Appendix A: Biclustering Algorithms, Validation Indices, and Results

LIST OF SYMBOLS

κ_{jk}	Membership indicator variable of j^{th} column of k^{th} bicluster
$\mid I \mid$	Number of genes
$\mid J \mid$	Number of conditions
ω	Fudge factor
ρ_{ik}	Membership indicator variable of i^{th} row of k^{th} bicluster
$\mathbf{A}$	Data matrix
$\mathbf{B}_i$	i^{th} Bicluster
$\mathbf{c}$	Condition vector
$\mathbf{g}$	Gene vector
$\mathbf{S}$	Diagonal matrix
$\mathbf{U}$	Unitary Matrix
$\mathbf{u}$	Prototype column vector
$\mathbf{V}$	Unitary Matrix
$\mathbf{v}$	Vector of factors
θ_{ijk}	Background layer of element at i^{th} row of j^{th} column of k^{th} bicluster
$\varphi(\mathbf{D})$	Significant score of $\mathbf{D}$ sub matrix
ξ	Additive noise
a_{IJ}	Mean of all elements
a_{Ij}	Mean of the j^{th} gene
a_{iJ}	Mean of the i^{th} gene
a_{ij}	Expression level of instance i under attribute j
B_{num}	Number of biclusters
C	Set of vertices indicating columns of matrix
E	Set of edges of graph
$F(.)$	Indicator function
G	Bipartite graph
HD	Hausdroff distance
I	Subset of rows/genes
I_r	Row sets of the r^{th} sub matrix
J	Subset of columns/conditions
J_r	Row sets of the r^{th} sub matrix
jac	Jaccard index function
l_r	Level of r^{th} sub matrix
N	Number of columns
O	Optimal set of vertices

p Number of rows in sub matrix
P_c Corruption Probability
Q Measure for B-type co-expression
q Number of columns in sub matrix
R Set of vertices indicating rows of matrix
T Measure for T-type co-expression
t_C Threshold coefficients for conditions
t_G Threshold coefficients for genes
$W(p, q)$ Upper bound on model of size $p \times q$

1 BIDEAL: READY FOR USE BICLUSTERING ALGORITHMS

Herein, a brief overview of biclustering algorithms embedded in BIDEAL Toolbox is provided. The objectives of biclustering algorithms may vary from one to another. On the basis of different approaches, it can be categorized as [1]:

(i) Iterative row and column clustering combination
(ii) Divide and conquer
(iii) Greedy iterative search
(iv) Exhaustive bicluster enumeration
(v) Distribution parameter identification

The brief overview of biclustering algorithms and validation indices embedded in BIDEAL toolbox are discussed in further sections.

1.1 Cheng and Church (CC):

In [2], Cheng and Church proposed a biclustering algorithm to process expression data. Mean Squared Residue (MSR) score is used to extract δ-biclusters from the input data matrix. Coherency of the matrix can be measured by this MSR score as

$$\text{MSR} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} \left(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}\right)^2 \tag{1}$$

where I and J are the subsets of rows and columns respectively in given data matrix, $|I|$ and $|J|$ are the total number of rows and the total number of columns. a_{ij} is the element of Bicluster, a_{Ij} is the mean of j^{th} column, a_{iJ} is the mean of i^{th} rows and a_{IJ} is the mean of all the elements in a Bicluster.

In ideal case, the values of MSR is 0 which indicates Bicluster is having constant value. Higher value of MSR indicates lesser coherency. In this algorithm two main steps are performed: i) Node deletion and ii) Node addition. In node deletion algorithm initially full data matrix is considered then repeatedly rows or columns form the matrix is removed unless average residue score of submatrix reaches to MSR $< \delta$. Removal of row or column is done on the basis of average residue score of rows and columns as given by

$$\text{MSR}_{avgR} = \frac{1}{|J|} \sum_{j \in J} \left(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}\right)^2 \tag{2}$$

$$\text{MSR}_{avgC} = \frac{1}{|I|} \sum_{i \in I} \left(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}\right)^2 . \tag{3}$$

After this, node addition is performed on obtained submatrix to get maximal size Bicluster without increasing the score. Addition of row or column is done only if their MSR $< \delta$.

1.2 Bipartite Spectral Graph Partitioning (BSGP): In [3], Dhillon used BSGP to model data matrix as $G = (R, C, E)$. This algorithm is based on an exhaustive bicluster enumeration approach, which tries to find partitions of the minimum cut vertex in a bipartite graph between rows and columns. This can be represented as

$$cut(R_1 \cup C_1, ...R_k \cup C_k) = \min_{O_1,...O_k} cut(O_1, O_2...O_k) \tag{4}$$

where R and C are two sets of vertices representing rows and columns respectively. E represents set of edges in graph.

For given disjoint condition clusters, each column cluster has a corresponding row cluster. A measure of similarity of a gene with a columns cluster is the sum of the edge-weights to all the columns in the cluster. By this procedure, each of the row clusters can be decided by the column clusters. The same method is followed for finding the columns in clusters on the basis of the row clusters. Following this iterative procedure, column clusters decide the gene clusters and then the rows clusters decide the columns clusters. The aim is to identify a best row and column cluster to obtain the minimum weight crossing edges between partitions. Computationally, in terms of time and memory space, it is quite expensive.

1.3 Order Preserving Sub-matrices (OPSM): In [4], algorithm finds order preserving sub-matrices, which have expression level in strictly increasing linear order. The algorithm uses a heuristic approach for biclustering problem. A sub matrix can be said to be order preserving if under the permutation of the conditions the value of the gene expression data is linearly increasing or decreasing. Since the probability of finding exact order in the bicluster is bleak, an additional corruption probability P_c is introduced which allows an element in the bicluster not to follow the defined order by probability P_c. Let in data matrix $\mathbf{A}$, there is an order preserving submatrix of size $p \times q$. A bound to find the significant model is expressed as

$$W(p, q) = |J|...(|J| - p + 1) \sum_{t=q}^{|I|} \binom{|I|}{t} \left(\frac{1}{p!}\right)^t \left(1 - \frac{1}{p!}\right)^{\binom{|I|-t}{}}. \tag{5}$$

Although OPSM gives good biclusters, it is less efficient in identifying small and significant biclusters as formed clusters have support of large rows. Its performance largely depends upon the parameters used in the algorithm and to the choice of the partial models which is applied initially. OPSM is quite slow because at a time only one bicluster can be obtained. OPSM is very much sensitive with respect to its susceptibility to noise.

1.4 Iterative Search Algorithm (ISA): Coherently overlapping biclusters, also referred as Transcription Modules (TM), can be extracted by iterative search from the gene expression data matrix as proposed in [5]. At first, a seed bicluster is selected that constitutes randomly selected rows from the data matrix. The seed is finely tuned by adding rows and columns to it iteratively until a stable set TM is achieved. Lets represent the m_{th} TM by a pair of gene vector $\mathbf{g}_m = (g_m^1, g_m^2...g_G^N)$ and condition vector $\mathbf{c}_m = (c_m^1, c_m^2...c_C^N)$. For each condition in TM, the average expression level of genes in the TM is above a certain threshold t_C similar to gene's threshold t_G. This can be expressed as

$$\exists(t_G, t_C) = \begin{cases} \mathbf{c}_m = f_{t_C}(\mathbf{c}_m^{proj}), \\ \mathbf{g}_m = f_{t_G}(\mathbf{g}_m^{proj}) \end{cases}. \tag{6}$$

The threshold function is defined as

$$f_t(x) = \begin{pmatrix} w(X_1).\Theta(\tilde{X}_1 - t) \\ \vdots \\ w(X_1).\Theta(\tilde{X}_{N_x} - t) \end{pmatrix} \tag{7}$$

Series $\{\mathbf{g}_{(0)}, \mathbf{g}_{(1)} \cdots\}$ converges to a fixed point gene vector $g^{(n^*)}$ that satisfies by

$$\frac{|g^* - g^n|}{|g^* + g^n|} < \epsilon \tag{8}$$

where ϵ determines the accuracy of the fixed point $g^{(k^*)}$. Similarly, $c^{(k^*)}$ is estimated to obtain the final TM. ISA is a non deterministic and greedy algorithm. It can easily find the stronger and larger modules with ease but it is quite difficult to extract the weaker and smaller modules from the data matrix.

1.5 Spectral Clustering (kSpectral): Spectral techniques for finding the biclusters are based on Eigen vectors of data matrix. Spectral biclustering was proposed by Kluger [6]. The datasets are normalized. Then, Singular value decomposition technique is applied on the microarray, where the constant part wise Eigen values give the checkerboard patterns in the sub matrix. Finally, k mean clustering is applied to obtain the checkerboard structures from the data matrix.The biclusters extracted from the data matrix by this method is up or down regulated. The normalization can be done in a coupled or independent manner. Independent scaling, bistochastization, or logarithmic interactions are the normalization techniques used by this method.

1.6 Information Theoretic Co-clustering (ITL): In [7], information-theoretic formulation for biclustering is presented. Main concept is to view microarray as the non-negative contingency table and co-clustering as a pair of maps which consists of random clustered variables for row and column to give row and column cluster respectively. The approach considers input matrix as a joint probability distribution of row and column variables. Note that the algorithm tries to cluster both the variables simultaneously. It can be viewed as optimization problem, which maximizes the mutual information between the clustered random variables subject to constraints on the number of row and column clusters. In this method, row and column clustering are done simultaneously at every step and arriving at the local minimum in a finite number of iterations. Row and column cluster prototypes are created through which the closeness of each row and column distribution is respectively compared in terms of relative entropy, thus intending to optimize the loss function and intending to preserve the mutual information. This algorithm can reduce the problem of high dimensionality and sparsity. Even though, the algorithm is designed for two dimensional data, this can be extended to multi dimensional data.

1.7 xMotif Algorithm: Murali *et al.* [8] proposed a representation for gene expression data called as conserved gene expression motifs or xMotifs. This algorithm tries to find large conserved gene expression motifs from the given discretized data matrix. To find biclusters, this algorithm uses a greedy approach which conserves row. A sub matrix is said to be a conserved motif if the expression level of a gene is found consistent in the respective sub matrix. Comparing different gene motifs for different conditions, we get to know of genes which are conserved in multiple conditions but are in different state in different conditions. To determine xMotif, we need to compute a set G which contains the conserved gene in it, the state of particular gene and a set C of condition which match the motif. xMotif is developed on the notion of Projective cluster. The

algorithm uses a NP-Hard approach to find the largest motifs in the sample by transforming the objective function of finding the maximum-edge bipartite clique in bipartite graph to maximum largest motif in the conserved motifs.

1.8 Plaid Algorithm: In this algorithm [9], data matrix can be considered as a superposition of layers where layer is a subset of genes and conditions of the data matrix. The data is tried to fit in a plaid model that can be expressed as

$$a_{ij} = \sum_{k=1}^{B_{num}} \theta_{ijk} + \rho_{ik} + \kappa_{jk} \tag{9}$$

where k is the assumed number of biclusters, θ_{ijk} is the sum of different responses as described in [9], ρ_{ik} is a instance bicluster membership indicator variable,it has range from 0 to 1, κ_{jk} is a sample bicluster membership indicator variable,the range of the indicator variable also lies between 0 and 1.

1.9 FLexible Overlapped Biclustering (FLOC): Missing values often introduce random disturbances and affect the quality of the acquired biclusters, so to remove this problem FLOC [10] was introduced. FLOC also speeds up the operation of identifying biclusters. The best bicluster obtained would act as a seed for the other biclusters and this process will be continued iteratively until the best bicluster is obtained. The overall quality of the bicluster obtained from FLOC algorithm is defined by the MSR of all specified entries as in (1). The overall quality of the bicluster obtained from FLOC algortihm is defined by the mean residue of all specified entries as

$$\text{MSR}_{IJ} = \frac{\sum_{i \in I, j \in J} r_{ij}^2}{v_{IJ}} \tag{10}$$

where r_{ij} is the residue of the entry, v_{IJ} the volume of the bicluster. The biclusters acquired by this method give better results for a larger matrix with smaller mean squared residue when compared with the CC algorithm. In FLOC, there is no issue of missing value but if missing values are quite high than it would be difficult to extract coherent biclusters, so a parameter α ranging between 0 and 1 is introduced.

1.10 Binary Inclusion Maximal (BiMax) algorithm: An algorithm based on fast divide and conquer approach was proposed in [11]. It tries to find all the bi maximal biclusters which contains element 1 only. The algorithm requires the discretization of the gene expression level matrix into binary matrix by deciding a threshold; elements above it become 1 while elements below it become 0 which allows it to catalog a large number of biclusters. It divides the binary matrix into three sub matrices. One of them might contain element 0 therefore it can be taken out from further scrutiny. On the remaining two sub matrices, the algorithm is applied iteratively till it does not find biclusters containing element 1 only. The performance of the BiMax can be reduced drastically in case of large number of recursive calls. Also, in the presence of noise in the data matrix it can produce a huge number of biclusters.

1.11 Large Average Submatrix (LAS): A statistically advanced biclustering algorithm which finds large average sub matrix within the data matrix was presented in [12]. The algorithm uses a Gaussian null model for expressed data. A significance score based on the Bonferroni corrected p-value is calculated for each bicluster sub matrix. The calculated significance score is motivated by the Normal Cummulative Distribution Function. So, the LAS Algorithm finds the

bicluster to give the largest significance score. LAS can be described as sum of k overlapping biclusters and an additive noise ξ given as

$$a_{ij} = \sum_{k=1}^{B_{num}} l_r F\big(i \in I_r,\ j \in J_r\big) + \xi. \tag{11}$$

Now, significance score is calculated using the Null model. The elements of the data matrix are subtracted from the mean of the significance score to form a residual matrix.

1.12 Factor Analysis for Bicluster Acquisition (FABIA): Hochreiter *et al.* [13] presented a multiplicative model biclustering algorithm that takes linear alliance of genes and conditions into account. In multiplicative model, the row and column vector need to be multiple of each other. FABIA models the data matrix as the addition of k biclusters and an additive noise. Here, linear dependency of subsets of rows and columns can be described by outer product $\mathbf{u} \times \mathbf{v}^T$. Here genes which are not present in bicluster are represented as 0. The overall model is given by

$$A = \sum_{t=1}^{B_{num}} \mathbf{u}_t \mathbf{v}_t^T + \xi. \tag{12}$$

FABIA uses a two factor analysis for generation of biclusters. Also, the relationship between genes and conditions are generally defined by fuzzy and thresholds are used for unique biclusters. Generally, FABIA forms few biclusters due to presence of small number of conditions and performs poorly for small dataset.

1.13 BitBit Algorithm: In this algorithm [14], the bit-patterns are extracted from the data matrix using two phase process. First phase includes a novel encoding process to divide the columns of the data matrix to a certain length determined by the minimum number of columns (mnc). In the second phase, biclustering of bit-patterns takes place using selective search. Each pair of row generates a pattern. An initial bicluster is generated if the number of elements in the pattern is more than mnc and the pattern has not been generated previously. Now, rows are added to the bicluster if the pattern generated by subsequent rows matches the pattern . Bicluster is considered as valid, if the number of rows in the final bicluster is greater than or equal to the minimum number of rows (mnr). In the final bicluster, required input parameters are Binary Input Matrix, mnr, mnc. After the encoding step, the comparison between rows takes place at bit level. As a result, the dimension of the data matrix and the operation required in the second phase would be reduced significantly. Experiments shows that BitBit can generate similar results when compared to BiMax while taking less computational time and less no of biclusters. The result also show BitBit maintained its shape to the size as well as density of input data.

1.14 BiSim Algorithm: Excessive computations can also be tackled by using the iterative approach instead of the divide and conquer approach as in BiMax by avoiding recursion and also additional traversals of the matrix [15]. First of all, the column starting index, column ending index and row number for each row wise bicluster are identified. Then, larger biclusters are formed by comparing the similarity between the column starting index and column ending index of the row wise biclusters. Row index based comparisons are also performed. So, they identified the biclusters with only one traversal of the matrix in $O(n \times m)$ complexity.

1.15 MSVD Algorithm: Wang *et al.* [16] proposed an algorithm which is called MSVD-MOEB (Modular Singular Value Decomposition Multi-Objective Evolutionary biclustering). Algorithm splits the gene expression data matrix into a set of sub-matrices with equal dimensions into a non-overlapping manner. SVD is performed on the partition sub-matrices. SVD factorizes any kind of matrices either rectangular or squared in dimension into unitary matrices and a diagonal matrix as

$$\mathbf{A} = \mathbf{USV}^T. \tag{13}$$

Then it projects the data obtained for desired number of eigen values and applies k-means clustering to cluster them. Thus, identifying the number of biclusters for defined number of clusters and repeats it for all the other sub matrices split from the data matrix.

1.16 QUalitative BIClustering (QUBIC): Deterministic algorithm which calculates biclusters in the discretized matrix by employing a qualitative or semi-qualitative means to gene expression data and optimization techniques was proposed in [17]. In this algorithm, expression level of genes is expressed in qualitative or semi-qualitative manner under multiple conditions as an integer value. Now, we consider a pair of genes and determine under how many conditions they share the same expression level. Similarly, for negatively related patterns we look for genes with opposite signs. It can find positive as well as negative correlated expression levels. The basic aim of the algorithm is to find sub matrices according to some optimization function or criteria.

1.17 Robust Biclustering Algorithm (ROBA): Tchagang *et al.* proposed ROBA [18], where basic linear algebra techniques were used. There are three main steps of this algorithm. First step involves pre-processing of data to handle missing values and noise. Second step decomposes the given data matrix into binary matrices. Last step involves identification of bicluster on the basis of type of bicluste i.e. bicluster with constant value, constant value on rows, constant value on columns, coherent value, and coherent evolution. For example, to extract constant value on rows biclusters distinguishable non-zero rows of all the elementary matrices are found. Similarly to find constant value on column biclusters search is performed for distinguishable non-zero columns. It also finds biclusters with coherent value. Various variants of Roba were later proposed like time and space efficient implementation in which authors have reduced the time and space complexity of the basic ROBA algorithm by a factor of λ, where λ represents the number of distinct values of the data matrix where they did not check the number of duplicate biclusters considering that they will not produce much substantial variation being the duplicate biclusters nominal in number

2 BIDEAL: Accessible Validation Indices for Performance Measures

Since there is no optimal algorithm for finding biclusters, various indices are used to check the quality of the biclusters. The validation indies present in BIDEAL toolbox are described in further subsections.

2.1 Jaccard Coefficient: Sebastian *et al.* [19] has proposed Jaccard index. This index compares the biclusters obtained by applying the two different biclustering algorithms and finding out the number of similar biclusters between them. Jaccard index is given as

$$jac_c\big(\mathbf{B}_1, \mathbf{B}_2\big) = \frac{jac\big(\mathbf{B}_1, \mathbf{B}_2\big)}{max\big(jac_c\big)}. \tag{14}$$

Jaccard index gives a value of zero if the biclusters given by the two different algorithms are totally dissimilar and gives a value of 1, if all the biclusters mined by the two algorithms are completely same.

2.2 SB Score: Differential co expression ranking score was proposed in [20]. Lets assume we have two biclusters, $\mathbf{B}_1$ and $\mathbf{B}_2$. Here, $\mathbf{B}_1$ is formed by gene under first set of conditions, while $\mathbf{B}_2$ is formed by same gene with second set of conditions. Chia proposed algorithm to compare the goodness of gene w.r.t to two different set of conditions. If $\mathbf{B}_1$ is good gene than there will be co-expression between gene and first set of conditions while differential co-expression between gene and second set of condition. Quantification of differential co-expression [20] can be measured using SB ($\mathbf{B}_1$) score of $\mathbf{B}_1$ bicluster as (15)

$$SB(\mathbf{B}_1) = \log\left(\frac{max(T_1(\mathbf{B}_1)+\omega), max(Q_1(\mathbf{B}_1)+\omega)}{max(T_2(\mathbf{B}_1)+\omega), max(Q_2(\mathbf{B}_1)+\omega)}\right) \tag{15}$$

where ω is used to offset the large ratios.

2.3 Constant Variance: In [21], corresponding variance of genes/conditions as the average of the sum of Euclidean distances between rows and columns of bicluster (biclusterwise and overall) is taken into consideration. Higher the value of variance lowers the quality of bicluster. The expression of the variance is given as

$$var = \sum_{i\epsilon I, j\epsilon J} (a_{ij} - a_{IJ})^2. \tag{16}$$

2.4 Sign Variance: To measure coherence of a bicluster Sign Variance is used. For more coherant bicluster, value of sign variance is lower. It is same as constant variance except it preprocesses the data matrix into sign matrix and then estimates variance.

2.5 Hausdroff Distance: This quality index is based on Hausdorff distance. Distance between the biclusters signifies the dissimilarity between them [22]. Hausdroff distance calculates the distance between the pair of sub matrices obtained from the input gene expression data matrix. So, Hausdorff distance is the maximum distance for traversal from the element of first bicluster to the nearest element of second bicluster. Lets take $\mathbf{B}_1$ and $\mathbf{B}_2$, from same metric space. Mathematically, Hausdorff can be modeled as

$$\begin{aligned} HD\big(\mathbf{B}_1, \mathbf{B}_2\big) = max\big\{ & sup_{b\epsilon\mathbf{B}_1} inf_{b\epsilon\mathbf{B}_2}, \\ & d(b_1, b_2), sup_{b\epsilon\mathbf{B}_1} inf_{b\epsilon\mathbf{B}_2}, d(b_1, b_2)\big\}. \end{aligned} \tag{17}$$

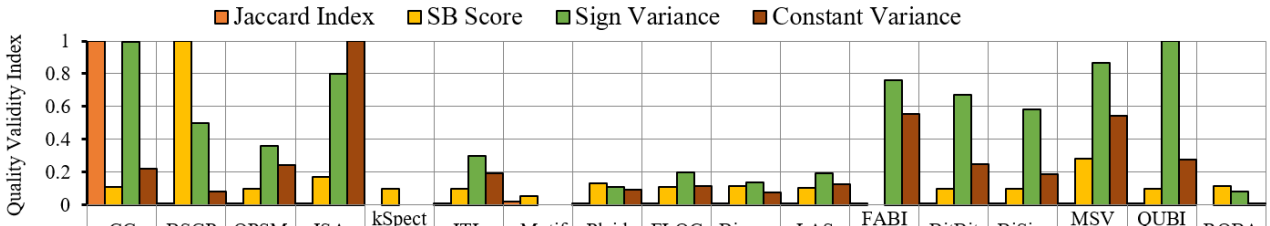
2.6 Mean Residue Score: To calculate mean residue score, mean square error (MSE) of the each and every bicluster of a particular algorithm can be calculated and then overall MSE can be calculated by taking the mean of the individual values.

3 BIDEAL: Testing and Validations on Benchmark Datasets

For the testing and validation of developed toolbox, four different datasets Saccharomyces cerevisiae cell cycle dataset (Yeast) [2], Leukemia adatset (ALL vs. AML) [23], Mammary tissue profile dataset (GDS205) [24], and Ligand screen in B cells dataset: Epstein Barr virus-induced molecule-1 [25] have been considered. Figs. 1-4 show the validity indices obtained using biclustering algorithms embedded in BIDEAL on four datasets. Table 1 shows the number of biclusters

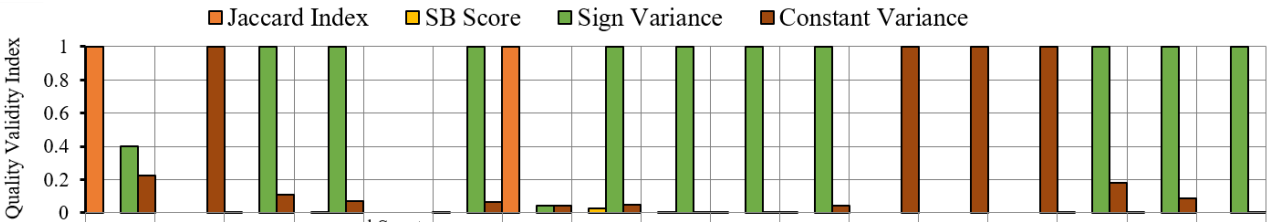
TABLE 1: Number of biclusters using 17 algorithms available in BIDEAL on four datasets.

Dataset	CC [2]	BSGP [3]	OPSM [4]	ISA [5]	kSpectral [6]	ITL [7]	xMotif [8]	Plaid [9]	Floc [10]	BiMax [11]	LAS [12]	Fabia [13]	BitBit [14]	BiSim [15]	MSVD [16]	Qubic [17]	ROBA [18]
Yeast [2]	100	10	16	16	-	1	97	4	20	75	20	2	212	1547	13	10	10104
ALLvsAML [23]	1	-	37	500	-	100	89	4	20	100	52	5	-	-	100	-	32591
GDS205 [24]	1	7	7	13	6	-	5	-	20	11	5	5	-	1	3	10	3925
GDS301 [25]	1	-	10	1	-	100	39	-	20	100	5	1	1	1	1	-	-



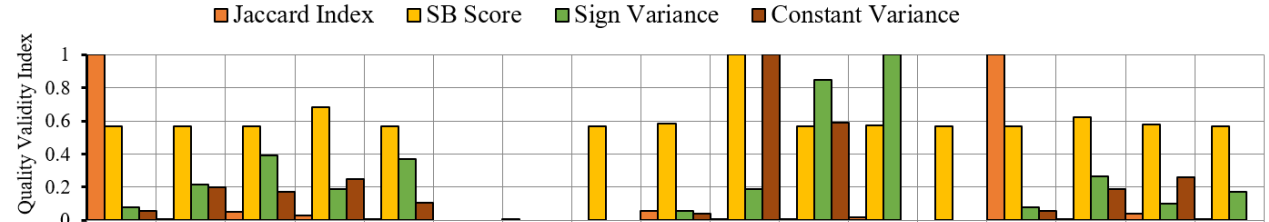
	CC	BSGP	OPSM	ISA	kSpectral	ITL	xMotif	Plaid	FLOC	Bimax	LAS	FABIA	BitBit	BiSim	MSVD	QUBIC	ROBA
Jaccard Index	1	0.0003	0.0077	0.0003	0	0.0002	0.0211	0.0003	0.0019	0.0004	0.0022	0.002	0.0002	0.0002	0.0049	0.0004	0
SB Score	0.111	1	0.0997	0.1701	0.0997	0.0997	0.0529	0.1328	0.1092	0.114	0.1018	0	0.0997	0.1	0.2815	0.0997	0.1129
Sign Variance	0.9947	0.4991	0.3602	0.7973	0	0.2966	0	0.1075	0.196	0.1361	0.1942	0.7619	0.6732	0.5804	0.869	1	0.0828
Constant Variance	0.2198	0.0804	0.2405	1	0	0.1905	0	0.0925	0.1174	0.0769	0.1263	0.5553	0.2488	0.1889	0.5411	0.2756	0.01

Fig. 1: Validation indices for quality of biclusters formed using various biclustering algorithms on Yeast dataset



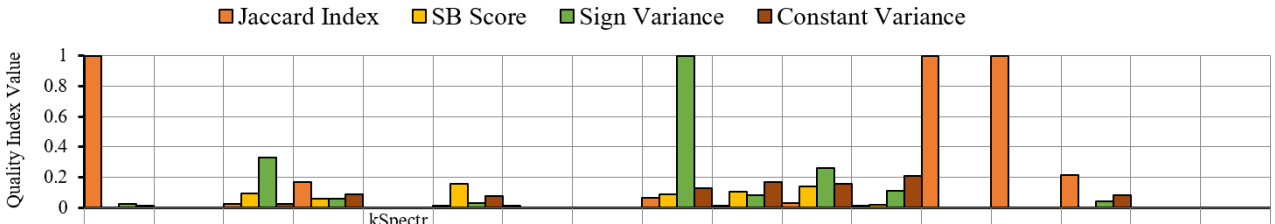
	CC	BSGP	OPSM	ISA	kSpectral	ITL	xMotif	Plaid	FLOC	Bimax	LAS	FABIA	BitBit	BiSim	MSVD	QUBIC	ROBA
Jaccard Index	1	0	0.00615	0.00E+0	0	2.77E-0	1	0	0	3.84E-0	0.0009	0	0	0	0.00358	0.00259	0
SB Score	0	0	0	0.00419	0	0	0	0.0264	0.0039	0	0	0	0	0	0	0	0
Sign Variance	0.39872	0	1	1	0	1	0.04088	1	1	1	1	0	0	0	1	1	1
Constant Variance	0.22193	1	0.10723	0.06991	0	0.06667	0.04088	0.0468	0.00515	0.00246	0.0422	1	1	1	0.18296	0.08743	0.00641

Fig. 2: Validation indices for quality of biclusters formed using various biclustering algorithms on ALL vs. AML dataset



	CC	BSGP	OPSM	ISA	kSpectral	ITL	xMotif	Plaid	FLOC	Bimax	LAS	FABIA	BitBit	BiSim	MSVD	QUBIC	ROBA
Jaccard Index	1	0.00156	0.04844	0.0275	0.00109		0.00109	0	0.05419	0.00187	0.00103	0.0174	0	1	0.001	0.0414	0.00026
SB Score	0.56804	0.56804	0.56804	0.68443	0.56804		0	0.56804	0.58429	1	0.56804	0.57398	0.56804	0.56804	0.62203	0.57696	0.56804
Sign Variance	0.07957	0.21709	0.39049	0.18842	0.3668		0	0	0.05414	0.18835	0.84639	1	0	0.07957	0.26646	0.09832	0.17081
Constant Variance	0.05528	0.19781	0.16943	0.24583	0.10481		0	0	0.03694	1	0.5866	0	0	0.05528	0.1874	0.25851	0

Fig. 3: Validation indices for quality of biclusters formed using various biclustering algorithms on GDS205 dataset



	CC	BSGP	OPSM	ISA	kSpectral	ITL	xMotif	Plaid	FLOC	Bimax	LAS	FABIA	BitBit	BiSim	MSVD	QUBIC	ROBA
Jaccard Index	1	0	0.025	0.1665	0	4.20E-0	0.00213	0	0.0645	5.21E-0	0.0314	0.0064	1	1	0.213	0	0
SB Score	0	0	0.094	0.0605	0	0.16	0	0	0.0888	0.107	0.14	0.0185	0	0	0	0	0
Sign Variance	0.027	0	0.33	0.06	0	0.03	0	0	1	0.0829	0.26	0.11	0	0	0.0442	0	0
Constant Variance	0.0139	0	0.023	0.09	0	0.078	0	0	0.13	0.17	0.16	0.21	0	0	0.0849	0	0

Fig. 4: Validation indices for quality of biclusters formed using various biclustering algorithms on GDS301 dataset

obtained using biclustering algorithms embedded in BIDEAL on four datasets. The first dataset consists of 2884 genes and 17 conditions and second dataset consists of 3571 genes and 72 conditions. Third dataset comprises 822 genes and 8 conditions. Fourth dataset comprises of 16271 genes and 11 conditions. To process the data first preprocessing has been performed, then one by one each of the algorithms was executed to obtain biclusters. Next step was to check the validity of generated biclusters. All validation indices were normalized between 0-1. Each algorithm is run either using default parameters or random initialization as needed.

REFERENCES

[1] S. C. Madeira, and A. L. Oliveira, "biclustering algorithms for biological data analysis: a survey," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 1, no. 1, pp. 24-45, 2004.

[2] Y. Cheng, and G.M. Church, "Biclustering of expression data," *Conf. on Intelligent Systems for Molecular Biology*, vol. 8, pp. 93-103, 2000.

[3] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," *Int. Conf. on Knowl. discovery and data mining*, pp. 269-274, 2001.

[4] A. Ben-Dor, B. Chor, R. Karp, and Z.Yakhini, "Discovering local structure in gene expression data: the order-preserving submatrix problem," in *Int. Conf. on Computational biology*, vol. 10, pp. 49-57, 2000.

[5] S. Bergmann, J. Ihmels, and N. Barkai, "Iterative signature algorithm for the analysis of large-scale gene expression data," *Physical Review E*, vol. 67. no. 3, pp. 031902, 2003.

[6] Y. Kluger, R. Basri, J.T. Chang, and M. Gerstein, "Spectral biclustering of microarray data: coclustering genes and conditions," *Genome research*, vol. 13, no. 4, pp. 703-716, 2003.

[7] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," *Int. Conf. on Knowl. discovery and data mining*, pp. 89-98, 2003.

[8] T. M. Murali, and S. Kasif, "Extracting conserved gene expression motifs from gene expression data," in *Proc. of Pacific Symposium Biocomputing*, vol. 3, pp. 77-88, 2003.

[9] L. Lazzeroni, and A. Owen, "Plaid models for gene expression data," *Statistica Sinica*, vol. 12, pp. 61-86, 2002.

[10] J. Yang, H. Wang, W. Wang, and P. S. Yu, "An improved biclustering method for analyzing gene expression profiles," in *Int. J. on Artificial Intelligence Tools*, vol. 14, no. 5, pp. 771-789, 2005.

[11] A. Preli, S. Bleuler, P. Zimmermann, A. Wille, P. Bhlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler, "A systematic comparison and evaluation of biclustering methods for gene expression data," *Bioinformatics* vol. 22, no. 9, pp. 1122-1129, 2006.

[12] A.A. Shabalin, V.J. Weigman, C.M. Perou, and A.B. Nobel, "Finding large average submatrices in high dimensional data," *The Annals of Applied Statistics*, pp. 985-1012, 2009.

[13] S. Hochreiter, U. Bodenhofer, M. Heusel, A. Mayr et. al. "FABIA:Factor analysis for bicluster Information Acquisition," *Bioninformatics*, vol 26, no. 12, pp. 1520-1527, 2010.

[14] D.S. Rodriguez-Baena, A.J. Perez-Pulido, and J.S. Aguilar-Ruiz, "A bi-clustering algorithm for extracting bit-patterns from binary datasets," *Bioinformatics*, vol. 27, no. 19, pp. 2738-45, 2011.

[15] N. Noureen, and M.A. Qadir, "BiSim: A Simple and Efficient Biclustering Algorithm," *Int. Conf. on Soft Computing and Pattern Recognition*, pp. 1-6, 2009.

[16] D. Wang, and Zheng, "MSVD-MOEB algorithm applied to cancer gene expression data," *Int. Conf. on Awareness Science and Technology* (iCAST), pp. 119-124, 2015.

[17] L. Guojun, Q. Ma, H. Tang, A. H. Paterson, and Y. Xu., "QUBIC: a qualitative biclustering algorithm for analyses of gene expression data," *Nucleic acids research*, pp. gkp491, 2009.

[18] A.B. Tchagang, and A.H. Tewfik, "Robust Biclustering algorithm (ROBA) for DNA microarray data analysis," *Proc. IEEE/SP 13th Workshop on Statistical Signal Processing*, pp. 984-989, 2005.
[19] Filippone, M., Masulli, F., and S. Rovetta, "Stability and performances in biclustering algorithms," *Int. Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pp. 91-101, 2008.
[20] B. K. H. Chia and R. K. M. Karuturi "Differential co-expression framework to quantify goodness of biclusters and compare biclustering algorithms," *Algorithms for Molecular Biology*, vol. 5, no. 1, pp. 23, 2010.
[21] N.K. Verma, and A. Roy, "Self Optimal Clustering Technique Using Optimized Threshold Function," *IEEE Syst. J.* , vol. 99, pp. 1-14, 2013.
[22] N. K. Verma, E. Dutta, and Y. Cui, "Hausdorff Distance and Global Silhouette Index as Novel Measures for Estimating Quality of Biclusters,"*Int. Conf. on Bioinformatics and Biomedicine*, pp. 267-272, 2015.
[23] T.R. Golub " Molecular classification of cancer: class discovery and class prediction by gene expression monitoring". Science, vol. 286, no. 5439, pp. 531-537,1999.
[24] S.P. Suchyta, S. Sipkovsky *et al.* "Bovine mammary gene expression profiling using a cDNA microarray enhanced for mammary-specific transcripts", Physiol Genomics, vol. 16, no. 1, pp. 8-18, 2003. Available Online: https://www.ncbi.nlm.nih.gov/sites/GDSbrowser?acc=GDS205
[25] Available Online: https://www.ncbi.nlm.nih.gov/sites/GDSbrowser?acc=GDS301