# Bimax (and something else)

Angélica Alejandra Serrano-Rubio,
Amílcar Meneses and Guillermo Morales-Luna
Computer Science Department
CINVESTAV-IPN, Mexico City, Mexico
Email: aserrano@computacion.cs.cinvestav.mx,
{ameneses,gmorales}@cs.cinvestav.mx

Mireya Paredes López
Mathematics Department
CINVESTAV-IPN, Mexico City, Mexico
Email: mireya.paredes@gmail.com

***Abstract*—Differential gene expression analysis and clustering techniques have been current tools to study the relation between a gene and biological processes. Since a group of genes may show co-expression under certain conditions, biclustering techniques have been used to find sets of genes sharing similar expression patterns. We propose a methodology for optimizing the performance of the BIMAX biclustering algorithm using parallel programming techniques. Its performance is evaluated using synthetic datasets.**

***Keywords*—Biclustering, Clustering, Gene expression, High-Performance Computing, Parallelism**

## I. Introduction

Data Mining is useful in the analysis of data structures provided in massive quantities by sophisticated new processing technologies. Bioinformatics focuses on the research and development of new computational methodologies based on computer techniques for organization and analysis of information associated with the "omics" sciences [1]. The organization and analysis of biological data at the level of DNA sequence and RNA generate information related to cellular mechanisms and processes [2]. One of the main aims is the analysis of gene expression [3].

Hybridization-based techniques or *microarrays* in gene expression analysis has achieved high performance in quantifying the level of gene expression. However, such analysis is performed using hypothesis tests, which require a relatively small number of conditions and only genes that have been selected can be parsed.

*Clustering* describes patterns classifying the information by unsupervised methods. *Biclustering* techniques allow the clustering of genes with a similar genetic profile in experimental conditions, thus overcoming the traditional clustering techniques by the simultaneous clustering of genes, diseases and overlap.

In this paper we focus on BIMAX, described by Prelić *et al.* [4], based on *Divide-and-Conquer* strategies to determine optimal biclusters in reasonable time.

In general, biclustering techniques face the same problem as the clustering techniques proposed in the literature, due to the type of information being analyzed, which is characterized by high-dimensional databases. Consequently, robust noise algorithms must be proposed minimizing runtime and showing high-quality solutions. A high-performance computing system is essential to improve the performance of any computational algorithm. We describe the implementation of the parallel BIMAX algorithm. Section II describes the background, and Section III presents BIMAX algorithm.

## II. Background

The main application of Bioinformatics within the biological context is the use of Data Mining techniques for the analysis of information obtained in the study of molecules relevant for life [5]. However, before the application of computational algorithms, it is necessary to adapt and elaborate new models and methodologies that fit the demand of the problem under study [6], [7].

Techniques of clustering and biclustering allow to perform transcriptome analysis for the detection of genes differentially expressed in a set of experimental conditions. The patterns can be identified from datasets through *microarray* experiments, aiming to infer the biological mechanisms modeling the genotype-phenotype relationship and support decision-making. The information from microarray analysis is organized in a *Gene Expression Matrix*

$$W_{m,n} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{pmatrix}. \tag{1}$$

The $i$-th row contains data of a specific gene $g_i$ while the $j$-th column represents a experimental condition $c_j$. Let $G = \{g_1, g_2, \cdots, g_n\}$ and $C = \{c_1, c_2, \cdots, c_m\}$ be the collections of genes and conditions, respectively. For each $i, j$, the value $w_{ij}$ represents the expression level of gene $i$ at condition $j$.

The *gene expression matrix* may contain noise, null values and systemic variations produced during the execution of the experiments. Usually, there are much more genes than conditions. Pre-processing of data is essential before applying any bioinformatic analysis technique, to form hypotheses about the potential pathways of information flow between the involved genes.

### A. Clustering techniques

Clustering techniques aim to group data containing common characteristics, they identify densely populated regions called *clusters*, namely, a partition $\mathcal{U} = \{U_1, U_2, \cdots, U_k\}$ of an universe $U$ is built.

*Gene-based-clustering* obtains functional relations between genes based on their expression levels in comparison with experimental conditions. Such relations consider genes as the data to be grouped and the states as attributes.

Instead, *sample-based-clustering* [8] matches each group of experiments with a phenotype.

A good clustering solution must consider the maximization of homogeneity and separation metrics, which act oppositely. There are many proposals to solve grouping problems, and most of them are NP-hard. Thus heuristics and approximations are used. Five clustering algorithms and their characteristics are summarized in Table I.

TABLE I: Clustering algorithms

| **Algorithm** | **Description** | Ref. |
|---|---|---|
| K-means | Method of vector quantization whose objective is to classify $n$ observations into $k$ clusters. The time complexity is $O(lkn)$ where $l$ is the number of iterations, and $k$ is the number of clusters. | [9] |
| Self Organizing Maps | Algorithm based on competitive learning without supervision to classify a set of nearby observations through a graph. In some cases, it may fail because interesting patterns can be classified in several ways [10] | [11]. |
| CAST | Algorithm based on the notion of *corrupted clique graph* data model. The input data set is assumed to come from the underlying cluster structure with "contamination" due to random errors caused by the complex process of gene expression measurement. | [12] |
| CLICK | Algorithm to identify highly connected components in the proximity graph as clusters using probabilistic assumptions so that two criteria are satisfied: *homogeneity*, due to mates: highly similar to each other; and *separation* due to non-mates: little similarity to each other. | [13] |
| Hierarchical Clustering | This algorithm generates a hierarchical series of nested clusters which can be graphically represented by a tree named dendrogram. | [14] |

Advantages and lacks appear at clustering algorithms when identifying highly correlated gene sets in gene expression. K-means, SOM, and hierarchical clustering have shown high-performance [15] but their purpose is general and they may fail to address particular challenges of gene expression analysis. On the other hand, CLICK and CAST may solve this problem [16].

### *B. Biclustering techniques*

A gene expression matrix $W_{m,n}$ as in eq. (1) can be seen as indexed by the set $G \times C$ where $G = \{g_1, g_2, \cdots, g_n\}$ and $C = \{c_1, c_2, \cdots, c_m\}$ are the sets of considered genes and conditions, respectively. The $ij$-th value $w_{ij}$ represents the expression level of gene $i$ at condition $j$. Accordingly, we will write from now on $W_{GC}$ instead of the previously introduced notation $W_{m,n}$.

For any subsets $F \subseteq G$ and $B \subseteq C$, the submatrix $W_{FB} = (w_{ij})_{i \in F, j \in B}$ consists of the expression levels corresponding to genes in $F$ under conditions in $B$, and can be seen as the gene expression matrix of the *bicluster* $F \times B$ (in the cases in which $F = G$ or $B = C$ it is conventional to refer to *clusters* instead of *biclusters*). A bicluster may satisfy a *homogeneity property*, e.g. it may have constant entries, or constant entries by rows or columns, or have *additive or multiplicative coherent values* or *coherent evolutions*, etc.

The *Biclustering Problem* (BP) receives as input a gene expression matrix $W_{GC}$ and a collection $\mathcal{H}$ of homogeneity properties and the goal is to find a covering of $G \times C$ consisting of maximal biclusters $(F_\iota \times B_\iota)_{\iota \in I}$ such that $W_{GC} = \bigcup_{\iota \in I} W_{F_\iota B_\iota}$ and each bicluster $W_{F_\iota B_\iota}$ satisfies a homogeneity property.

The BP can be hardened by requiring that the cover solution is indeed a partition. Namely, its sets are pairwise disjoint.

BP, in general, is *NP-complete* [17], hence the vast majority of algorithms prioritize the reduction of computing costs, opting for the use of heuristic, stochastic search, divide and conquer and pre-processing techniques to make the patterns of interest more evident.

Most biclustering techniques are used for the analysis of transcriptomic data. The choice of an algorithm will implicitly favor getting a particular type of grouping. Five biclustering algorithms have been selected. Table II gives a brief description of them.

TABLE II: Biclustering algorithms

| **Algorithm** | **Description** | Ref. |
|---|---|---|
| Cheng & Church | This algorithm is the first application of biclustering for analysis of expression profile, and the aim is to find more delicate signals than the clustering algorithm using the Mean Residue Score. | [18] |
| SAMBA | SAMBA emerged as a method of biclustering that produced statistically significant results, and that will also involve a normalization of the expression matrix that represents the essential characteristics of the data. | [19] |
| CTWC | The idea of this algorithm is to identify subsets of genes and conditions so that some of these subsets are used to define new groups that define stable and significant partitions. It should note that the number of submatrices grows exponentially with the size of the problem. | [20] |
| Plaid Models | This algorithm considers a matrix of genes and conditions as a layer superposition, each of them being a subset of rows and columns, which are rearranged to obtain a matrix formed by blocks, where each block is a bicluster | [21] |
| BIMAX | This algorithm is based on the technique of "*divide and conquer*". Specifically, the algorithm begins with a division of the matrix $W_{m \times n}$ into sets of columns based on one of the rows as the reference. Next, the rows or genes rearranged, as they correspond to the different sets of conditions previously obtained. | [22] |

### C. Related work

Improving the computational efficiency of Data Mining algorithms by introducing some method of Parallel Computation proves to be significant as the dimension of the datasets to be analyzed increases. In this sense, several robust parallel methodologies have been proposed that model a biological model under study and also support decision making efficiently.

Metaheuristics are techniques that help in the solution of combinatorial optimization problems. In [23] Gomez-Pulido *et al.* propose the parallelization based on a fine granularity scheme of a biclustering algorithm based on EAs. The methodology indicates the selection of the section that takes the longest computation time, then copies of the implementation will be processed in different processing units. The results showed high efficiency in computing time and power consumption when using reconfigurable hardware instead of multiprocessor architectures.

Lin *et al.* proposed the parallelization of the biclustering algorithm: Large Average Submatrices (LAS) based on the MapReduce technique. Intuitively, the algorithm is organized in two phases: 1) search $k$ rows with the largest sum over the columns, this phase consists of a function *map* and two functions *reduce*; and 2) based on adaptive row search to sum over the columns indicated for each row, then sort all the row sums in sequential order, this phase contains only a map function [24]. This algorithm proved to have a better performance in the quantitative characteristics of each cluster compared to other algorithms such as BIMAX.

On the other hand, Ardaneswari *et al.* propose a method of grouping in two phases to be able to determine a bicluster [25]. During the first step, the parallel algorithm k-means is used to classify a matrix $W_{m\times n}$. In the second phase, the algorithm proposed by Cheng & Church is used. Also, the benefits and limitations related to the design of a parallel biclustering algorithm in GPUs are presented in [26]. This paper proposes the minimization of latency using a coarse grain to maximize energy efficiency and performance through the use of parallel patterns.

Sarazin *et al.* in [27] propose an implementation of the algorithm self-organizing maps using MapReduce with the Spark platform. This application is focused on fault correction, information management and distribution in a distributed architecture. The main idea is based on the initiation of two functions of map-reduce type, which manage the iterations between the rows and the columns.

A parallel algorithm of biclustering must be robust, that is, it must show relevant results in a reasonable amount of time, and that must also be scalable to the target architecture. In the following section, we describe the BIMAX: Binary Inclusion-MAXimal algorithm sequential, which assumes two possible levels of expression: level change, and no difference, concerning a control experiment.

## III. BIMAX: Binary Inclusion-MAXimal

Heuristics are used to solve problems that have proved to be *NP-complete*. The advantages of one algorithm over another may be due to a more suitable optimization method for a specific dataset. In this sense, for the choice of BIMAX, the number of references within the scientific community and the facility to reconstruct the code based on the original publication were considered.

BIMAX algorithm uses a search strategy based on "*divide and conquer*" to determine all optimum maximal biclusters within a reasonable time of a matrix of binary gene expression. Each gene in a condition assumes two possible values: 1 if the gene responds differentially to a condition and 0 if it does not concern a control condition [4].

Let $G = \{g_1, g_2, \cdots, g_n\}$ and $C = \{c_1, c_2, \cdots, c_m\}$ be the sets of genes and conditions.

Find a gene $g_i \in G$, such that the following two condition subsets $B_{i0}$, $B_{i1}$ are non-empty, where for each $k \in \{0, 1\}$ $B_{ik}$ consists of the conditions $c_j \in C$ such that $w_{ij} = k$.

Let $W_{GB_{i0}}$ and $W_{GB_{i1}}$ be the submatrices whose columns are indexed by these sets.

Let us partition the gene (row) index set into three subsets:

$$\begin{aligned}
G_{i0} &= \{g_{i'} \in G| \text{ the restriction of the the } i'\text{-th row of } W_{GB_{ik}} \text{ is constant with value } k, \text{ for some } k \in \{0,1\} \ \} \\
G_{i1} &= \{g_{i'} \in G| \text{ the restriction of the the } i'\text{-th row of } W_{GB_{ik}} \text{ is constant with value } 1-k, \text{ for some } k \in \{0,1\} \ \} \\
G_{i2} &= G - (G_{i0} \cup G_{i1})
\end{aligned}$$

and let $F_{i1} = G \setminus G_{i1}$ be its complement in the gene index set. Form the submatrices

$$W_0 = W_{GB_{i1}} \quad , \quad W_1 = W_{F_{i1}C}.$$

Repeat the procedure to each of these biclusterings, $W_0$ and $W_1$, till the above activation non-emptyness condition fails.

### A. Algorithm (Reference method)

*Algorithm* The following algorithm realizes the divide-and-conquer strategy. Note that individual operations are required for processing the $W_{FB}$ submatrices. The algorithm needs to guarantee that only optimal, i.e., inclusion-maximal biclusters are generated [4].

**Algorithm 1** Procedure Bimax

```
Require: $W_{m,n}$
Ensure: $W_{FB}$
$Z = 0$
$W_{FB}$ =conquer($W_{m,n}, (\{G\}, \{C\}), Z)$)
```

**Algorithm 2** Procedure Conquer

```
Require: $W_{m,n}, (\{G\}, \{C\}), Z$
Ensure: $(\{G\}, \{C\})$
if $\forall\ i \in G, j \in C : w_{ij} = 1$ then
    return $(\{G\}, \{C\})$
end if
$(G_{i1}, G_{i2}, G_{i3}, B_{ik}) = \text{divide}(W_{m,n}, (\{G\}, \{C\}), Z)$
$W_0 = 0, W_1 = 0$
if $G_{i1} \neq 0$ then
    $W_0$=conquer($W_{m,n}, (\{G_{i1} \cup G_{i2}\}, \{B_{i1}\}), Z$)
end if
if $G_{i3} \neq 0 \wedge G_{i2} = 0$ then
    $W_1$ =conquer($W_{m,n}, (\{G_{i3}\}, \{B_{i0}\}), Z$)
else if $G_{i2} \neq 0$ then
    $Z' = Z \cup \{B_{i0}\}$
    $W_1 = \text{conquer}(W_{m\times n}, (\{G_{i2} \cup G_{i3}\}, \{B_{i0} \cup B_{i1}\}), Z')$
end if
return $(W_0 \cup W_1)$
```

**Algorithm 3** Procedure Divide

```
Require: $W_{m,n}, (\{G\}, \{C\}), Z$
Ensure: $G_{i1}, G_{i2}, G_{i3}, B_{ik}$
$G'$ =reduce($W_{m,n}, (\{G\}, \{C\}), Z$)
Choose $i \in G'$ with $0 < \sum_{j \in C} w_{ij} < |C|$
if $i \in G'$ then
    $B_{i1} = j | j \in C \wedge w_{ij} = 1$
else
    $B_{i1} = C$
end if
$B_{i0} = C \setminus B_{i1}$
$G_{i1} = 0, G_{i2} = 0, G_{i3} = 0$
for $i \in G'$ do
    $C^* = \{j | j \in C \wedge w_{ij} = 1\}$
    if $C^* \subseteq B_{i1}$ then
        $G_{i1} = G_{i1} \cup \{i\}$
    else if $C^* \subseteq B_{i0}$ then
        $G_{i2} = G_{i2} \cup \{i\}$
    else
        $G_{i3} = G_{i3} \cup \{i\}$
    end if
end for
return $(G_{i1}, G_{i2}, G_{i3}, B_{ik})$
```

**Algorithm 4** Procedure Reduce

```
Require: $W_{m\times n}, (\{G\}, \{C\}), Z$
Ensure: $G'$
$G' = 0$
for $i \in G$ do
    $C^* = \{j | j \in C \wedge w_{i,j} = 1\}$
    if $C^* \neq 0 \wedge \forall\ C \in Z : C \cap C^* \neq 0$ then
        $G' = G' \cup \{i\}$
    end if
end for
return $G'$
```

## IV. Strategies for parallel computation

## V. Platform architecture and results

## VI. Conclusion

## VII. Acknowledgement

## References

[1] Luscombe, N. M., Greenbaum, D., & Gerstein, M., "What is bioinformatics? an introduction and overview," *Yearbook of Medical Informatics*, vol. 1, no. 1, pp. 83–99, 2001.

[2] Mount, D. W. , "Sequence and genome analysis," *Bioinformatics: Cold Spring Harbour Laboratory Press: Cold Spring Harbour*, vol. 2, 2004.

[3] Griffiths, A. J. F., Gelbart, W. M., Miller, J. H., & Lewontin, R. C. , "Genética moderna. editorial mc graw hill," 2003.

[4] Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., & Zitzler, E., "A systematic comparison and evaluation of biclustering methods for gene expression data," *Bioinformatics*, vol. 22, no. 9, pp. 1122–1129, 2006.

[5] Perezleo Solórzano, L., Arencibia Jorge, R., Conill González, C., Achón Veloz, G., & Araújo Ruiz, J. A. , "Impacto de la bioinformática en las ciencias biomédicas," *Acimed*, vol. 11, no. 4, pp. 0–0, 2003.

[6] Pontes, B., Giráldez, R., Divina, F., & Martínez-Álvarez, F., "Evaluación de biclusters en un entorno evolutivo," *IV Taller nacional de minería de datos y aprendizaje (TAMIDA)*, pp. 1–10, 2007.

[7] Rossi, F., Van Beek, P. & Walsh, T., *Handbook of Constraint Programming*, ser. Foundations of Artificial Intelligence. Elsevier Science, 2006.

[8] Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., ... & Lander, E. S., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *science*, vol. 286, no. 5439, pp. 531–537, 1999.

[9] Levine, J. H., Simonds, E. F., Bendall, S. C., Davis, K. L., El-ad, D. A., Tadmor, M. D., ... & Finck, R., "Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis," *Cell*, vol. 162, no. 1, pp. 184–197, 2015.

[10] Sathishkumar, K., Thiagarasu, V., & Balamurugan, E., "An Analysis on Clustering Based Gene Selection and Classification for Gene Expression Data," *International Journal of Innovative Trends in Engineering (IJITE)*, vol. 11, no. 01, pp. 55–60, 2015.

[11] Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., ... & Golub, T. R., "Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation," *Proceedings of the National Academy of Sciences*, vol. 96, no. 6, pp. 2907–2912, 1999.

[12] Bellaachia, A., Portnoy, D., Chen, Y., & Elkahloun, A. G. , "E-cast: a data mining algorithm for gene expression data," in *Proceedings of the 2nd International Conference on Data Mining in Bioinformatics.* Springer-Verlag, 2002, pp. 49–54.

[13] Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., ... & Mortazavi, A., "A survey of best practices for RNA-seq data analysis," *Genome biology*, vol. 17, no. 1, p. 13, 2016.

[14] Galili, T. , "dendextend: an R package for visualizing, adjusting and comparing trees of hierarchical clustering," *Bioinformatics*, vol. 31, no. 22, pp. 3718–3720, 2015.

[15] Weber, L. M., & Robinson, M. D. , "Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data," *Cytometry Part A*, vol. 89, no. 12, pp. 1084–1096, 2016.

[16] Salazar, G., Bellocchi, C., Todoerti, K., Saporiti, F., Piacentini, L., Scorza, R., & Colombo, G. I. , "Gene expression profiling reveals novel protective effects of Aminaphtone on ECV304

endothelial cells," *European journal of pharmacology*, vol. 782, pp. 59–69, 2016.

[17] Pontes, B., Giráldez, R., & Aguilar-Ruiz, J. S., "Biclustering on expression data: A review," *Journal of biomedical informatics*, vol. 57, pp. 163–180, 2015.

[18] Cheng, Y., & Church, G. M., "Biclustering of expression data." in *Ismb*, vol. 8, 2000, pp. 93–103.

[19] Fiannaca, A., La Rosa, M., La Paglia, L., Rizzo, R., & Urso, A., "Analysis of mirna expression profiles in breast cancer using biclustering," *BMC bioinformatics*, vol. 16, no. 4, p. S7, 2015.

[20] Wani, M. A., & Riyaz, R. , "A novel point density based validity index for clustering gene expression datasets," *International Journal of Data Mining and Bioinformatics*, vol. 17, no. 1, pp. 66–84, 2017.

[21] Oghabian, A., Kilpinen, S., Hautaniemi, S., & Czeizler, E., "Biclustering methods: biological relevance and application in gene expression analysis," *PloS one*, vol. 9, no. 3, p. e90801, 2014.

[22] Roy, S., Bhattacharyya, D. K., & Kalita, J. K. , "Analysis of gene expression patterns using biclustering," *Microarray Data Analysis: Methods and Applications*, pp. 91–103, 2016.

[23] Gomez-Pulido, J. A., Cerrada-Barrios, J. L., Trinidad-Amado, S., Lanza-Gutierrez, J. M., Fernandez-Diaz, R. A., Crawford, B., & Soto, R., "Fine-grained parallelization of fitness functions in bioinformatics optimization problems: gene selection for cancer classification and biclustering of gene expression data," *BMC bioinformatics*, vol. 17, no. 1, p. 330, 2016.

[24] Lin, Q., Xue, Y., Chen, W., Ye, S., Li, W., & Liu, J., "Parallel large average submatrices biclustering based on MapReduce," in *Computational Intelligence and Security (CIS), 2015 11th International Conference on.* IEEE, 2015, pp. 134–137.

[25] Ardaneswari, G., Bustamam, A., & Siswantining, T. , "Implementation of parallel k-means algorithm for two-phase method biclustering in Carcinoma tumor gene expression data," in *AIP Conference Proceedings*, vol. 1825, no. 1. AIP Publishing, 2017, p. 020004.

[26] Orzechowski, P., & Boryczko, K., "Effective biclustering on GPU-capabilities and constraints," *Prz Elektrotechniczn*, vol. 1, pp. 133–6, 2015.

[27] Sarazin, T., Lebbah, M., & Azzag, H., "Biclustering using Spark-MapReduce," in *Big Data (Big Data), 2014 IEEE International Conference on.* IEEE, 2014, pp. 58–60.