Proceedings of 2010 International Conference on Systems in Medicine and Biology
16-18 December 2010, IIT Kharagpur, India

A Comparison of Biclustering Algorithms

Nishchal K. Verma[1], Sheela Meena[5]
Department of Electrical Engineering,
Indian Institute of Technology, Kanpur.
[1]nishchal@iitk.ac.in, [5]sheelameena11@gmail.com

Amarjot Singh[3]
Department of Electrical and Electronics Engineering, NIT Warangal
Warangal-506004, India
[3]amarjotsingh@ieee.org

Yan Cui[6]
Center for Integrative and Translational Genomics, Dept of Molecular Sciences, University of Tennessee,
Memphis, USA
[6]ycui2@uthsc.edu

Shruti Bajpai[2]
Department of Biomedical Engineering, IET, Bundelkhand University, Jhansi.
[2]shruti.2bmd@gmail.com

Aditya Nagrare [4]
Department of Biotechnology Engineering, NIT Warangal
Warangal-506004, India
[4]adit.nagrare@gmail.com

***Abstract*— In the past years, various microarray technologies have been used to extract useful biological information from microarray data. Microarray technologies have become a central tool in biological research. The extraction or identification of gene groups with similar expression pattern, plays an important role in the analysis of genes. The primary techniques involve clustering and biclustering methods. Besides classical clustering methods, biclustering is being preferred to analyze biological datasets, due to its ability to group both genes across conditions simultaneously. Biclustering is being practiced in a number of applications to club genes across specified conditions, used mainly in identifying sets of coregulated genes, tissue classification etc. Gene Ontology is another important area of application, where biclusters are used to presume the class of non-annotated genes. Gene Ontology database is competent of annotating and analyzing a large number of genes. Gene Ontology is a standard approach of representing the gene with their product attributes, across different species and databases. Typical annotations for the analyzed list of genes can be well understood using the BicAT and BiVisu toolbox. The toolbox provides a platform which enables us to compare different biclustering algorithms, inside the graphical tool. This paper compares different biclustering approaches used to analyze carcinoma and DLBCL (diffuse large B-cell lymphoma) microarray datasets. The algorithms were compared on the grounds of enrichment values with support from runtime analysis. The paper explains in detail the biclusters associated with each algorithm and the intellects affecting the enrichment values, leading to the best biclustering technique for the datasets mentioned above.**

Keywords- Biclustering, Gene Ontology, BicAT, Microarray.

I. Introduction

Advances in microarray technology have motivated the developments of several techniques applied to analyze biological data. These methodologies play a fundamental tool in the field of biological research. A number of algorithms have been designed for the convenience of scientists to analyze different kind of genes under diverse states and experimental samples. The gene expression data generated by the microarray technologies arrange themselves as a matrix with genes as rows and experimental conditions as columns. Clustering [1] and biclustering [2] are two primary techniques, extensively used to extract useful information from the microarray datasets. Clustering is defined as the grouping of genes behaving similarly under specified experimental conditions. Conventional clustering techniques divide an expression matrix into smaller sub-matrices which extend over the whole set of conditions, endowing equal weightage to all conditions. A question of biological interest would be constrained, if all genes are assumed to behave similarly over all conditions. Thus, we aim to look for subset of genes which show similar behavior under a subset of conditions, for example, a cellular process which is active only under certain specific conditions. It is an extremely useful tool used effectively in a number of applications and moreover, clustering algorithms have certain limitations, as the cluster formed hides the finest knowledge about the genes. Many important characteristics of several useful genes are ignored due to broader approach and inability of these algorithms to reach up to finer level.

Thus, researchers introduced biclustering as a new technique, capable of clustering the dataset in both dimensions simultaneously. Biclustering is an extremely useful data mining tool used for identifying patterns, where different genes are correlated based on the subset of columns in the gene expression dataset. This methodology is effectively applied to extract finer details about the behavior of genes under certain experimental samples. A number of authors have used biclustering to study yeast [3], [8] and human gene expression datasets [3], [12]. Biclustering approach has been applied mainly in the field of (i) Identification of corregulated genes, (ii) Gene functional annotation, and (iii) Sample classification.

In the past recent years several studies have been made on the comparison and evaluation of the clustering (one dimensional clustering) and biclustering methods (two dimensional clustering). A large research has been carried out to analyze the results obtained from different algorithms. These studies used different indices along with several validations for the quantitative analysis of the results. These indices were divided into three broad categories namely (i) internal indices (based on the input data) (ii) external indices, referring to the

978-1-61284-038-3/10/$26.00 ©2010 IEEE

procedure where the substantiation can be done by consulting genes based on the similar regulation mechanism (iii) relative indices, measuring the relation of the input parameter setting with the clustering outcome.

Biclustering methods have been proposed and validated by a number of papers, while only a few papers focus on the comparison between different biclustering techniques. Majority of the papers make a comparative study of only biclustering techniques with the clustering techniques or singular value decomposition theorem. We have come forth with the paper to provide evaluation and comparison of the systematic biclustering methods. The papers looks forward to answer questions related to the working of algorithms in different environments and conditions and discusses their computational complexities in short. Biclustering is a complex process formulated with certain complexities, to solve completely or find a complete solution to the problem. The complexity of the problem depends upon the exact formulation, the Biclustering approach and the merit function used to evaluate the quality of the Bicluster. NP stands for Non-deterministic Polynomial time (NP). NP-complete and NP-hard are the two terms that explain the complexity arising in solving the problem formulated during Biclustering as shown in fig.1.

The aim is to make the results comprehensive in nature and independent to the sensitive order of input datasets [3]. This paper namely compares six biclustering methods Cheng Church, ISA, xMotifs, Bimax and OPSM applied using BicAT [4] toolbox and Parallel Clustering Plot algorithm (Split and Merge) using BiVisu[5] toolboxes:

(a) BicAT is a graphical platform used for data analysis utilizing various clustering and biclustering methods. The toolbox provides the facility for data normalization, discretization, filtering the bicluster across a specific condition or gene pair analysis for bipartite graphs.

(b) BiVisu is a software tool with an interactive graphical user interface (GUI) used to implement parallel coordinate plot biclustering algorithm. It is used to analyze, refine and visualize the detected biclusters in a 2D setting in a convenient way.

In general, it is extremely difficult to make a genuine comparison of the biclustering approaches due to different problem formulation used by every algorithm which may fit to the fullest for one data scenario and fail completely to give any results in the other. Enrichment value is an effective way to compare the performance of different algorithms. An effective and standard way of measuring this functional coherence, or enrichment, is to compute p−value comparison for a pattern to be enriched by a given functional class. For a given bicluster, the ratio of the number of genes specified in a category to the number of genes in the bicluster provides a possible enrichment value. The ratio varies from a minimum value zero to a maximum value 1. We judge the performance of the algorithms for p=0.0001 to p=0.1. Thus, the lower this p−value, the more functionally enriched this gene group is with this class. Finally, the results obtained are enriched using web based GO enrichment [13]. The comparison was made by taking enrichment values as reference theme with the essential assistance from runtime analysis, for all biclustering methods. Gene Ontology (GO) component allows the exploration of the Gene Ontology (GO) terms represented within a list of genes. It is an extremely reliable and useful index for the standardized representation of genes and their product attributes.

The remaining paper is divided and elaborated into following sections. The algorithms are explained in section II while Section III explains the necessary preprocessing step required to run the algorithm on the datasets. Section IV discusses the results obtained after extraction of biclusters by different algorithms, followed by the enrichment results. The final section concludes the results and the states the best biclustering technique for the underlined datasets.

II. Algorithms

A. *Cheng and Church:*

The phenomenon of biclustering used to analyze gene expression data was firstly introduced by Cheng and Church, as an optimization problem in the algorithmic framework [3]. The algorithm aims to extract biclusters followed by solving the restricted optimization problem defined by the respective scoring function. This algorithm works on greedy iterative search method, based on the idea of maximizing the local gain by adding or removing rows or columns from the bicluster. The algorithms in general fail to find the globally optimal solution as they do not operate exhaustively on all datasets. These algorithms are widely used as the computation time is decreased drastically by these methods.

The algorithm considers a gene expression data matrix $B(U,V)$. Many subset averages related to the input data matrix are essentially required for the algorithm and are computed as mentioned below. The row subset average denoted by b_{uV} over the row subset U is:

$$b_{uV} = \frac{\sum_{v\varepsilon V} b_{uv}}{|V|} \tag{1}$$

Similarly, the column subset average b_{Uv} over V subset is:

$$b_{Uv} = \frac{\sum_{u\varepsilon U} b_{uv}}{|U|} \tag{2}$$

The sub-matrix average, b_{UV}, of all the rows and columns of the gene expression matrix is:

$$b_{UV} = \frac{\sum_{u\in U, v\in V} b_{uv}}{|U||V|} \tag{3}$$

The residual score of an element in the sub-matrix is:

$$RS_{UV}(u,v) = (b_{uv} - b_{Uv} - b_{uV} - b_{UV}) \tag{4}$$

Mean square residue score of entire sub-matrix is:

$$MSR(U,V) = \frac{1}{|U||V|} \sum_{u\in U, v\in V} (RS_{uv}^{2}) \tag{5}$$

The score for the row and column mean is evaluated by equation (6) and (7) respectively.

$$d(u) = \frac{1}{|V|} \sum_{v\in V} RS_{U,V}(u,v) \tag{6}$$

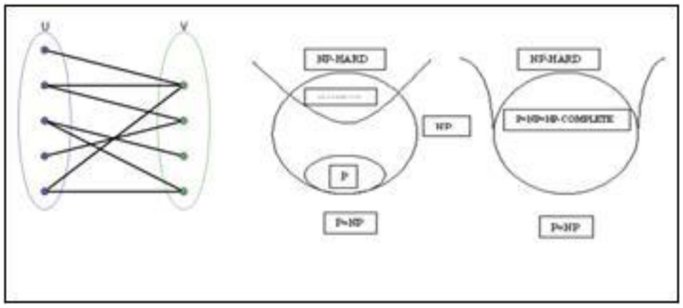


Fig. 1 Algorithm complexity

$$f(v) = \frac{1}{|U|} \sum_{u \in U} RS_{U,V}(u,v) \tag{7}$$

In the next step, the rows $d(u)$ or columns $f(v)$ having the higher value are deleted, if the MSR is greater than the threshold, δ . In the second phase the rows and columns are being added looking for the lowest mean squared residues at each move and terminating where none of the moves increase the matrix size without crossing the threshold, δ .This iterative algorithm on convergence results into δ -biclusters, having low mean squared residue and locally maximal size. The bicluster elements are masked with randomly generated uniform values in the original matrix, for usage in next iteration.

B. Iterative Signature Algorithm (ISA):

The algorithm [6], [7] described here uses the normalized copies of gene expression matrix. Normalization plays a very important role in iterative signature algorithm. The matrices M^U and M^V contain the rows and columns normalized to mean 0 and variance 1 respectively. The mean expression of genes from V' in the sample u is denoted by $e^U_{uV'}$, similarly the mean expression of gene v in samples from U' is denoted by $e^V_{U'v}$. A bicluster $D = (U', V')$ is required to have:

$$\begin{aligned} U' &= \{u \in U : \left| e^V_{uV'} \right| > T_V \sigma_V\} \\ V' &= \{v \in V : \left| e^U_{U'v} \right| > T_U \sigma_U\} \end{aligned} \tag{8}$$

where T_U is termed as the threshold parameter for the row set U' and σ_U is termed as the standard deviation of means (v ranges over all possible genes and U' is fixed). In a similar way for the conditions, T_V and σ_V are the corresponding parameters for the column set V' . Next, to initiate an algorithm, we assume an arbitrary set of genes $V_\alpha = V_{in}$ (it may be randomly generated). The algorithm works iteratively and applies the following update equation:

$$U_i = \{u \in U : \left| e^V_{uV_i} \right| > T_V \sigma_V\} \tag{9}$$

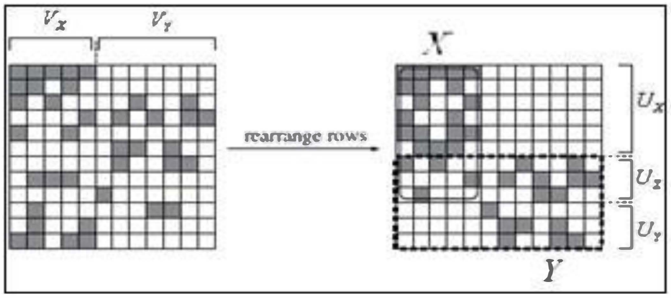


Fig. 2: showing arrangement of columns in bicluster extrationmethod through Bimax algorithm.

$$V_{i+1} = \{v \in V : \left| e^U_{vU_i} \right| > T_U \sigma_U\} \tag{10}$$

The iterations are terminated when the below mentioned equation is satisfied given an arbit number l :

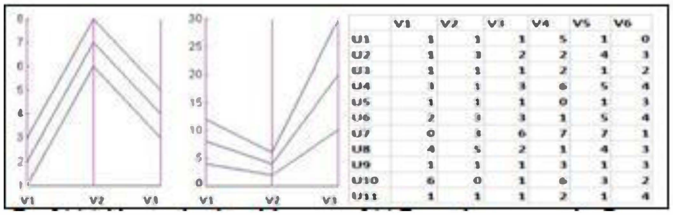
$$\frac{\left| V_{l-i} / V_{l-i-1} \right|}{\left| V_{l-i} \cup V_{l-i-1} \right|} < \varepsilon \tag{11}$$

The algorithm at last converges to an approximate fixed point called bicluster. To obtain a set of biclusters ISA can be made to run again on the data matrix using different initial conditions and varied thresholds.

C. Order Preserving Sub-matrices (OPSM):

Order preserving sub-matrices algorithm was first proposed by *Ben-Dor et al* [8]. OPSM as the name suggests extracts the biclusters, with columns organized in a monotonically increasing order. The algorithm aims to search for a biological progression of size $\hat{k}$ by $\hat{s}$ (sub-matrix $D(U',V')$) hidden in the data matrix, $B(U,V)$, of the order $m \times n$ with rows U' and columns V' having the linear ordering. The complete model can be defined as a pair (V', π) , where $\pi = \left(v'_1, v'_2, v'_3 \ldots v'_{\hat{s}}\right)$ is a linear ordering of $\hat{s}$ columns in V'. The model (V', π) is supported by a row, if the $\hat{s}$ corresponding values are ordered according to the π monotonically increasing permutations. An unknown order preserving matrix of the order (U', V') has been planted in a gene expression dataset which are modeled by a random data matrix. The steps involved in the process of generating a data matrix with a planted order preserving sub matrix are stochastic in nature. In the first step, the random indices for planted rows and columns are selected. Secondly the random ordering of the planted rows is chosen and ranks are assigned randomly to the data matrix in a way, which is in consistency with the planted sub-matrix. The planted sub matrix $D(U',V')$ is determined along with the data matrix at the completion of above three steps.

The complete model thus defined is supported by a row if two conditions are satisfied (i) the $\hat{s}$ corresponding values are ordered according to the permutation π (ii) they should be monotonically increasing.



	V1	V2	V3	V4	V5	V6
U1	1	1	1	5	1	0
U2	1	1	2	2	4	3
U3	1	1	1	2	1	2
U4	1	1	3	6	5	4
U5	1	1	1	0	1	3
U6	2	3	3	1	5	4
U7	0	1	6	7	7	1
U8	4	5	2	1	4	3
U9	1	1	1	3	1	3
U10	6	0	1	6	3	2
U11	1	1	1	2	1	4

Fig. 3(a)Additive related model (b) Multiplicative model (c) Example matrix used in pc plot

As it becomes really difficult to check all the complete models thus partial models discovered from the data matrix are grown iteratively until they get converted into complete models. The indices of the $\hat{a}$ "smallest" elements $< v'_1 .....v'_{\hat{a}} >$ and the indices of the $\hat{b}$ "largest" elements $< v'_{m-(\hat{b}-1)} .....v'_m >$of a complete model and its size m is specified by a partial model of order$(\hat{a},\hat{b})$. When defining partial models the OPSM algorithm focus on the columns at the extremes of the ordering as here it is assumed that these columns are more useful in identifying the target rows,(the rows that support the assumed linear order). The algorithm starts by evaluating all (1,1) partial models and keeping the best k of them. It expands until it gets $(\hat{s}/2,\hat{s}/2)$ models, which are said to be complete models. It then outputs the best one amongst all the models.

D. Bimax Algorithm:

Bimax algorithm [9] is basically an exhaustive divide and conquer strategy based algorithm used in order to extract biclusters from gene expression data matrix. This method preprocesses the data matrix to convert it into a binary matrix by fixing a threshold, the transcription levels above this threshold becomes one and below it becomes zero(or vice versa). This will lead to formation of sub-matrices with constant values of ones in order to get the up/down regulated conditions and the biclusters following additive model, where the expression values vary over a set of conditions. Concerning the bicluster structures, two scenarios are considered: multiple biclusters without overlap in any dimensions and multiple biclusters with overlap. The algorithm needs to make sure that only inclusion-maximal or optimal biclusters are formed.

Fig. 2 shows the working of bimax algorithm. The rows are added one by one to form the main matrix. The column set is partitioned into V_X -the columns in which the new row has ones, and its complementV_Y . The row set is split into U_X - the rows that have only ones inV_X , U_Y those that have ones in V_Y only, and U_Z -those that have ones in both. Let U be the sub-matrix $(U_X \cup U_Z, V_X)$ and V be the sub-matrix $(U_Z \cup U_Y, V_Y)$, respectively.

The problem arises when overlap (between sub-matrices X and Y) occurs and as consequence we need to ensure that the algorithm only considers those biclusters in Y that extend over V_Y . The parameter Ψ serves this goal, which contains the sets of columns that restricts the number of admissible biclusters. A bicluster is called as admissible, if the bicluster shares one or more columns with a column set V in Ψ , i.e., $\forall \mathrm{V}^+ \in \Psi : V \cap V^+ \neq 0$.

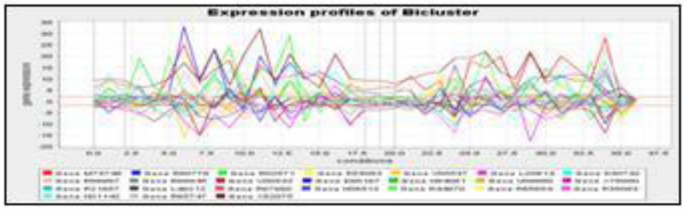


Fig. 4(a) Profile of bicluster obtained from Cheng & Church algorithm.

The incremental algorithms to find all inclusion-maximal cliques containing the nodes works by visiting each node in the input graph. Every bicluster found by iteration through all other nodes of the graph is globally extended to its maximum. Iteration is carried either across the row or column and hence all hidden modules are found out from the matrix due to its effective design.

E. xMotif Algorithm:

xMotif algorithm was suggested by Murali and Kasif[10]. The level of expression of genes is maintained by conserved gene motifs, xMotifs, addressed to a subset or a part of genes which are simultaneously conserved across a subset of samples. The gene's level of expression is said to be conserved if the gene is expressed in almost same abundance in all the samples. With the aim of algorithm to identify the largest motif from the gene expression matrix, a motif is defined for every gene that makes our approach over specific on the other hand too many genes make it restrictive. Thus, we can define an xMotif with the number of samples present in a subset (of samples equal to r_1 fraction of all the samples and for every gene not present in a subset (of genes), the gene is conserved to almost r_2 fraction of the samples present in a subset of samples. Consistent to the fact that a gene sample can appear in more than one motif but the samples which have earlier appeared in one motif are not omitted from the gene expression data.

Initially consider the null hypothesis that gene expression values are generated by a uniform distribution. 'λ' value is then to be evaluated and only those values with λ less than the considered parameter is selected. The genes corresponding to these values yield intervals containing large number expression values with the removal of extra values. On the onset of algorithm a set of genes, a set of conditions, each gene sample pair expression value and a list of intervals representing the states in which the gene is expressed in the sample is taken as input. A set of conserved genes, their states, and a set of samples matching the motif are the prerequisites for determining an xMotif. In short we can also say that with all these prerequisites given we can compute the xMotif by

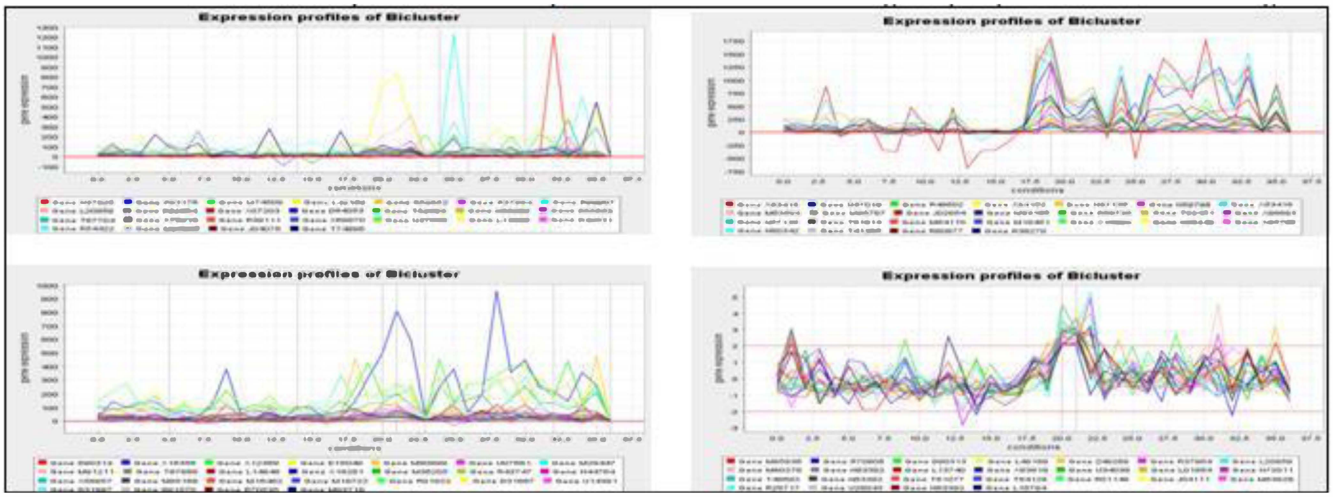


Fig. 4(b) Profile of bicluster obtained from xMotif algorithm (c) Profile of bicluster obtained from OPSM algorithm (d) Profile of bicluster obtained from ISA algorithm (e) Profile of bicluster obtained from Bimax algorithm

simply checking for each sample c' whether the genes that are conserved are in the same state in c and c'. Here c is considered as a seed from which whole of the xMotif can be computed. A sample c matching the largest xMotif is considered along with the discriminating set of samples with following properties (a) there is only one state in which a gene can be in c and c' to be in the largest xMotif and (b) if the gene is not in the same state in c and c' then it is not said to be contained in the largest xMotif.

With the seed sample and discriminating set given our largest xMotif comprises of the samples that agree with c on all the gene states that satisfy these conditions. An xMotif for every single class of gene is different that makes it difficult to analyze so we try to find a motif of maximum range. This motif covers maximum gene and is referred as best motif.

F. Parallel Coordinate Algorithm:

An effective tool known as parallel coordinate plot technique [11] is formulated to visualize and analyze the biclusters present in the biological gene expression as well high dimensional data. According to this technique, the geometrical property of the data is preserved, though the orthogonal property is destroyed resulting into unrestricted number of dimensions. The technique yields a picture of a high dimension data on a one dimensional plane where all the axes are arranged parallel to each other. Two types of biclusters obtained are: the additive related bicluster and the multiplicative related bicluster as shown in fig. 3(a) and 3(b). The additive related bicluster on a parallel coordinate plot can be identified as a single clustered point on the lines having the same slope across a set of conditions in the bicluster. The multiplicative related bicluster can be located as a single clustered point on an overlapped line obtained on the same line obtained on the same plot. Since, all the biclusters are represented as single point or a structure, thus the task has been simplified as to find hidden biclusters in a pc-plot.

In the parallel plot method, every two columns of the gene expression matrix are compared to each other to find the correlated columns. Let us consider a gene expression matrix, refer fig. 3(c), containing six columns namely $V1, V2, V3, V4, V5, V6$ and eleven rows starting from $U1$ till $R11$ in the gene expression matrix. Values are assigned to all the rows and columns respectively. Consider columns $V3$ and $V5$ in order to calculate the biclusters in the input data matrix. Clustered points are being looked for, the values received after taking the difference between $V3$ and $V5$. If these clustered points repeat more than once, the points can be referred to as called a bicluster. A valid bicluster must necessarily contain more than one row, i.e. a bicluster containing only one row is not considered as a valid bicluster. Suppose by amalgamating columns $V3$ and $V5$ two valid biclusters are obtained from the data matrix. Likewise the analysis is extended to continuous investigation whether any other columns can be merged to $(V3, V5)$. For example: taking either $V3$ or $V5$ as reference and find out that any of the columns amongst $V1, V2, V4$ and V6 can be merged to $(V3, V5)$ examining the first difference matrix it is analyzed that two paired columns, $(V1\text{-}V3)$ and $(V2\text{-}V3)$ show a single clustered point with difference value equal to zero. This suggests that columns " $V1$ " and " $V2$ " can be merged to $(V3, V5)$ for rows $U1, U3, U5, U9$ and $U11$.The second difference matrix also has a clustered point with value equal to one. From here it is concluded that $V6$ can also be merged with $(V3, V5)$ for rows $U2, U4, U6, U8,$ and $U10$. Thus by the above mentioned merge and split procedure, which is mainly merging of the paired columns and splitting of the rows , the hidden biclusters in the expression matrix or the given data matrix can be identified.

III. Preprocessing Step

In the domain of gene expression data analysis, extracting data from Boolean matrices has been found to be highly efficient and promising. The preprocessing steps are crucial for the quantity and relevance of extracted patterns. The preprocessing of the data can be done in the following two ways: (1).*Discretization:* Discretization is used to convert continuous

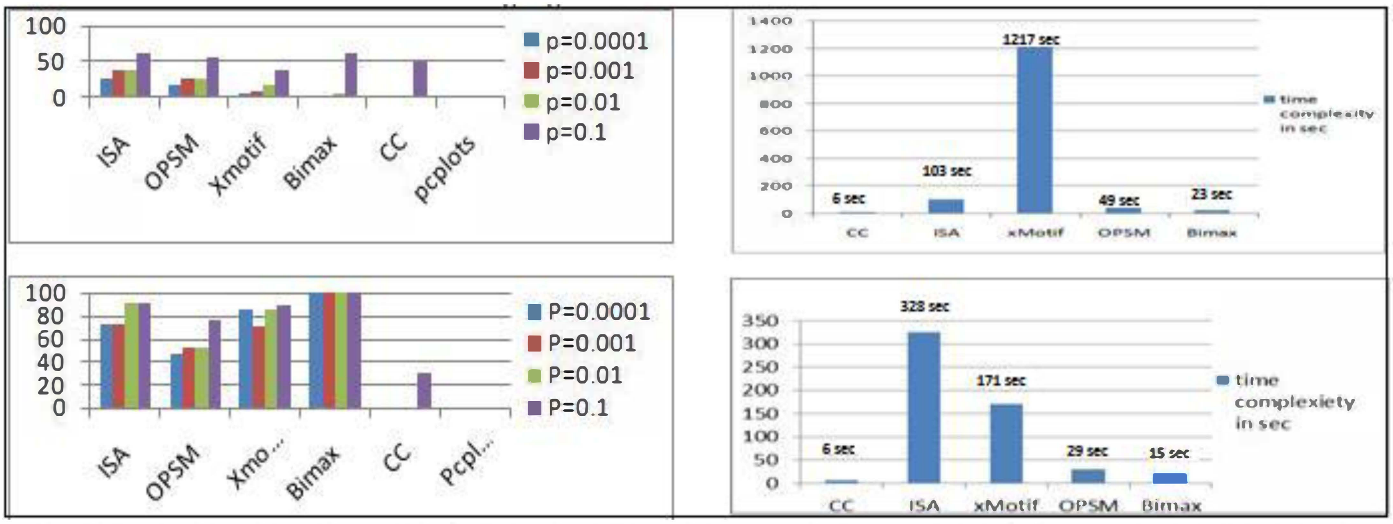


Fig. 5(a): Bar graph showing p-value categorization on Carcinoma dataset (b) Bar graph showing p-value categorization on DLBCL dataset (c) Bar graph showing running-time (in sec) on Carcinoma dataset (d) Bar graph showing running time(in sec) for DLBCL dataset.

models into discrete parts. It is an essential step applied before a biclustering algorithm is practiced, in order to improve the quality of the bicluster to be extracted. In some cases, especially in bimax algorithm, the discretization of the data to binary values becomes essential, to extract biclusters.

(2).*Normalization:* Similar to the discretization, normalization also plays an important role in obtaining the biclusters from the datasets. It is an systematic procedure that

ensures, that the database is free from undesired properties like insertion, deletion etc.. Bimax algorithm is an important example of normalization where the dataset needs to be normalized before the algorithm is applied.

IV. Results

The data derived from the aforementioned artificial model enable us to investigate the capability of the methods to recover known groupings. This section elaborates the results obtained after applying six different biclustering algorithms on carcinoma and DLBCL (diffuse large B-cell lymphoma) datasets available at [14]. The biclustering algorithms are applied using BicAT and BiVisu toolboxes, followed by web based GO-enrichment.

The biclusters say a lot about the regulatory mechanism and the classification of genes. The bicluster shown in fig. 4(b) represents expression profile for the bicluster obtained from xMotif applied to carcinoma dataset with reference to a common theme (Expression profile). The profile of biclusters involves the most significant 25 genes classified across few specific conditions. The bicluster shows 3 sharp peaks L49169, R99907 and H37925 at conditions 21, 25 and 31.5 respectively, along with many small ones deviating from the flatter expression value. Similarly for the expression profile obtained from ISA, a number of sharp peaks are observed for gene X53416, with peaks at 18.5 and 30 showing the maximum variation from the flatter region.

For the biclusters extracted from bimax algorithm, we can observe three sharp peaks U34038, L20859 and M60278 while for OPSM we get 2 relatively sharp peaks for gene X16356 at 21.5 and 28.5 respectively. The bicluster obtained from CC algorithm is extremely large in size and has a number of peaks deviating from the common reference theme. The peaks for R80778, X52075, R44970 and R97980 show the maximum variation with respect to the reference theme. The variations experienced by the bicluster in case of CC, ISA and bimax algorithm are more as compared to OPSM and xMotif since the expression profile shows a number of peaks above the reference level.

The GO enrichment results for the extracted biclusters for DLBCL (diffuse large B-cell lymphoma) dataset and carcinoma dataset are graphed in fig. 5(b) and 5(a). The fig. 5(a) shows the enrichment results obtained for carcinoma dataset with the proportion of biclusters for a number of overrepresented GO categories at different precision levels. The best enrichment results are retrieved for bimax (>55% for p=0.1) followed by ISA algorithm (>50% for p=0.1), proving the presence of large number of enriched constant biclusters in the dataset. The results obtained from xMotif algorithm (>40% for p=0.1) represents a large portion of the coherent value of the biclusters in the reference dataset. OPSM (>50% for p=0.1) also shows significant amount of enrichment with a drop for xMotif algorithm (>40% for p=0.1), showing the presence of a maximum number of biclusters with coherent row values on their rows in the dataset. CC (=30% for p=0.1) gives low enrichment results as mentioned above. The lowest numbers of biclusters were obtained in pcplot algorithm (file of zero cells) with lowest enrichment levels. Fig. 5(c) shows the running time analysis for the carcinoma dataset. Among all the algorithms, CC, takes the least time with 6 sec followed by bimax algorithm which takes 23 sec for the completion. OPSM is third fastest with 49 sec, followed by ISA at 103 sec respectively. xMotif takes maximum time of 1217 sec for completion[1]. The GO enrichment results evaluated for DLBCL (diffuse large B-cell lymphoma) is graphed in fig. 5(b). The best results retrieved here are for bimax (>95% fot p=0.1) and ISA (>90% for p=0.1) for all the precision values, this analysis

[1] All the algorithms tested on Intel Pentium Processor at 2GHz, 800MHz FSB and 2GB RAM.

shows that it contains a large number of constant type bicluster that are functionally enriched. xMotif (>80% for p=0.1) has slight advantage over OPSM (>70% for p=0.1) with CC(>20% for p=0.1) showing low value of enrichment. Pcplot algorithm shows no enrichment even for this dataset. Fig. 5(d) shows the run time analysis for different biclustering algorithms applied to DLBCL dataset. Among these algorithms, CC takes the least time with 6 sec, followed by bimax algorithm which takes 15 secs for completion. OPSM is third fastest with 29 sec while xMotif followed with 171 sec. ISA takes maximum time for completion, 328 sec. Cheng church is the fastest algorithm as the size of the biclusters obtained is as large as the original matrix.

Optimality of the bicluster is tested using a polynomial time algorithm [15] tested for all the algorithms mentioned in the paper. The algorithm used has the following advantages: (1) no discretization procedure is required, (2) performs well for overlapping bi-clusters and (3) works well for additive bi-clusters. The same parameters used in the reference are used for the optimality. The results obtained from these algorithms are compared with the results obtained from the above algorithms, it was found that the biclusters obtained by these algorithms were optimal and statistically significant in terms of enrichment when checked with GO annotation.

V. Conclusions

The present study compares six prominent biclustering methods with respect to their capability of identifying groups of (locally) co-expressed genes. To this end, different microarray datasets corresponding to different notions of biclusters as well as real transcription profiling data are considered. The paper focuses on explaining the basic concepts of biclustering with a number of algorithms related to the approach. The results for six different biclustering algorithms applied to Carcinoma and DLBCL (diffuse large B-cell lymphoma) datasets are analyzed, using enrichment criteria as benchmark. We compare the efficacy of the range support patterns discovered from microarray data with biclusters produced by Cheng and Church's algorithm for discovering constant row/column biclusters, ISA, a commonly used biclustering algorithm, using the mean squared error (MSE) coherence measure, OPSM, used to extract patterns with a specific ordering, XMotif extracts conserved gene motifs while bimax is famous for obtaining biclusters by divide and conquer methodology, and their functional enrichment in terms of GO biological process annotations. The key results are as follows: There are significant performance differences among the six biclustering methods. On both the datasets, ISA and Bimax provide similarly good results: a large portion of the resulting biclusters is functionally enriched. In the context of the DLBCL and carcinoma datasets, Bimax provides a high proportion of enriched biclusters as compared to other algorithms. On the other hand, ISA can be used to find multiple biclusters with both constant and coherently increasing values while OPSM is mainly tailored to identify a single bicluster of the latter type. The remaining two algorithms CC and xMotif, both tend to generate large biclusters that often represent gene groups with unchanged expression levels and therefore not necessarily contain interesting patterns in terms of, e.g. co-regulation.

Accordingly, the scores for CC and xMotif are significantly lower than that for the other biclustering methods under consideration. The percentage of enrichment is maximum for Bimax and ISA algorithms. The enrichment value falls for OPSM and xMotif with a minimum for Cheng Church algorithm. The enrichment value for Cheng Church algorithm is also low as compared to other algorithms because of the large sized extracted biclusters. The above statement is justified by a number of sharp peaks deviating from the flatter region as shown in the expression profile obtained from Cheng Church. As supported by the expression profile, OPSM and xMotif have relatively higher enrichment value, as the number of peaks deviating from the common theme is fewer, as compared to the previous case. Moreover the size of the bicluster obtained is also small which makes it more stable than the biclusters obtained from Cheng Church algorithm. The Bimax algorithm achieves similar scores as the best performing biclustering techniques in this study. It has the maximum enrichment value and the minimum completion times except Cheng Church algorithm. As the biclusters produced by Cheng Church are insufficient to produces interesting patterns, it can't be effectively used to extract biclusters in every area of application. Nevertheless, the reference method can be used as a preprocessing step leading to many potentially relevant biclusters. Later, the chosen biclusters can be used, for example, as an input for more accurate biclustering methods in order to speed up the processing time and to increase the bicluster quality. An advantage of Bimax is that it is capable of generating all optimal biclusters, given the underlying binary data model. It is an extremely powerful approach and can be effectively used in a number of applications for useful purposes.

References

[1] Hartigan J.A., "Direct clustering of a data matrix", *Journal of the American Statistical Association* (American Statistical Association) 67 (337): 123–9. doi:10.2307/2284710, 1972.

[2] Madeira S.C., Oliveira A.L., "Biclustering Algorithms for Biological Data Analysis: A Survey", *IEEE Transactions on Computational Biology and Bioinformatics* 1 (1): 24-45, doi:10.1109/TCBB.2004.2, 2004.

[3] Cheng Y., Church G.M., "Biclustering of expression data". *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*: 93–103, 2000.

[4] Barkow S., Bleuler S., Prelic A., Zimmerman P., Zitzler E., "BicAT : a biclustering analysis toolbox", Bioinformatics, pp.1282-1283, 2006.

[5] K. O. Cheng, N.F. Law, W.C. Siu, and T. H. Lau, "BiVisu: software tool for bicluster detection and visualization", *Bioinformatics*, vol. 23, pp. 2342-2344, 2007.

[6] A. Tanay. R. Sharan, and R. Shamir, "Biclustering Algorithms: A Survey", In *Handbook of Computational Molecular Biology*, Edited by Srinivas Aluru, Chapman, May 2004.

[7] S. Bergmann, J. Ihmels, and N. Berkai., "Iterative signature algorithm for the analysis of large-scale gene expression data,"

in *Physical Review*, volume 67, pages 1–18. American Physical Society, 2003.

[8] Amir Ben-Dor, Benny Chor, Richard Karp, and Zohar Yakhini, “discovering local expression structure in gene expression data: The order preserving sub-matrices problem,” in *proc. Of 6th international conference on computational biology (RECOMB’02), pp.49-57, 2002.*

[9] Prelic A, Bleuler S, Zimmermann P, Wille A, Buhlmann P, et al. “A systematic comparison and evaluation of biclustering methods for gene expression data.” *Bioinformatics* 22: 1122–1129, 2006

[10] T.M. Murali and Simon Kasif, “Extracting conserved gene expression motifs from gene expression data” *pacific symposium on biocomptuing*, Kauai, Hawaii, 3-7 january 2003.

[11] K.O. Cheng, N.F. Law, and W.C. Siu and A. W.C. Liew, "Biclusters Visualization and Detection Using Parallel Coordinate Plots", *AIP Conf. Proc.*, Volume 952, pp. 114-123, November 2, 2007.

[12] A. Tanay, R. Sharan, and R. Shamir, “Discovering statistically significant biclusters in gene expression data”, *Bioinformatics*, 18 (Suppl. 1), pp. S136–S144, 2002.

[13] Tim Beißbarth and Terence P. Speed, “GOstat: find statistically overrepresented Gene Ontologies within a group of genes”, *Advance Access Publication,* February 12, 2004.

[14] Dataset available on “http://www.cbil.upenn.edu/RAD3/php/download.php”.

[15] Xiaowen Liu and Lusheng Wang, “Computing the maximum similarity bi-clusters of gene expression data”, Bioinformatics,vol. 23, pp.50-56, 2007.