biclustering algorithm of binary microarray data. In section 4, we present an illustrative example. In section 5, we present the experimental results obtained thanks to BiBinConvmean on binary synthetic, and we compare these results with those obtained by other biclustering algorithms of binary microarray data. Finally, in section 6 we present the conclusion.

II. Preliminiaries

In this section we present some preliminaries necessary for the presentation of important formulas and relationships and the used theory.

Let $I = \{1, 2, .., n\}$ be a set of indices of n genes, $J = \{1, 2, .., m\}$ be a set of indices of m conditions and $M(I, J)$ be a data matrix associated with I and J.

The biclustering problem of a binary microarray data boils down to a minimization of the criterion W(z, w, A) defined by:

(DOI Reference Number: 10.18293/SEKE2015-007)

$$W(z,w,A) = \sum_{k=1}^{a}\sum_{l=1}^{m}\sum_{i\in z_k}\sum_{j\in w_l} |m_{ij} - a_{kl}|. \quad (1)$$

where:

- z is a partition of I into g clusters.
- w is a partition of J into m clusters.
- A is the summary of the data matrix where k (resp. l) represents number of clusters on rows (resp. columns). We note that the bicluster kl is defined by the m_{ij} with $z_{ik}w_{jl} = 1$.

III. Contribution

In this section, we develop our biclustering algorithm, BiBinConvmax that is based on the IRCCC approach, K-means initialisation and the CroBin evaluation function. It consists to permute the rows and the columns in order to obtain homogeneous biclusters. As a preprocessing step of this algorithm is applied:

a) First, when the data matrix M is not a binary one, we apply a thresholding function to transform it to a binary one. To the best of our knowledge, the main thresholding functions are discretize, normalize and binarize [7]. According to [14], [9][14], [9], the most adequate thresholding functionto binarize microarray data is binarize.

b) Then, we make an initial clustering z^0 of rows and an initial clustering w0 of columns, thanks to k-means algorithm [10] [2] .

After the initialization, we update the clustering of rows and columns.

The preprocessing step consists in obtaining a binary matrix Mb that can be directly used by our algorithm. The binary values of 1 and 0 under an experimental condition c mean that a gene is expressed or not, respectively. For example, in [13], a discretization threshold was set to $e^2+(e^1- e^2)=2$, with e^1 and e^2 as the minimum and maximum expression values in the data matrix, respectively.

Our biclustering algorithm, BiBinConv$_{\text{mean}}$ receives as input a binary matrix M gives as output (z^{opt}, w^{opt}, A^{opt}), where zopt is the final clustering of rows of Mb, wopt is the final clustering of columns of Mband Aopt is the summary matrix related to zopt and wopt.

By adopting BiBinConv$_{\text{mean}}$, we operate as follows:

First, we compute (z^0, w^0, A^0) thanks to k-means algorithm [10], [2], where z^0 is the initial clustering of rows of M_b, w^0 is the initial clustering of columns of M_b and A^0 is the summary matrix related to z^0 and w^0.

Then, we repeat this process :

We compute (z^c, w^{c-1}, A') starting from (z^{c-1}, w^{c-1}, $A^{c1'}$), where A' is an intermediate summary matrix

We compute (z^c, w^c, $A^{c'}$) starting from (z^{c-1}, w^{c-1}, A')

Until (z^c, w^c, $A^{c'}$) = (z^{c-1}, w^{c-1}, $A^{c-1'}$).

IV. Illustrative Example

We present in this section steps to perform the biclustering on binary datasets. Our algorithm allows to reorder the rows and the columns of the data matrix in both dimensions to obtain homogeneous biclusters. The algorithm minimizes the difference between the initial matrix according the two way and the ideal matrix.

To illustrate our algorihtm method, we propose to run it on an simple example. Let M_b be a (4,5) matrix of binary data to perform biclustering. The initialization is to group the rows and columns with K-means reference algorithm method. After initialization, we compute z and w matrix whose elements determine the membership of rows or column in horizontal or vertical clusters, respectively.

Then, we reorganize the binary matrix. After that, we compute the summary matrix is obtained from z and w and the bicluster kl is defined by the x_{ij} 's with $z_{ik}w_{jl} = 1$.

The summary matrix is presented by the major value akl which presents the degree of homogeneity via summary matrix. We update clusters on rows and columns by computing our criterion on both dimensions. Initial binary matrix M_{bis} given by :

	C1	C2	C3	C4	C5
G1	1	1	0	1	0
G2	0	0	1	0	1
G3	1	1	0	1	0
G4	0	0	1	0	1

- **Initialization:**

z^0 = (1, 2, 2, 3), w^0 = (1, 1, 0, 0, 0), A^0 = (1, 0, 1, 1, 0, 1)

- **Iteration 1:**

c = 1

We compute (z^1, w^0, A') starting from (z^0, w^0, A^0), we obtain:

(z^1, w^0, A') = ((1, 3, 2, 1); (1, 1, 0, 0, 0); (1, 1, 1, 0, 0, 1))

We compute (z^1, w^1, A') starting from (z^1, w^0, A'), we obtain:

(z^1, w^1, A') = ((1, 3, 2, 1); (2, 2, 1, 2, 1); (1, 1, 1, 0, 0, 1))

We obtain (z^2, w^2, A^2) != (z^1, w^1, A^1).

- **Iteration 2: c = 2**

We compute (z^2, w^1, A') starting from (z^1, w^1, A^1), we obtain

(z^2, w^1, A')= ((2, 1, 2, 3); (2, 2, 1, 2, 1); (0, 1, 1, 0, 0, 1))

We compute (z^2, w^2, A') starting from (z^2, w^1, A'), we obtain:

(z^2, w^2, A')= ((2, 1, 2, 3); (2, 2, 1, 2, 1); (0, 1, 1, 0, 0, 1))

We obtain (z^3, w^3, A^3)!= (z^2, w^2, A^2)

- **Iteration 3: c = 3**

We compute (z^3, w^2, A') starting from (z^2, w^2, A^2), we obtain:

$(z^3, w^2, A') = ((1, 1, 1, 1); (2, 2, 1, 2, 1); (1, 0))$

We compute (z^3, w^3, A') starting from (z^3, w^2, A'), we obtain:

$(z^3, w^3, A') = ((2, 1, 2, 3); (2, 2, 1, 2, 1); (1, 0))$

We obtain $(z^3, w^3, A') != (z^2, w^2, A^2)$.

- **Iteration 4: c = 4**

We compute (z^4, w^3, A') starting from (z^3, w^3, A'), we obtain:

$(z^3, w^3, A') = ((1, 2, 1, 2); (2, 2, 1, 2, 1); (1, 0, 0, 1))$

We compute (z^4, w^3, A') starting from $(z^4, w^3, A') = (z^3, w^3, A')$, we obtain:

$(z^4, w^4, A') = ((1, 2, 1, 2); (1, 1, 1, 1, 1); (0, 1))$

We obtain $(z^4, w^4, A^4) != (z^3;w^3;A^3)$.

- **Iteration 5: c = 5**

We compute (z^5, w^4, A') starting from (z^4, w^4, A'), we obtain:

$(z^5, w^4, A') = ((1, 2, 1, 2); (1, 1, 1, 1, 1); (0, 1))$

We compute (z^5, w^5, A') starting from (z^5, w^4, A'), we obtain:

$(z^5, w^5, A') = ((1, 2, 1, 2); (1, 1, 2, 1, 2); (1, 0, 0, 1))$

We obtain $(z^5, w^5, A^5) = (z^4, w^4, A^4)$.

After five iterations, we obtain $(z^{opt};w^{opt};A^{opt}) = (z^5;w^5;A^5)$.

	C1	C2	C4	C3	C5	Z
G1	1	1	1	0	0	1
G3	1	1	1	0	0	1
G2	0	0	0	1	1	2
G4	0	0	0	1	1	2
w	1	1	1	2	2	X

Colored blocs represent the obtained biclusters thanks to BiBinConv_{mean}

.

IV. EXPERIMENTAL RESULTS

We have experimented the $\text{BiBinConv}_{\text{mean}}$ algorithm on simulated data. To generate our simulated data, we operarte as follows:

We choose the number of biclusters; in our case we choose 3 clusters on rows (g = 3) and 2 clusters on columns (m = 2).

We use Latent Bernoulli Mixture (LBM) model to generate mixtures by considering pattterns of overlapping well separated +5% or fairly separated: + + 15% or poorly Separated +++25% and sizes of data are used; small one (50, 30), medium one (100, 60) and large one (200, 120).

For instance, to apply the simulation on (100, 60) matrix for 10 samples with a low degree of mixing such that a 3 clusters proportions on rows p_k, 2 clusters proportions on columns q_l

Where:

p_k = [0:2; 0:3; 0:5], $theta_0$= [0:3; 0:7], $Alpha_0$ = [0:7; 0:3; 0:3; 0:7; 0:7; 0:7] and a degree of mixture belonging [0:00; 0:05].

The $\text{BiBinConv}_{\text{mean}}$ algorithm is fast and gives good results when the biclusters have the same proportions and degrees of homogeneity similar except that you set the number of clusters on the rows and columns. However, it seems to be bad when the proportions of partitions are dramatically different which leads to think that $\text{BiBinConv}_{\text{mean}}$ assumes equal proportions of clusters. This algorithm rearranges the rows and columns of the data matrix along the two sheets of the rows and columns of homogeneous biclusters. The algorithm minimizes the difference between the initial matrix structured and the ideal matrix according scores.

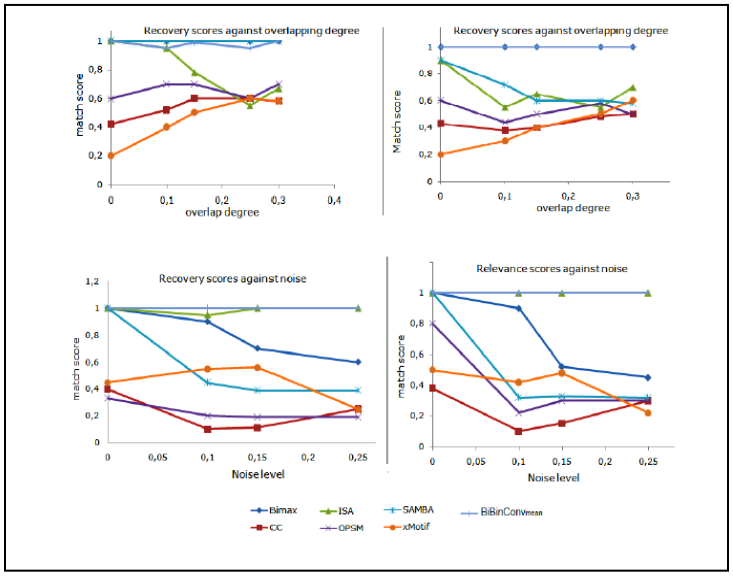

Figure 1. Recovery Scores against overlapping degree and noise

The summarize of the most important points obtained from these simulations are as follows: Obviously, we can interpret the reasonable results according to the model underlying the data structure. In this research, we proves that if the proportions of the components are considered equal give good results. The convergence has a fast progress: most of the time. The $\text{BiBinConv}_{\text{mean}}$, is faster and gives us interesting results.

The obtained bicluster is very clear since extracted biclusters containing a majority number of ’1’ and very illustrative to help the biologist to extract knowledge. According to our implementations, we note first that the choice and application of a given criterion is not always obvious or easy to find. According to our study, we find that $\text{BiBinConv}_{\text{mean}}$ does not give good results when the matrix becomes more large. We remark also that the error rates are proportional according the overlap rate.

V. CONCLUSION

In this paper, we presented our method of binary biclustering considered to be relevant by the expert starting from the data resulting from the microarrays. The suggested method generates interesting biclusters. To achieve this purpose, we selected data to which we applied a thresholding to release the binary data. Our proposal has been implemented and evaluated on synthetic datasets.

According to our implementation, we note first that the choice and application of a given criterion is not always obvious or easy to find. Enjoying the benefits of our algorithm, we proposed a methodology for the identification of homogeneous biclusters. The first results are very encouraging and persuade us of the obvious interest of such an approach.

Finally, further analysis and biological validation of the obtained results is under study.

REFERENCES

[1] Aguilar-Ruiz and Jesús S. Shifting and scaling patterns from gene expression data. Bioinformatics, 21(20):3840– 3845, 2005.

[2] Khalid Benabdeslem and Kais Allab. Bi-clustering continuous data with self-organizing map. Neural Computing and Applications, 22(7):1551–1562, 2013.

[3] M. Charrad. Une approche genrique pour l-analyse croisant contenu et usage des sites web par des methodes de bipartitionnement. PhD thesis, Paris and ENSI, University of Manouba, 2010.

[4] Jiun-Rung Chen and Ye-In Chang. A conditionenumeration tree method for mining biclusters from dna microarray data sets. Elsevier, 97:44–59, 2007.

[5] Joana P. Gonalves and Sara C. Madeira. e-bimotif: Combining sequence alignment and biclustering to unravel structured motifs. In IWPACBB, volume 74, pages 181–191, 2010.

[6] Gerard GOVAERT. La classification croisee. Modulad, 1983.

[7] Santamaria R. Khamiakova T. Sill M. Theron R. Quintales L. Kaiser, S. and F. Leisch. biclust: Bicluster algorithms. R package., 2011.

[8] Ouafae Kaissi. Analyse de Données Transcriptomiques pour La Recherche de Biomarqueurs Liés à Certaines Pathologies Cancéreuses. PhD thesis, University Abdelmalek Essaadi, Tangier, Morocco,, sep 2014.

[9] Mehmet Koyuturk. Using protein interaction networks to understand complex diseases. Computer, 45(3):31–38, 2012.

[10] G. C. Marcos A.S. da Silva AND, Antonio M.V. Monteiro AND. Som-code: Design patterns and generic programming in the implementation of self organizing maps. BMC Genomics., 2013.

[11] Ananth Grama Mehmet Koyuturk, Wojciech Szpankowski. Biclustering gene-feature matrices for statistically significant dense patterns. In 2004 IEEE Computational Systems Bioinformatics Conference (CSB'04), pages 480–484, 2004.

[12] Stefan Bleuler Oliver Voggenreiter and Wilhelm Gruissem. Exact biclustering algorithm for the analysis of large gene expression data sets. Eighth International Society for Computational Biology (ISCB) Student Council Symposium Long Beach, CA, USA.July, pages 13–14, 2012.

[13] Amela Prelic, Stefan Bleuler, Philip Zimmermann, Anja Wille, Peter Bühlmann, Wilhelm Gruissem, Lars Hennig,

[14] Lothar Thiele, and Eckart Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics, 22:1122–1129, 2006.

[15] Perez-Pulido A. J. Rodriguez-Baena, D. S. and J.S. Aguilara-Ruiz. A biclustering algorithm for extracting bit-patterns from binary datasets. Bioinformatics., 2011.

[16] Bhattacharyya D. K. Roy, S. and J. K. Kalita. Cobi: Pattern based coregulated biclustering of gene expression data. Pattern Recognition Letters., 2013.

[17] Akdes Serin. Biclustering analysis for large scale data. Phd., 2011.

[18] Akdes Serin and Martin Vingron. Debi: Discovering differentially expressed biclusters using a frequent itemset approach. Algorithms for Molecular Biology, 6:18, 2011.

[19] Chris Ding Xian Wen Ren Xiang Sun Zhang Zhong Yuan Zhang, Tao Li. Binary matrix factorization for analyzing gene expression data. Data Mining and Knowledge Discovery, 20:28–52, 2010.

//

[20] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions, " Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

.