ROBUST BICLUSTERING ALGORITHM (*ROBA*) FOR DNA MICROARRAY DATA ANALYSIS

Alain B. Tchagang, and Ahmed H. Tewfik

Electrical and Computer Engineering, University of Minnesota
200 Union Street SE, MN, 55455, Minneapolis, USA
phone: + (1) 612 625 6024, fax: + (1) 612 625 4583, email: tcha0003@umn.edu, tewfik@umn.edu
web: http://www.ece.umn.edu/users/tewfik/

ABSTRACT

Recently, biclustering algorithms have been used to extract useful information from large sets of *DNA* microarray experimental data. They refer to a distinct class of clustering algorithms that perform simultaneous row-column clustering. The goal is to find submatrices, that is, subgroups of genes and subgroups of conditions, where the genes exhibit highly correlated activities for every condition. Almost all of the methods proposed in the literature search for one or two types of bicluster among four. Also, most of the proposed methods rely on solving an optimization problem. Therefore, the method is dependant on the optimally criterion which most of the time, is likely to miss some significant biclusters. In this study, we develop a Robust Biclustering Algorithm to address the two issues mentioned above. The proposed algorithm is simple because it uses basic linear algebra and arithmetic tools and there is no need to solve an optimization problem.

1. INTRODUCTION

The data obtained from *DNA* microarray experiments is usually in the form of large matrices of data illustrating the expression levels of genes, rows of the matrix under different samples such as tissues or experimental conditions, columns of the matrix. Investigations show that more often, several genes contribute to a disease; also, many activation patterns are common to a group of genes only under specific experimental conditions. These facts motivate researchers to identify a subset of genes whose expression levels exhibit a coherent pattern under a subset of conditions. Discovery of such pattern is therefore essential in revealing the significant connections in gene regulatory networks.

The biclustering algorithm was first used by Cheng and Church in [2] to extract such patterns from large sets of experimental data. It refers to a distinct class of clustering algorithms that perform simultaneous row-column clustering. The goal is to find submatrices, that is, subgroups of genes and subgroups of conditions, where the genes exhibit highly correlated activities for every condition. Many other biclustering algorithms have been proposed in the literature to perform such task [2 - 8]. Almost all of the proposed methods search for one or two types of biclusters among four types that have been identified in the literature [1]: biclusters with constant values, biclusters with constant values on rows or columns, biclusters with coherent values, and biclusters with coherent evolution. Also, most of the proposed methods rely on solving an optimization problem. Therefore, the method is dependant on the optimally criterion which most of the time, is likely to miss some significant biclusters. For example, biclustering algorithms based on greedy methods rarely find the globally optimal solution consistently, since they usually don't operate exhaustively on all the data.

In this study, we develop a Robust Biclustering Algorithm (*ROBA*) to address the two issues mentioned above. The proposed algorithm is simple because it uses basic linear algebra and arithmetic tools and there is no need to solve and optimization problem. We illustrate the proposed algorithm here by focusing on the identification of biclusters with constant values, biclusters with constant values on rows, biclusters with constant values on columns, and biclusters with coherent values.

The rest of this paper is organized as follows. After a quick description of gene expression matrix in section *2*, we perform a quick review of previous biclustering algorithms and their limitations in section *3*. We develop part of the Robust Biclustering Algorithm in section *4*. In section *5*, we show some simulation results and compare the performance of our algorithm with previous ones.

2. GENE EXPRESSION MATRIX

A *DNA* microarray data is an *NxM* matrix *A* whose rows represent the genes, its columns the experimental conditions, and a_{nm} is a real number that represents the expression level of gene *n* under condition *m*.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1M} \\ a_{21} & a_{22} & \dots & a_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nM} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NM} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \\ \vdots \\ r_N \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & \dots & c_m & \dots & c_M \end{bmatrix}$$

$r_n = [\, a_{n1}\, a_{n2} \dots a_{nm} \dots a_{nM}]$, $c_m = [a_{1m}\, a_{2m} \dots a_{nm} \dots a_{Nm}]^T$

Conditions = [Condition1 ... Condition m ... Condition M]

Genes = [Gene1 Gene2 Gene 3 ... Gene n ... Gene N]^T

The row vector r_n corresponds to the expression levels of the n^{th} gene under *M* conditions. The column vector c_m corresponds to the expression levels of the *N* genes under the m^{th}

condition. The row vector *Conditions (1xM)* and the column vector *Genes (1xN)* are label vectors. They are defined to keep track of every condition and gene.

3. PREVIOUS WORKS

As mentioned above, there exists an extensive literature on biclustering techniques, e.g., [2-8]. Most of those previous techniques are greedy and will miss meaningful biclusters. Some, such as [8], are exhaustive. To ensure a reasonable run time, exhaustive techniques will restrict the maximum size of the bicluster. Also, almost all of the previous techniques used a cost function to define biclusters. For example, the cost function can measure the square deviation from the sum of the mean value of expression levels in the entire bicluster, and the mean values of expression levels along each row and column in the bicluster.

In contrast, in our approach, we operate on all the data and we do not limit the number of genes that can appear in a bicluster. Secondly, we use a deterministic method that allows the user to identify all qualified biclusters in each type. Specifically, we consider each type of bicluster defined above, and unlike prior work, we proceed by identifying the number of biclusters they contain by decomposing the gene expression matrix into its distinct elements which later run, allow us to get a hand on all of the qualified biclusters. This approach avoids the need for exhaustive enumeration or heuristic cost functions that can miss some pertinent biclusters. We propose an effective algorithm to mine biclusters. Compared with other biclustering approaches, our method is deterministic in that it discovers all qualified biclusters, while previous biclustering approaches are random algorithms that provide only an approximate answer.

4. ROBUST BICLUSTERING ALGORITHM (*ROBA*)

The Robust Biclustering Algorithm is made up of three main parts. The first part consists of performing the data conditioning, to get rid of the noise and to solve the problem of missing values. The second part consists of decomposing the data matrix *A* into its elementary matrices. The third part consists of extracting any type of biclusters defined by the user.

4.1 Data Conditioning

The first part of *ROBA* consists of performing the data conditioning due to the fact that we are not only working with noisy data but also the *DNA* experimental data contains missing values. Many techniques to recover missing values have been developed in the literature [9, 10]. In this study we have used the zero method that is replacing each missing value by zero. To deal with noise, we first identify the number L of distinct values α_l that constitutes the gene expression matrix A next, we redefine α_l using equation 1.

$$\alpha_l = (b_l + b_{l-1})/2 \qquad (1)$$

Where: $b_l = b_0 + le$, with $l = 1$ *to* L, $e = (b_L - b_0)/L$, $b_0 = min([a_{nm}])$ and $b_L = max([a_{nm}])$. The interval $[b_0\ b_L]$ is then divided into L equal intervals. $[b_0\ b_L] = [b_0\ b_1[\ U\ \ldots\ U\ [b_{l-1}\ b_l[\ U\ \ldots\ U\ [b_{L-1}\ b_L]$. Finally, a new data matrix is obtained using Algorithm1.

Algorithm 1

```
Input A = Microarray Data
Compute: L, b_L, b_0, e, b_l, α_l
For l = 1 to L
        For n = 1 to N
                For m = 1 to M
                        If a_nm ∈ [b_l-1 b_l[
                                a_nm = α_l
                        End
                End
        End
End
```

4.2 Gene Expression Matrix Decomposition

The second part of *ROBA* consists of decomposing the matrix A into its elementary matrices. Given that A is made up of L distinct values, A can be decomposed using equation 2.

$$A = \sum_{l=1}^{l=L} \alpha_l A_l = \alpha_1 A_1 + \ldots + \alpha_L A_L \qquad (2)$$

$$A_l = \left[r_1^l \quad r_2^l \quad \ldots \quad r_n^l \quad \ldots \quad r_N^l\right]^T = \left[c_1^l \quad c_2^l \quad \ldots \quad c_m^l \quad \ldots \quad c_M^l\right]$$

$r_n = \sum_{l=1}^{l=L} \alpha_l r_n^l$, and $c_m = \sum_{l=1}^{l=L} \alpha_l c_m^l$. From equation (2), A_l's are binary NxM matrices, r_n^l's are binary $1xM$ vectors and c_m^l's are binary $Nx1$ vectors.

4.3 Biclusters Identification

4.3.1 Biclusters with Constant Values

A biclusters with constant values is any submatrix B *(IxJ)* of A whose elements are constant:

$$B = [a_{ij}] = \mu.Ones(I,J) \qquad (3)$$

With: $a_{ij} = \mu$, $i = 1$ *to* I, $j = 1$ *to* J. Such matrices represent subgroups of genes with constant expression levels under different conditions or vice versa. From (2), such matrices can be obtained by analyzing each A_l separately to obtain subgroups of genes that have constant expression level α_l under different conditions. Since A_l, is a binary matrix, and since the number of genes N is always greater than the number of conditions M, the number of biclusters (N_b) with constant values can be defined using equation (4).

$$N_b = \sum_{l=1}^{l=L} P_l \qquad (4)$$

P_l is the number of distinct rows r_i^l of each A_l whose sum is greater than *0*, that is; $sum(r_i^l) > 0$. Each distinct row r_i^l of A_l constitutes the principal row element of the i^{th} bicluster B_i^l of the matrix A_l considered. Therefore, in order for any other row r_n^l of A_l to belong to the i^{th} bicluster, equation (5) has to be verified: that is the element wise product of the two given row vectors.

$$r_i^l .* r_n^l = r_i^l \qquad (5)$$

With: *i = 1 to P_l, n = 1 to N, and l = 1 to L*. Algorithm 2 is then used to extract biclusters that have constant expression level α_l.

4.3.2 Biclusters with Constant Values on Columns

A bicluster with constant values on column is any submatrix *B (IxJ)* of *A* which has one of the following forms:$B = [a_{ij}]$, with $a_{ij} = \mu + \beta_j$ additive model or $a_{ij} = \mu.\beta_j$, multiplicative model. The general form can be represented using equation (6).

$$B = \begin{bmatrix} . & . & \dots & . \\ \mu_1 & \mu_2 & \dots & \mu_J \\ . & . & \dots & . \end{bmatrix} \quad (6)$$

In a *DNA* microarray experimental data, they represent a subgroup of genes with same evolution under a subgroup of conditions. From (2), the number of such biclusters (N_b) is given by equation (7).

$$N_b = P_c \quad (7)$$

P_c is the number of distinct columns c_j of the entire A_l whose sum is greater than *0*; that is; $sum(c_j) > 0$. Each distinct column c_j of the entire A_l constitutes the principal column element of the j^{th} bicluster B_j . Therefore, in order for any other column c_m^l of any A_l to belong to the j^{th} bicluster, equation (8) has to be verified: that is the element wise product of the two given column vectors.

$$c_j .* c_m^l = c_j \quad (8)$$

With: *j = 1 to P_c, m = 1 to M, and l = 1 to L*. Algorithm 3 is then used to extract biclusters that have constant values on columns.

4.3.3 Biclusters with constant values on rows

A bicluster with constant values on rows is any submatrix *B (IxJ)* of *A* which has one of the following forms.$B = [a_{ij}]$, with $a_{ij} = \mu + \alpha_i$ additive model or $a_{ij} = \mu.\alpha_i$, multiplicative model. The general form of such biclusters can be represented using equation (9).

$$B = \begin{bmatrix} \dots & \mu_1 & \dots \\ \dots & \mu_2 & \dots \\ \dots & \dots & \dots \\ \dots & \mu_I & \dots \end{bmatrix} \quad (9)$$

In a *DNA* microarray experimental data, they represent a subgroup of conditions that exhibit same evolution under a subgroup of genes. From (2), the number of such biclusters (N_b) is given by equation (10).

$$N_b = P_r \quad (10)$$

P_r is the number of distinct rows r_i of the entire A_l whose sum is greater than *0*, that is; $sum\ (\ r_i\ ) > 0$. Each distinct row r_i of the entire A_l constitutes the principal row element of the i^{th} bicluster B_i . Therefore, in order for any other row r_n^l to belong to the i^{th} bicluster, equation (11) has to be verified: that is the element wise product of the two given row vectors.

$$r_i .* r_n^l = r_i \quad (11)$$

With *i = 1 to P_r, n = 1 to N, and l = 1 to L*. Algorithm 4 is then used to extract biclusters that have constant value on rows.

Algorithm 2

Compute: P_l, r_i^l,r_n^l
For l = 1 to L
  For i = 1 to P_l
    $B_i^l = [];$
    For n = 1 to N
      If $r_i^l .* r_n^l == r_i^l$
      $B_i^l = [B_i^l$; *[Genes(n)* $\alpha_l r_i^l$ *]]*
      End
    End
End; $B_i^l = [[0$ *Conditions];* $B_i^l];$

Algorithm 3

Compute: P_c, c_j, c_m^l
  For j = 1 to P_c
    $B_j = [];$
    For l = 1 to L
      For m = 1 to M
        If $c_j .* c_m^l == c_j$
      $B_j = [B_j$ *[Conditions(m) ;* $\alpha_l c_j$ *]]*
      End
    End
  End; $B_j = [[0$ *Genes]* $B_j];$
End

Algorithm 4

Compute: P_r, r_i, r_n^l
For i = 1 to P_r
  $B_i = [];$
  For l = 1 to L
    For n = 1 to N
      If $r_i .* r_n^l == r_i$
      $B_i = [B_i$; *[Genes(n)* $\alpha_l r_i$ *]]*
      End
    End
  End; $B_i = [[0$ *Conditions];* $B_l];$
End

4.3.4 Biclusters with Coherent Values

A bicluster with coherent values is any submatrix *B (IxJ)* of *A* which has one of the following forms. $B = [a_{ij}]$, with $a_{ij} = \mu + \alpha_i + \beta_j$ additive model or $a_{ij} = \mu.\alpha_i.\beta_j$, multiplicative model. In this study, we will only deal with additive model. $B = [\mu + \alpha_i + \beta_j] = [\mu] + [\alpha_i] + [\beta_j]$ can be viewed as the sum of three

matrices: B_1 with constant values, B_2 with constant values on rows, and B_3 with constant values on columns. Therefore, to obtain biclusters with coherent values from a *DNA* microarray experimental data, the following approach can be used.

Approach: The Gene expression matrix A is first written as the sum of three matrices Z_1 , Z_2, and Z_3 where Z_1 is a matrix with constant values, Z_2 a matrix with constant values on columns and $Z_3 = A - (Z_1 + Z_2)$. Next, use algorithm *4* to extract all biclusters with constant values on rows from Z_3. Next, add them back to their corresponding matches into Z_1 and Z_2 and finally, obtain subgroups of gene with coherent values.

Note that, the choice of the matrix $Z_1 + Z_2$ which has constant values on columns is not arbitrary. It must be constructed using each row of the gene expression matrix A that is also part of the bicluster with coherent values see the bellow property.

Property: Let X be a matrix that contains a bicluster with coherent values embedded within its structure. By subtracting from X a matrix Y that has constant values on columns, and which is constructed using a row of X that is also part of the bicluster with coherent values, the result is a matrix Z that contains a bicluster with constant values on rows embedded within its structure and located at the same address as the bicluster with coherent values. See [13] for proof.

Since we do not have any knowledge about the rows of the gene expression matrix A, we iteratively construct the matrix $Z_1 + Z_2$ which has constant values on columns using each row of A. After each construction, obtain $Z_3 = A - (Z_1 + Z_2)$, use algorithm *4* to extract all biclusters with constant values on rows from Z_3, add them back to their corresponding matches into $(Z_1 + Z_2)$ and obtain biclusters with coherent values.

5. SIMULATION RESULTS AND CONCLUSION

As in [11], we implemented the proposed biclustering algorithm in Matlab and tested it on the yeast gene microarray data that can be found at [12]. The data consists of *2884* genes and *17* conditions. Initially, the data contained $L = 206$ distinct values. We had $b_L = max[a_{nm}] = 595$, $b_0 = min[a_{nm}] = 0$ thus $e = 2.8883$, and $b_l = b_0 + le = 2.8883l$, with $l = 1$ *to* L. After data conditioning, we obtained $L = 111$ new distinct values. Then from our simulation, we obtained $N_b = 10225$ biclusters with constant values, $N_b = 3391$ biclusters with constant values on rows, and $N_b = 836$ biclusters with constant values on columns. Because of the large number of biclusters found, we will present here a few illustrative results that will help the reader grasp the magnitude of the problem and the nature of the results produced by the algorithm. Figure 1 shows an example of biclusters with constant values, biclusters with constant values on rows and biclusters with constant values on columns obtained. Figure 2 shows an example of biclusters with coherent values obtained. A complete discussion of the results can be found in [13]. Finally note that the proposed algorithm has performance advantages over previously reported approaches. The proposed algorithm does not rely on solving an optimization problem. It can be used to search for any type of biclusters defined by the user in a timely manner. After data conditioning which takes approximately *250s*, it takes less than *10s* to get a bicluster. Thus its running time is better than that of [2] which reportedly takes *300-400s* to find a single bicluster. As future work, we will be focusing on the biological meaning of the results obtained.

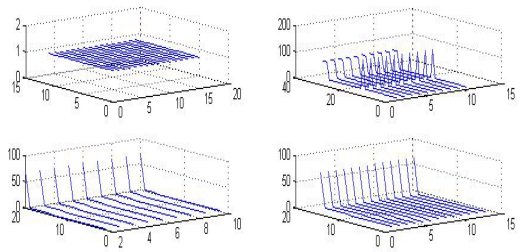

Figure 1: Example of biclusters with constant values, biclusters with constant values on rows, and biclusters with constant values on columns obtained. The x axis represents the conditions, the y axis the genes and z axis the expression level

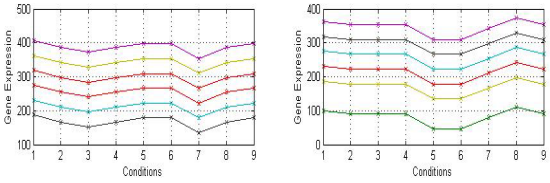

Figure 2: Example of biclusters with coherent values. Each line represents different genes.

6. REFERENCES

[1]- S. C. Madeira, A. L. Oliveira, "Biclustering Algorithms for Biological Data Analysis: A Survey", IEEE Transactions on Computational Biology and Bioinformatics, Vol. 1, No. 1, Jan-March 2004.

[2]- Y. Cheng, G.M. Church, "Biclustering of Expression Data", In Proc. ISMB'00, pages 93-103. AAAI Press, 2000.

[3]- G. Getz, E. Levine, E. Domany, "Coupled Two-way Clustering Analysis of Microarray Data, Proc. Natl. Acad. Sci. USA, 97(22): 12079-84, 2000.

[4]- S. Bergmann, J. Ihmels, N. Barkai, "Iterative Signature Algorithm for Analysis of Large Scale Gene Expression Data. Phys Rev E Stat Nonlin Soft Matter Phys, 67(3 pt 1): 03190201.

[5]- R. Sharan, A. Maron-Katz, N. Arbili, R. Shamir, "CLICK and EXPANDER: a System for Clustering and Visualizing Gene Expression Data", Bioinformatics, 2003.

[6]- Y. Kluger, R. Barsi, JT. Cheng, M. Gerstein, "Spectral Biclustering of Microarray Data: Coclustering Genes and Conditions", Genome Res., 13(4): 703-16, 2003.

[7]- L. Lazzeroni , A. Owen, "Plaid Models for Gene Expression Data", Statistica Sinica, 12: 61-86, 2002

[8]- A. Tanay, R. Sharan, and R. Shamir, "Discovering Statistically Significant Biclusters in Gene Expression Data," *Bioinformatics*, vol. 18, pp. S136-S144, 2002

[9]- O. Alter, P.O. Brown, D. Botstein, "Processing and Modeling Gene Expression Data Using Singular Decomposition", Proceedings SPIE, vol. 4266 (2001), 171-186

[10]- O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein and R. Altman, "Missing Value Estimation for DNA Microarrays", Bioinformatics 17(2001), 1-6.

[11]- A.H. Tewfik, A.B. Tchagang, "Biclustering of DNA Microarray Data with Early Prunning" In Proc, ICASSP 2005.

[12]- S. Tavazoie, J. Hughes, M. Campbell, R. Cho, and G. Church. Yeast micro data set. At http://arep.med.harv*ard.edu/biclustering*

[13]- A. B. Tchagang, A. H Tewfik "Robust Biclustering Algorithm: *ROBA*", Technical Report, University of Minnesota, 2005.