Ind Jour Math & Comp Sc. Jhs, Vol. I-II, 2011 Pg. 1-7

Handling high-dimensional data using Bi-Clustering

U. S Pandey, Beena Bakshi and Sanjay Batra
School of Open Learning. Delhi University, Email : us_pandey@hotmail.com
email : suris@vsnl.com

Received on 25/1/2012

ABSTRACT - We are interested in clustering genes as well as that small subset of conditions under which they are co-expressed. Such clusters are called bi-clusters. Each set of genes is expected to be affected by one or more transcription factors. As a gene may be responsible for several cellular activities it may be included in more than one bi-clusters. Having clustered genes of similar nature, next question is what makes them behave similarly. Thus one is interested in finding patterns in their genetic composition. These patterns are called motifs. Several algorithms are known to discover motifs in sequences. Earlier, people believed that only one TF was responsible for the similar expression pattern of several genes. Recently, people have started exploring the possibility of several motifs being responsible for the similarity, and some positive results have been obtained to this effect. Thus an interesting question evolves: If we know the set of motifs occurring on the promoter regions of a set of genes and their locations too, can we find out certain patterns (some combinations of these motifs) which may be causing the genes to behave similarly.

Positive results of these analysis help us to find the medicinal solutions for diseases like cancer , autism etc. As we know that mutation in genetic data results in these kind of incurable diseases. So finding out the significant patterns from the high dimensional data will help in drug discovery.

Clustering high-dimensional data

Clustering high-dimensional data is the cluster analysis of data with anywhere from a few dozen to many thousands of dimensions. Such high-dimensional data spaces are often encountered in areas such as medicine, where DNA microarray technology can produce a large number of measurements at once, and the clustering of text documents, where, if a word-frequency vector is used, the number of dimensions equals the size of the dictionary.

Problems

There are four main problems which one can face in clustering in high-dimensional data according to Kriegel, Kröger & Zimek (2009).

- Curse of dimensionality : Multiple dimensions are hard to think in, impossible to visualize, and, due to the exponential growth of the number of possible values with each dimension, impossible to enumerate.
- The concept of distance becomes less precise as the number of dimensions grows, since the distance between any two points in a given dataset converges. The discrimination of the nearest and farthest point in particular becomes meaningless:

$$\lim_{d\to\infty} \frac{dist_{\max} - dist_{\min}}{dist_{\min}} \to 0$$

- A cluster is intended to group objects that are related, based on observations of their attribute's values. Some of the attributes will usually not be meaningful for a given cluster. E.g., in newborn screening a cluster of samples might identify newborns that share similar blood values, which might lead to insights about the relevance of certain blood values for a disease. But for different diseases, different blood values might form a cluster, and other values might be uncorrelated. This is known as the *local feature relevance* problem: different clusters might be found in different subspaces, so a global filtering of attributes is not sufficient.
- Given a large number of attributes, it is likely that some attributes are correlated.

Bi-clustering

Bi-clustering, co-clustering, or subspace clustering is a data mining technique which allows simultaneous clustering of the rows and columns of a matrix. The term was first introduced by Mirkin (recently by Cheng and Church in gene expression analysis), although the technique was originally introduced much earlier (i.e., by J.A. Hartigan).

Clustering methods can be applied to either the rows or the columns of the data matrix, separately. Bi-clustering methods, on the other hand, perform clustering in the two dimensions simultaneously. This means that clustering methods derive a *global model* while bi-clustering algorithms produce a *local model*. When clustering algorithms are used, each gene in a given gene cluster is defined using all the conditions. However, each gene in a bi-cluster is selected using only a subset of the conditions and each condition in a bi-cluster is selected using only a subset of the genes.

The goal of bi-clustering techniques is thus to identify subgroups of genes and subgroups of conditions, by performing simultaneous clustering of both rows and columns of the gene expression matrix, instead of clustering these two dimensions separately. We can then conclude that, unlike clustering algorithms, bi-clustering algorithms identify groups of genes that show similar activity patterns under a specific subset of the experimental conditions.

Therefore, bi-clustering approaches are the key technique to use when one or more of the following situations applies:

1) Only a small set of the genes participates in a cellular process of interest.

2) An interesting cellular process is active only in a subset of the conditions.

3) A single gene may participate in multiple pathways that may or not be co-active under all conditions.

For these reasons, bi-clustering algorithms should identify groups of genes and conditions, obeying the following restrictions:

1. A cluster of genes should be defined with respect to only a subset of the conditions.
2. A cluster of conditions should be defined with respect to only a subset of the genes.
3. The clusters should not be exclusive and/or exhaustive: a gene or condition should be able to belong to more than one cluster or to no cluster at all and be grouped using a subset of conditions or genes, respectively.

II. DEFINITIONS AND PROBLEM FORMULATION

We will be working with an n by m matrix, where element $a_{IJ,}$ will be, in general, a given real value. In the case of gene expression matrices, aij represents the expression level of gene i under condition j .

A large fraction of applications of bi-clustering algorithms deal with gene expression matrices. However, there are many other applications for bi-clustering. For this reason, we will consider the general case of a data matrix, A, with set of rows X and set of columns Y , where the elements aij corresponds to a value representing the relation between row i and column j . Such a matrix A, with n rows and m columns, is defined by its set of rows, X = { x_1……x_n}, and its set of columns, Y ={ y_1……y_n}. We will use (X, Y) to denote the matrix A. considering that I € X and J € Y are subsets of the rows and columns, respectively, AI J= (I ; J) denotes the sub-matrix A that contains only the elements aij belonging to the sub-matrix with set of rows I and set of columns J . Given the data matrix A, *cluster of rows* is a subset of rows that exhibit similar behavior across the set of all columns. This means that a row cluster A_{IY}= (I ; Y) is a subset of rows defined over the set of all columns Y , where I = { i1……. ik} is a subset of rows (I € X and k <= n). A cluster of rows (I ,Y) can thus be defined as a k by m sub-matrix of the data matrix A.

Similarly, a *cluster of columns* is a subset of columns that exhibit similar behavior across the set of all rows. A cluster A_{XJ}= (X , J) is a subset of columns defined over the set of all rows X , where J = {j1……. js is a subset of columns (J € Y and s <= m). A cluster of columns (X , J) can then be defined as an n by s sub-matrix of the data matrix A.

A *bi cluster* is a subset of rows that exhibit similar behavior across a subset of columns, and vice-versa. The bi-cluster AI J= (I , J) is a subset of rows and a subset of columns where I ={ i……. ik} is a subset of rows (I € X and k <= n), and J = { j1….. js} is a subset of columns (J € Y and s <= m). A bi-cluster (I , J) can then be defined as a k by s sub-matrix of the data matrix A .

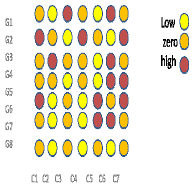
The specific problem addressed by bi-clustering algorithms can now be defined. Given a data matrix, A, we want to identify a set of bi-clusters Bk= (Ik , Jk) such that each bi-cluster B_k satisfies some specific characteristics of homogeneity.

Problem Complexity

Due to high dimension and high noise level complexity of the bi-clustering problem may depend on the exact problem formulation, and, specifically, on the merit function used to evaluate the quality of a given bi-cluster, almost all interesting variants of this problem are NP-complete. In its simplest form the data matrix A is a binary matrix, where every element a is either 0 or 1. When this is the case, a bi-cluster corresponds to a bi-clique in the corresponding bipartite graph. Finding a maximum size bi-cluster is therefore equivalent to finding the maximum edge bi-clique in a bipartite graph.

Microarray analysis allows the monitoring of the activities of many genes over many different conditions.

CONDITION



GENES

1.55	1.05	0.5	2.5	1.75	0.25	0.1
1.7	0.3	2.4	2.9	1.5	0.5	1.0
1.55	1.05	0.5	2.5	1.75	0.25	0.1
1.7	0.3	2.4		1.5	0.5	1.0
1.55		0.5	2.5	1.75	0.25	0.1
	0.3	2.4	2.9	1.5	0.5	1.0
1.55	1.05	0.5	2.5	1.75	0.25	0.1

To facilitate computational analysis the physical matrix which may contain 1000’s of gene’s is converted into a numerical matrix using image analysis equipment.

Type of Bi-cluster

1. Bi-cluster with constant values (a),
2. Bi-cluster with constant values on rows (b) or columns (c),
3. Bi-cluster with coherent values (d, e).

a) Bi-cluster with constant values				
2.0	2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0	2.0

b) Bi-cluster with constant values on rows				
1.0	1.0	1.0	1.0	1.0
2.0	2.0	2.0	2.0	2.0
3.0	3.0	3.0	3.0	3.0
4.0	4.0	4.0	4.0	4.0
4.0	4.0	4.0	4.0	4.0

c) Bi-cluster with constant values on columns				
1.0	2.0	3.0	4.0	5.0
1.0	2.0	3.0	4.0	5.0
1.0	2.0	3.0	4.0	5.0
1.0	2.0	3.0	4.0	5.0
1.0	2.0	3.0	4.0	5.0

d) Bi-cluster with coherent values (additive)				
1.0	4.0	5.0	0.0	1.5
4.0	7.0	8.0	3.0	4.5
3.0	6.0	7.0	2.0	3.5
5.0	8.0	9.0	4.0	5.5
2.0	5.0	6.0	1.0	2.5

e) Bi-cluster with coherent values (multiplicative)				
1.0	0.5	2.0	0.2	0.8
2.0	1.0	4.0	0.4	1.6
3.0	1.5	6.0	0.6	2.4
4.0	2.0	8.0	0.8	3.2
5.0	2.5	10.0	1.0	4.0

Algorithms

There are many bi-clustering algorithms developed for bioinformatics, including: Cheng and Church's Algorithm, CTWC (Coupled Two-Way Clustering) , ISA(The Iterative Signature Algorithm) ITWC (Interrelated Two-Way Clustering), δ-bi-cluster, δ-pCluster, δ-pattern, FLOC, OPC, Plaid Model, OPSMs (Order-preserving submatrixes), SAMBA (Statistical-Algorithmic Method for Bi-cluster Analysis),[6] , Robust Bi-clustering Algorithm (RoBA), Crossing Minimization [7] , cMonkey,[8] PRMs, DCC, LEB (Localize and Extract Bi-clusters), QUBIC (QUalitative Bi-clustering), BCCA (Bi-Correlation Clustering Algorithm) and FABIA (Factor Analysis for Bi-cluster Acquisition).[9] Bi-clustering algorithms have also been proposed and used in other application fields under the names co-clustering, biodimentional clustering, and subspace clustering.[10]

There is an ongoing debate about how to judge the results of these methods, as bi-clustering allows overlap between clusters and some algorithms allow the exclusion of hard to reconcile columns/conditions. Not all of the available algorithms are deterministic and you need to pay attention to the degree to which results represent stable minima.

IV. BI-CLUSTER STRUCTURE

Bi-clustering algorithms assume one of the following situations: either there is only *one bi-cluster* in the data matrix (see Fig. 4(a)), or the data matrix contains K *bi-clusters*, where K is the number of bi-clusters we expect to identify and is usually defined *apriori*. When the bi-clustering algorithm assumes the existence of several bi-clusters in the data matrix, the following bi-cluster structures can be obtained (see Fig. 4(b) to Fig. 4(i)):

1) Exclusive row and column bi-clusters (rectangular diagonal blocks after row and column reorder).

2) Non-Overlapping bi-clusters with checkerboard structure.

3) Exclusive-rows bi-clusters.

4) Exclusive-columns bi-clusters.

5) Non-Overlapping bi-clusters with tree structure.

6) Non-Overlapping non-exclusive bi-clusters.

7) Overlapping bi-clusters with hierarchical structure.

8) Arbitrarily positioned overlapping bi-clusters.

A natural starting point to achieve the goal of identifying several bi-clusters in a data matrix A is to form a color image of it with each element colored according to the value of a_{ij}.. It is natural then to consider ways of reordering the rows and columns in order to group together similar rows and similar columns, thus forming an image with blocks of similar colors. These blocks are subsets of rows and subsets of columns with similar expression values, hence, bi-clusters. An ideal reordering of the data matrix would produce an image with some number K of rectangular blocks on the diagonal (see Fig. 4(b)).

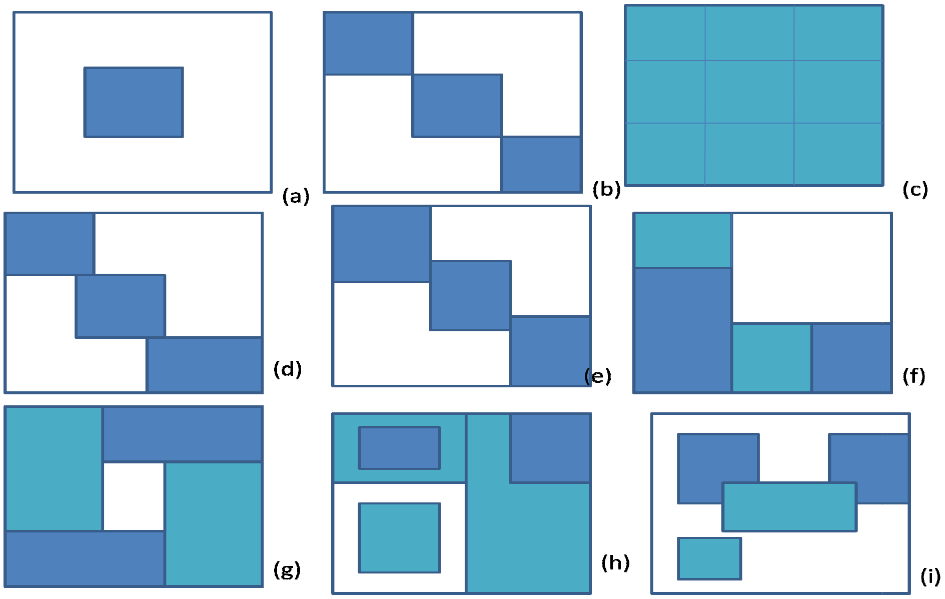


Fig.4. Bi-cluster Structure (a) Single Bi-cluster (b) Exclusive row and column bi-clusters (c) Checkerboard Structure (d) Exclusive-rows bi-clusters (e) Exclusive-columns bi-clusters(f) Non-Overlapping bi- clusters with tree structure (g) Non-Overlapping non-exclusive bi-clusters (h) Overlapping bi-clusters with hierarchical structure (i) Arbitrarily positioned overlapping bi-clusters

In this bi-clustering structure, every row in the row-block k is expressed within, and only within, those columns in condition-block k. That is, every row and every column in the data matrix belongs exclusively to one of the K bi-clusters considered (see Fig. 4(b)). The next natural step is to consider that rows and columns may belong to more than one bi-cluster, and assume a checkerboard structure in the data matrix (see Fig. 4(c)). By doing this we allow the existence of K non-overlapping and non-exclusive bi-clusters where each row in the data matrix belongs to exactly K bi-clusters. The same applies to columns. Kluger et al. assumed this structure on cancer data. The Double Conjugated Clustering (DCC) approach introduced by Busygin et al. also makes it possible to identify this bi-clustering structure. However, DCC tends to produce the structure in Fig. 4(b). Other bi-clustering approaches assume that rows can only belong to one bi-cluster, while columns, which correspond to conditions in the case of gene expression data, can belong to several bi-clusters. This structure, which is presented in Fig. 4(d), assumes exclusive-rows bi-clusters and was used by Sheng et al. and Tang et al. However, these approaches can also produce exclusive-columns bi-clusters when the algorithm is used using the opposite orientation of the data matrix. This means that the columns of the data matrix can only belong to one bi-cluster while the rows can belong to one or more bi-clusters (see Fig. 4(e)). The structures presented in Fig. 4(b) to Fig. 4(e) assume that the bi-clusters are exhaustive, that is, every row and every column in the data matrix belongs at least to one bi-cluster. However, we can consider non-exhaustive variations of these structures that make it possible that some rows and columns do not belong to any bi-cluster. Other exhaustive bi-cluster structures, include the tree structure considered by Hartigan and Tibshirani et al. and that is depicted in Fig. 4(f), and the structure in Fig. 4(g). A non-exhaustive variation of the structure presented in Fig. 4(g) was assumed by Wang et al. None of these structures allow overlapping, that is, none of these structures makes it possible that a particular column) belongs to more than one bi-cluster. The previous bi-cluster structures are restrictive in many ways. This structure, depicted in Fig. 4(h), requires that either the bi-clusters are disjoint or one includes the other. Two specializations of this structure, are the tree structure presented in Fig. 4(f), where the bi-clusters form a tree, and the checkerboard structure depicted in Fig. 4(c), where the bi-clusters, the row clusters and the column clusters are all trees. A more general bi-cluster structure permits the existence of K possibly overlapping bi-clusters without taking into account their direct observation in the data matrix with a common reordering of its rows and columns. Furthermore, these non-exclusive bi-clusters can also be non-exhaustive, which means that some rows or columns may not belong to any bi-cluster.

VII. BI-CLUSTERING APPLICATIONS

Bi-clustering can be applied whenever the data to analyze has the form of a real-valued matrix A, where the set of values a_{ij} represent the relation between its rows i and its columns j. An example is gene expression matrices. Large datasets of clinical samples are an ideal target for bi-clustering. However, and even though most recent applications of bi-clustering are in biological data analysis, there exist many other possible applications in very different application domains. Examples are information retrieval and text mining; and even analysis of electoral data.

A. Biological Applications

Cheng and Church applied bi-clustering to two gene expression data matrices, specifically to the Yeast Saccharomyces Cerevisiae cell cycle expression data with 2884 genes and 17 conditions and the human B-cells expression data with 4026 genes and 96 conditions. Yang et al. [29], [30] also used these two datasets. Wang et al. [28] and Liu and Wang [18] also used the Yeast data. Lazzeroni et al. [17] also used bi-clustering to identify bi-clusters in Yeast gene expression data: the rows of the data matrix represented 2467 genes and the columns were time points within each of 10 experimental conditions. Getz et al. [11] applied bi-clustering to two gene expression data matrices containing cancer data. The first data matrix was constituted by 72 samples collected from acute Leukemia patients at the time of diagnosis using RNA prepared from the bone marrow mononuclear cells of 6817 human genes: 47 cases were diagnosed as ALL (Acute Lymphoblastic Leukemia) and the other 25 as AML (Acute Myeloid Leukemia). They identified a possible diagnosis to leukemia by identifying different responses to treatment, and the groups of genes to be used as the appropriate probe. Busygin et al. [4] and Kluger et al. [16] also used these Leukemia data. The second gene expression matrix used by Getz et al. contained 40 colon tumor samples and 22 normal colon samples and 6500 human genes from which they choosed the 2000 of greatest minimal expression over the samples. Sheng et al. [23] also used leukemia expression data. The data matrix was this time constituted by 72 samples collected from acute Leukemia patients which were now classified into three types of Leukemia: 28 cases were diagnosed as ALL (Acute Lymphoblastic Leukemia), 24 as AML (Acute Myeloid Leukemia) and the remaining 20 as MLL (Mixed-Linkage Leukemia). The expression level of 12600 human genes was available. Tang et al. [25] applied ITWC to a gene expression matrix with 4132 genes and 48 samples of Multiple Sclerosis patients and Ben-Dor et al. [2] used a breast tumor dataset with gene expression data from 3226 genes under 22 experimental conditions. Kluger et al. [16] also used the Lymphoma expression data used by Tanay et al. but also applied bi-clustering to two extra gene expression matrices: a breast tumor dataset and a central nervous system embryonal tumor dataset.

B. Other Applications

Bi-clustering techniques can be used in identifying subgroups of customers with similar preferences or behaviors towards a subset of products with the goal of performing target marketing or use the information provided by the bi-clusters in recommendation systems. Recommendation systems and target marketing are important applications in the E-commerce area.

Bi-clustering can also be used to perform dimensionality reduction in databases with tables with thousands of records (rows) with hundreds of fields (columns). This application of bi-clustering is what the database community calls automatic subspace clustering of high dimensional data, which is extremely relevant in data mining applications. This problem is addressed by Agrawal et al. [1].

More exotic applications of bi-clustering use data matrices with electoral data and try to identify bi-clusters to discover subgroups of rows with the same political ideas and electoral behaviors among a subset of the attributes considered.

COMPARATIVE STUDY OF BICLUSTERING ALGORITHMS

Three different algorithms are chosen as all of them use different strategies. Bimax searches for constant up-regulated biclusters (using Madeira and Oliveira notation [1]), ISA searches for biclusters that highly deviate from the mean (both above or below) and OPSM searches for biclusters which preserve certain order (coherent evolution). Bimax uses a divide-and-conquer strategy while ISA uses Z-score statistics and OPSM performs a greedy iterative search. This way, we can present the results of the visualization under different biclustering conditions and discuss how those differences affect results by comparing their different layouts.

Bimax results analysis

Bimax is an exhaustive divide-and-conquer method that preprocesses the data matrix to convert it into a binary matrix by fixing a threshold, so transcription levels above this threshold become ones and transcription levels below become zeros (or vice versa). Then, it searches for *all* possible biclusters that contain only ones, so up or down-regulated constant biclusters are found.

OPSM results analysis

OPSM defines a bicluster as a group of rows whose values are monotonically increased under a certain column ordering, enabling us to find coherent evolution biclusters, i.e. genes and conditions that significatively increase or decrease at the same time regardless of the amount of the change. This is the broadest bicluster definition, yielding sometimes very large groups of genes.

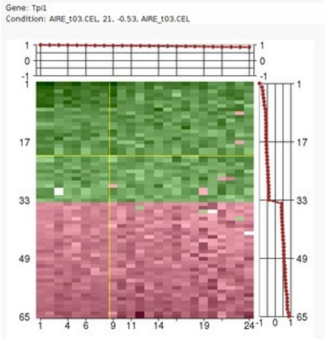
The detected conditions are biologically interesting because they are able to maintain an order in transcription levels over a large number of genes.

This feature could also be discovered by means of the visualization of single biclusters, but it requires much more effort. Also, third party relationships cannot be discovered unless all the elements in each bicluster are tracked one by one, while in this visualization they are quickly identified. For example, we can see that genes with locus tags

It is also remarkable that most of the genes grouped along with sporulation conditions at OPSM is not grouped by Bimax for the same conditions, suggesting that genes related to ribosomal subunits present order in transcription levels during sporulation, but they are not highly expressed.

ISA results analysis

Iterative Search Algorithm (ISA) aims at finding genes and conditions that deviate from the mean, so only highly up- or down-regulated genes and conditions are biclustered. The method starts with two normalized copies of the data matrix, one for genes and another one for conditions. Then, different thresholds are imposed for genes and conditions, and biclusters are searched using Z-score statistics. In the end, biclusters with both up- and down-regulated transcription levels are obtained.



Large sets of data, like expression profile from many samples, require analytic tools to reduce their complexity. The **Iterative Signature Algorithm (ISA)** was designed to reduce the complexity of very large sets of data by decomposing it into so-called "modules". In the context of gene expression data these modules consist of subsets of genes that exhibit a coherent expression profile only over a subset of microarray experiments. Genes and arrays may be attributed to multiple modules and the level of required coherence can be varied resulting in different "resolutions" of the modular mapping. Since the ISA does not rely on the computation of correlation matrices (like many other tools), it is extremely fast even for very large datasets.

Since ISA searches for both up and down-regulated biclusters, relevant nodes differ from Bimax.

BicOverlapper

We use the visual analysis approach described has been implemented as a Java framework called BicOverlapper. The overlapper technique was initially designed as a sketch in Processing, and later was translated to pure Java . Heatmap, TRN network and bubble map implementations make use of the Prefuse library.

. The framework makes use of three different sources of data:

- The bicluster results, which contain all the biclusters to be visualized in the overlapper.
- The microarray data matrix, necessary for the visualization of heatmaps and parallel coordinates.
- The TRN network with information about transcription regulations and necessary for the TRN visualization.

Although these data sources are fundamentally different, they all share genes and conditions as elementary entities, so the different visualizations in the framework can be linked by them.

Conclusion

Here we put light on concepts of bi-clustering and then compare results of few algorithms of bi-clustering. The proposed visualization done by Bicat, Bicoverlapper software allow us to display large number of biclusters in a single representation, enhancing the detection of overlap among biclusters.

Future work

We can combine all the results of different biclustering algorithms and then we can work upon them further to find out the significance of these clusters found.

Bibliography

[1] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulus, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM/SIGMOD International Conference on Management of Data*, pages 94– 105, 1998.

[2] Yizong Cheng and George M. Church. Bi-clustering of expression data. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB'00)*, pages 93–103, 2000.

[3] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 269–274, 2001.

[4] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. In *Proceedings of the Natural Academy of Sciences USA*, pages 12079–12084, 2000.

[5] Jinze Liu and Wei Wang. Op-cluster: Clustering by tendency in high dimensional space. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 187–194, 2003. [6] T. M. Murali and Simon Kasif. Extracting conserved gene expression motifs from gene expression data. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 8, pages 77–88, 2003.

[7] Amos Tanay, Roded Sharan, and Ron Shamir. Discovering statistically significant bi-clusters in gene expression data. In *Bioinformatics*, volume 18 (Suppl. 1), pages S136–S144, 2002.

[8] Jiong Yang, Wei Wang, Haixun Wang, and Philip Yu. Enhanced bi-clustering on expression data. In *Proceedings of the 3rd IEEE Conference on Bioinformatics and Bioengineering*, pages 321–327, 2003.

[9]Bergmann S, Ihmels J, and Barkai N. *Iterative signature algorithm for the analysis of large-scale gene expression data.* Phys Rev E Stat Nonlin Soft Matter Phys 2003 Mar; 67(3 Pt 1) 031902. pmid:12689096. PubMed HubMed PDF