Bi-clustering Gene Expression Data Using Co-similarity

Syed Fawad Hussain

Ghulam Ishaq Khan Institute of Engineering Sciences
and Technology, Pakistan
fawadsyed@gmail.com

Abstract. We propose a new framework for bi-clustering gene expression data that is based on the notion of co-similarity between genes and samples. Our work is based on a co-similarity based framework that iteratively learns similarity between rows using similarity between columns and vice-versa in a matrix. The underlying concept, which is usually referred to as bi-clustering in the domain of bioinformatics, aims to find groupings of the feature set that exhibit similar behavior across sample subsets. The algorithm has previously been shown to work well for document clustering in a sparse matrix representation. We propose a variation of the method suited for analyzing data that is represented as a dense matrix and is non-homogenous as is the case in gene expression. Our experiments show that, with the proposed variations, the method is well suited for finding bi-clusters with high degree of homogeneity and we provide empirical results on real world cancer datasets.

Keywords: Gene Expression Analysis, Bi-clustering, Co-similarity.

1 Introduction

The widespread use of microarray technologies during the last decade have enabled researchers to measure expression level of a large number (typically thousands) of genes under a number (typically in the hundreds) of different experimental samples (conditions). The resulting data is often represented as a matrix, called a gene expression data matrix, with rows usually representing an experimental condition, columns usually representing a gene and each cell of the matrix represents the intensity level of a given gene under a particular experimental condition. Each entry of the matrix corresponds to a numeric representation of the expression or activity of a particular gene under a given experimental condition, generally called sample. Applications of microarrays are quite wide, for instance the study of gene expression in yeast under different environmental stress conditions or the comparisons of gene expression profiles for tumors from cancer patients in order to find groups of genes that may exhibit similar patterns.

One usual goal in the analysis of such matrices is to extract the gene expression patterns inherent in the data and thus find potentially co-regulated genes. By comparing gene expression in normal and diseased cells, microarrays may be used to identify disease genes and targets for therapeutic drugs.

J. Tang et al. (Eds.): ADMA 2011, Part I, LNAI 7120, pp. 190–200, 2011.

Many unsupervised learning techniques such as self organizing maps (SOM) [1], k-means [2], and hierarchical clustering [3] have been used for gene expression analysis. Clustering algorithms have proved useful for grouping together genes with similar functions based on gene expression patterns under various conditions or across different tissue samples. All of the above work is focused on clustering genes using conditions as features. These techniques, however, may fail to discover patterns if a cluster of features are active in only a subset of samples.

In the last decade a new group of algorithms, known as bi-clustering algorithms, that tends to find patterns in gene expression across only a subset of conditions have been widely used [4–8]. An example of a bi-partition would be *{a; b; c},* and *{d; e}* for objects, and *{1; 4; 5}, {2; 3},* for features. This bi-clustering indicates that the commonality of objects from the partition *{a; b; c}, is* that they tend to share similar values among the feature group *{1; 4; 5}*. Similarly, features in *{2; 3}* can be used to characterize objects in *{d; e}*. Bi-clustering algorithms have been well studied in the context of gene expression data analysis because it provides valuable information about putative regulation mechanisms and biological functions since they are good in finding local patterns and have been shown to outperform traditional clustering algorithms in bioinformatics, for instance see [9]. Similarly, several bi-clustering algorithms have been proposed in other domains, such as in text mining [10], [11], in social networking[12], etc. Such algorithms have, however, not been tested for the task of bi-clustering genes and vice versa.

One such approach to bi-clustering has been recently proposed by [13], where the authors have used the concept of co-similarity to produce bi-clusters for solving a similar problem in text mining. Their algorithm, named χ-Sim, utilizes the concept of higher-order correlations between words and documents to generate two similarity matrices, each built on the basis of the other. The concept of 'higher-order' co-occurrences has been investigated in [14] among others, as a measure of semantic relationship between words. The motivation behind χ-Sim is the exploitation of the dual nature of problem i.e. the relationship between groups of words that occur in a group of documents. Thus, documents are considered similar and hence grouped together, if they contain similar words and words in turn are considered similar and therefore grouped together, if they occur in similar documents. The concept can be expressed in the form of iterative mathematical equations which can then be solved to arrive at a solution.

Our work in this paper is motivated by the work of [13]. We feel that the domain of text clustering and gene expression analysis have significant similarity and thus, algorithms developed for one domain can be exploited, albeit with modifications, into the other. The χ-Sim algorithm is well suited for data coming from a text corpus that is homogenous and usually discrete as occurs with textual data represented using the Vector Space Model (VSM) [15]. This, however, is not usually the case for gene expression data where different intensity levels might be present across experiments and between different genes. To this end, we employ several pre-processing techniques and modify the algorithm proposed by [13] necessitated by the nature of our data.

The rest of the paper is organized as follows. Section 2 introduces the basic concept of the χ-Sim algorithm as described by [13]. In section 2.2, we highlight the

potential shortcomings of using χ-Sim on gene expression data and proposes pre-processing and modification steps in the algorithm. Detailed empirical results substantiating the usefulness of co-clustering are provided in Section 3. In section 4, we give a brief survey of the related work. Finally we conclude with a summary of our work and provide directions in Section 5.

2 The χ-Sim Algorithm

Throughout this paper we use the classical notation: matrices (in capital letters) and vectors (in small letters) are in bold.

Let **D** be the data matrix representing a corpus having r rows and c columns; D_{ij} corresponds to the intensity of the j^{th} gene in the i^{th} sample. $\mathbf{d}_i$ represents the row vector corresponding to gene i and $\mathbf{d}^j$ represents the column vector corresponding to sample j. $\mathbf{D}^T$ and $\mathbf{d}_i^T$ denote the transpose of the matrix **D** and document vector $\mathbf{d}_i$ respectively. **SR** and **SC** represent the square and symmetric matrices of similarity between rows and similarity between columns of sizes r x r and c x c respectively with $SR_{ij} \in [0,1]$, $1 \le i,j \le r$ and $SC_{ij} \in [0,1]$, $1 \le i,j \le c$. **A*****B** represents the matrix multiplication between two matrices **A** and **B** while **A**⊗**B** denotes their Hadamard product.

2.1 The Algorithm

The χ-Sim algorithm is a co-similarity based approach which builds on the idea of generating simultaneously the similarity matrices between genes and between samples, each of them iteratively built on the basis of the other. We describe here the similarity between two genes (the similarity between samples being symmetrical). To calculate the similarity between two genes *i* and *j,* in addition to comparing the samples shared between the two genes, we also compare their 'similar' samples.

Thus all samples in document $\mathbf{d}_i$ are compared to all samples in document $\mathbf{d}_j$. The product is defined as a measure of similarity similar to the cosine measure. However, when comparing samples with different indices, say D_{ik} and D_{jl}, their product is weighted by the similarity value between the samples k and l given by SC_{kl}.

Mathematically speaking, the similarity measure is given by $SR_{i,j}=\mathbf{d}_i*\mathbf{SC}*\mathbf{d}_j^T$ where $\mathbf{d}_j^T$ denotes the transpose of the vector $\mathbf{d}_j$ and the symbol '*****' denotes a matrix multiplication. The algorithm starts with two matrices **SR** and **SC** initialized to the identity matrix **I**. In the absence of any prior knowledge about the similarity between any pair of genes, only the similarity value between a gene (or condition) with itself is considered as maximal and all other values are put to zero. **SR** and **SC** are then iteratively computed each one based on the other. Thus, genes are termed similar if they share similar conditions. Conditions, in turn are considered similar if they are up or down regulated in similar genes. This is termed as a co-similarity approach (as opposed to co-clustering which form hard clusters of the genes and conditions).

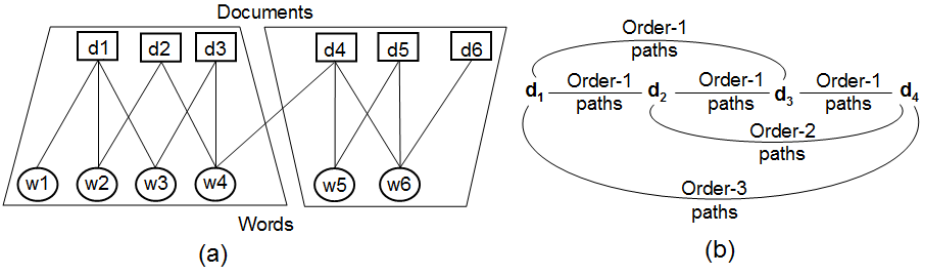

Fig. 1. (a) A bi-partite graph view of the matrix **D**. The square vertices represent genes and the rounded vertices represent samples, and (b) some of the higher order co-occurrences between genes in the bi-partite graph.

We now present a graph theoretical interpretation of the algorithm which would enable us to better understand the working of the algorithm. Consider the bi-partite graph representation of a data matrix in Fig. 1(a) having 6 genes d_1-d_6 and 6 conditions w_1-w_6. The genes and samples are represented by rectangular and oval nodes respectively and an edge between a gene i and a condition j in the graph corresponds to the entry D_{ij} in the matrix. There is only one order-1 path between genes d_1 and d_2 given by $d_1 \xrightarrow{D_{12}} w_2 \xrightarrow{D_{22}} d_2$. Hence the similarity value SR_{12} is given by the product $D_{12}D_{22}$. Note that since the **SC** matrix is initialized as identity, at the first iteration, SR_{12} corresponds to the dot product between $\mathbf{d}_1$ and $\mathbf{d}_2$ (since SC_{kl}=0 for all k≠l). The matrix $\mathbf{SR}^{(1)} = \mathbf{D}*\mathbf{D}^T$ thus represents the order-1 paths between all pair of genes $\mathbf{d}_i$ and $\mathbf{d}_j$, i=1..r and j=1..r. Each element of $\mathbf{SR}^{(1)}$ and $\mathbf{SC}^{(1)}$, denoted by $SR_{ij}^{(1)}$ and $SC_{ij}^{(1)}$ respectively is given by

$$SC_{ij}^{(1)} = \sum_{k=1}^{r} D_{ik} D_{kj} \quad \forall\ i,j \in [1,c] \tag{1}$$

$$SR_{ij}^{(1)} = \sum_{k=1}^{c} D_{ik} D_{kj} \quad \forall\ i,j \in [1,r] \tag{2}$$

Genes d_1 and d_4 do not have an order one path but are linked by d_2 and d_3. The similarity value contributed via the document d_2 can be explicitly represented as $d_1 \xrightarrow{D_{12}} w_2 \xrightarrow{D_{22}} d_2 \xrightarrow{D_{24}} w_4 \xrightarrow{D_{44}} d_4$. From Eq. (1), the sub-path $w_2 \rightarrow d_2 \rightarrow w_4$ can be represented as $SC_{24}^{(1)}$, which is the first order path between w_2 and w_4 given by $\mathbf{SC}^{(1)}$, and the contribution of d_2 in the similarity of SR_{14}^{1} via d_2 can be written as $D_{12}*SC_{24}^{1}*D_{44}$.

This is a partial similarity measure as d_2 is not the only gene that forms a link between d_1 and d_4. The similarity via d_3 (see fig 1(b)) is given by $D_{13}SC^{1}{}_{34}D_{44}$. The full similarity measure between d_1 and d_4 is thus given by $D_{12}SC^{1}{}_{24}D_{44} + D_{13}SC^{1}{}_{34}D_{44}$. This similarity can be represented as $\mathbf{SR}^{(2)}=\mathbf{D}*\mathbf{SC}^{(1)}*\mathbf{D}^T$ which corresponds to the **SR** matrix at the second iteration. Hence similarity matrix $\mathbf{SR}^{(2)}$ at the second iteration corresponds to paths of order-2 (the similarity matrix between samples is similarly

given by $\mathbf{SC}^{(2)}=\mathbf{D}^{T}*\mathbf{SR}^{(1)}*\mathbf{D}$). Computing similarities this way, however, can result in an unbalanced scale since different vectors can have different lengths. Therefore, we normalize the values SR_{ij} by $|\mathbf{d}_i| * |\mathbf{d}_j|$ where $|\mathbf{d}_i|=\Sigma_{k=1..c}\ (D_{ik})$ and $|\mathbf{d}_j|\ \Sigma_{k=1..c}(D_{jk})$. Hence the generalized algorithm to find similarity measure at iteration *n* is given by:

```
Algorithm χ-Sim
Input: Document by term matrix D, No. of iterations n;
Step 1: Initialize matrices SR(0) = I , SC(0)=I
Step 2: for t=1 to n
        SC(t)=DT.SR(t-1).D⊗NC with NCij = 1/(|di|.|dj|)   (3)
        SR(t)=D.SC(t-1).DT⊗NR with NRij = 1/(|di|.|dj|)   (4)
        Set diagonal of SR(t) and SC(t) to 1
```

Note that as a result of the L1 normalization used, the values of the matrix **SR** (or **SC**) are always bounded between 0 and 1 with 0 signifying no similarity (no connectedness between the two objects in order *n* paths in the graph) and 1 signifying maximum similarity.

2.2 χ-Sim as a Bi-clustering Algorithm

As mentioned previously, bi-clustering is the simultaneously clustering of rows and columns to identify row clusters that have some correlation with a certain column clusters. In this section, we describe how χ-Sim can be used as an algorithm to discover bi-clusters in non-homogeneous data such as gene expression data.

χ-Sim generates two similarity matrices – the row similarity matrix, **SR**, and the column similarity matrix, **SC**. The bi-clustering effect is achieved since each the similarity matrices, **SR** and **SC,** is built on the basis on the other, thus implicitly taking into account a feature selection. Applying the χ-Sim algorithm for bi-clustering gene expression data directly, however, poses a problem in performing comparisons. Genes, for instance, have different intensity profiles. For instance, if one gene's intensity ranges from 1-50 while another genes intensity ranges between 10000-20000, then comparing these genes can result in rather skewed and meaningless similarities, particularly since the basic operation of χ-Sim is the dot product. Thus, genes having higher intensities will generate greater similarity, which might lead to undue and undesired high similarity values.

Data transformation, sometimes referred to as normalization or standardization, is necessary to adjust individual hybridization intensities of genes profiles such that intensity values between different genes are balanced and a comparison between them becomes meaningful [6], [16]. Transformation of raw data is considered an essential element of data mining since the variance of a variable determines the importance of that feature. We consider two types of transformation as discussed below:

Column Standardization. This is a classical technique in data analysis usually referred to as *centering* or *scaling*. Column Standardization (CS) involves taking the

difference between intensity values of a gene from the mean in units of standard deviation. Mathematically speaking, column standardization is defined as

$$D_{ij} = \frac{D_{ij} - \bar{\mu}_j}{\sigma_j}, \ \forall i \in 1..m, j = 1..n \tag{5}$$

Where $\bar{\mu}_j = \frac{1}{m}\sum_{i=1}^{m} D_{ij}$ and $\sigma_j = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(D_{ij} - \bar{\mu}_j)^2}$.

Row Standardization. Similarly, Row Standardization (RS) is defined as

$$D_{ij} = \frac{D_{ij} - \bar{\mu}_i}{\sigma_i}, \ \forall i \in 1..m, j = 1..n \tag{6}$$

Where $\bar{\mu}_i = \frac{1}{n}\sum_{j=1}^{n} D_{ij}$ and $\sigma_i = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(D_{ij} - \bar{\mu}_i)}$.

Usually, one needs to perform either row standardization or column standardization in order to transform the data. In some cases, row standardization followed by column standardization or bi-normalization might be needed. However, we are in a case where we need to compare pair of rows in order to determine the similarity between columns, and compare pair of columns so as to compute the similarity between rows as given in equations (3) and (4) respectively. Therefore, using either of equations (5) or (6) will only solve our problem partially.

As a result, we propose using both types of transformations given in (5) and (6) while maintaining two sets of the original dataset, **D**. Let a row normalized matrix, transformed using equation (5), be denoted as $\mathbf{D}_R$ while a column normalized matrix, transformed using equation (6), be denoted as $\mathbf{D}_C$. Then, equations (3) and (4) can be re-written as

$$\mathbf{SC}^{(t)} = (\mathbf{D_R})^T.\mathbf{SR}^{(t-1)}.\ (\mathbf{D_R}) \otimes \mathbf{NC} \text{ with } NC_{ij} = 1/(|\mathbf{d}_i|.|\mathbf{d}_j|) \tag{7}$$

$$\mathbf{SR}^{(t)} = (\mathbf{D_C}).\mathbf{SC}^{(t-1)}.\ (\mathbf{D_C})^T \otimes \mathbf{NR} \text{ with } NR_{ij} = 1/(|\mathbf{d}_i|.|\mathbf{d}_j|) \tag{8}$$

Thus, by modifying the system of equations, we can now compare pair of rows and pair of columns using the appropriate normalized matrices. Of course, the overhead is that now we have to maintain separate copies of the dataset. The rest of the algorithm is unchanged as described previously in section 2.1. We will refer to this modified version of χ-Sim as χ-SIM_{mod}.

3 Experimentation

In order to validate the quality of bi-clusters generated by our algorithm, we used 2 real datasets that have been widely used in the literature and are publicly available.

3.1 Experimental Methodology

Colon Cancer. This dataset contains expression levels for 6500 human genes across 62 samples used by Alon et al [17]. The dataset corresponds to Colon Adenocarcinoma specimen collected from several patients, while normal tissues were also obtained from some of these patients. We selected the top 2000 genes with highest intensity across the samples. The resulting dataset contains 2000 genes and across 40 tumorous and 20 normal colon tissues. Note that this dataset do not contain negative values and only 1909 of the 2000 genes are unique. We further preprocessed the data by removing genes with $|max/min| < 15$ and $|max - min| < 500$ leaving a total of 1096 genes.

Leukemia. This dataset was used by Golub et al. [18] and contains 7129 genes across 72 samples. The dataset corresponds to RNA extracted from bone marrow samples of patients with leukemia at the time of diagnosis. About 47 samples were suffering from Acute Lymphoblastic Leukemia (ALL) while 25 samples were suffering from Acute Myeloid Leukemia (AML). We first used a floor value of 100 and a ceil value of 16000. Only genes with $|max/min| < 5$ and $|max - min| < 500$ were selected leaving a total of 3571 genes. The preprocessing steps applied here have been used to enable direct comparison with the results of Cho et al. [6].

The experimentation was run as follows: Each of the dataset was taken and the preprocessing was applied, resulting in a reduced matrix with only the selected genes across all the samples. The modified χ-Sim algorithm was run on the data matrix using Matlab and Agglomerative Hierarchical Clustering using Wards linkage was applied to the resulting similarity matrices. The number of sample clusters was set as the real number of clusters whereas the number of gene clusters was set to 100 (as in [6]). The top k bi-clusters were chosen as gene clusters with the greatest homogeneity across the sample clusters.

3.2 Sample Cluster Analysis

Our first analysis attempts to verify the quality of the sample clusters. We take the sample clustering from the generated bi-clusters and evaluate the accuracy of the samples. The accuracy measures the number of samples that were correctly grouped together in one bi-cluster as a percentage of the total samples in the dataset. For instance, the two sample categories for the colon cancer dataset are those samples that have tumor and those that do not have tumor. We report the results based on

1. applying no transformation (pre-processing) at all; and
2. Applying both row and column transformation (as discussed in the previous section; see equations (7) and (8))

The results are reported in table 1 below.

Table 1. Accuracy of sample clustering in the bi-clusters

	NT	RS + CS
Colon	0.8871	0.9032
leukemia	0.9583	0.9722

As can be seen from the table, without any transformation (i.e. baseline χ-Sim), only 88.71% of the samples are correctly clustered together. However, when applying the χ-SIM_{mod} algorithm, the accuracy of the sample clusters rises to 90.32%. A similar increase in the accuracy is observed in the case of leukemia dataset.

3.3 Gene Cluster Analysis

Unlike the condition clusters, we do not have a priori knowledge of the gene clusters. One way to analyze the gene clusters is by visually analyzing the profiles of genes that are clustered together in the bi-clusters. Ideally, we would like genes clustered together to exhibit similar profile behaviors under the condition clusters. Thus, plotting the bi-clusters give us a visual reference to the "profile" of the bi-cluster and the constituent genes. This approach has been employed previously to judge the quality of the generated bi-clusters, for instance by *[5], [6], [17]* among many others.

In order to report the best bi-clusters, we generated several bi-clusters and report the top k bi-clusters that exhibit similar profiles amongst the samples. The result for the top 2 clusters (k=2) for colon cancer dataset and leukemia dataset is shown in Figure 2. The x-axis corresponds to the tissues while the y-axis shows the intensity level of the gene. The figure clearly illustrates that χ-SIM_{mod} captured homogenous gene expression patterns in the gene clusters. Two bi-clusters representing a healthy tissue and tumor tissues for colon cancer dataset are shown in figure 2(a) and (b)

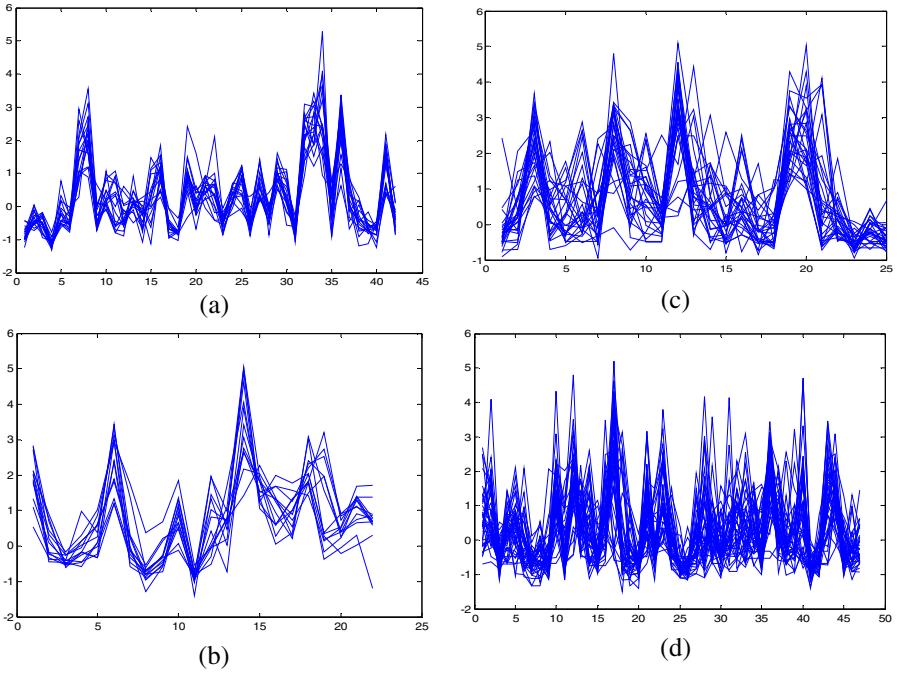

Fig. 2. Top 2 bi-clusters generated by applying the modified χ-Sim algorithm on (a) colon cancer dataset of tumor tissues, (b) colon cancer dataset of normal tissues; and (c) leukemia dataset with AML, (d) leukemia dataset with MLL.

respectively. As mentioned in [17] and [6], ribosomal genes usually have higher intensities in tumor samples and comparatively lower intensities in non-tumor (normal) samples. In fact, most of the ribosomal protein genes given in [17] can be found in two clusters.

Similarly, for the leukemia dataset, Figure 2(c) represent bi-cluster corresponding to genes profiles of AML while figure 2(d) corresponds to gene profiles of MLL. We can see that χ-SIM_{mod} is able to discover bi-clusters that have several genes behaving similarly over a large number of experimental conditions. For instance the bi-cluster shown in figure 2(a) contains only 16 genes showing similar behavior across all 42 conditions. Similarly, we also discover bi-clusters have many genes showing similar characteristics across different conditions, for instance in figure 2(c) where 27 genes show similar behavior across 25 conditions.

4 Related Work

Several algorithms for clustering of gene expression data have been proposed in the literature most algorithms were designed keeping bi-clustering of gene expression data in mind. Several kinds of bi-clusters have been identified and algorithms proposed that are able to identify them (see for instance [19] for common bi-clusters and a survey of algorithms proposed in the literature.). Perhaps the earliest and most well-known algorithm for finding bi-clusters in gene expression data was proposed by [5] that finds sub-matrices with the minimal squared residue. This algorithm however uses a greedy approach to find bi-clusters and does not take the overall similarity between genes and samples into account.

Tanay et al. [20] proposed an algorithm, known as the Statistical-Algorithmic Method for Bi-cluster Analysis (SAMBA) to discover bi-clusters. The model is based on a data matrix corresponding to a bipartite graph and uses statistical models to solve the problem by identifying bi-cliques in the graph. Depending upon whether a gene is up-regulated or down regulated, the corresponding edges are assigned a weight. The Order Preserving Sub Matrix (OPSM) technique proposed of Ben Dor et al. [21] assumes a probabilistic model of the data matrix. They define a bi-cluster as a group of rows such that the expression values in all features increase of decrease simultaneously.

The Bimax algorithm was used as a reference method in a comparative study of bi-clustering algorithms by [4]. The algorithm uses on 0's and 1's which can be obtained by discretization as a prior preprocessing step. A bi-cluster is then defined as a sub-matrix containing all 1's i.e. a set of genes that are up-regulated in a set of conditions.

More recently, a Minimum Sum Squared Residue Co-Clustering algorithm was proposed by Cho et al. [6] that do not take into account a correspondence between row and column clusters as such, but consider sub-matrices formed by them with the overall aim to minimize the sum of squared residue within the sub-matrix. Cho et al. proposed an algorithm that is based on algebraic properties of a matrix. The algorithm runs in an iterative fashion and on each iteration, a current co-clustering is updated

such that sum of squared residue is not increased. In other samples, the algorithm monotonically decreases and converges towards a locally optimal solution.

5 Conclusion

We provide an extension and adaptation of the χ-Sim algorithm for bi-clustering gene expression datasets. It has been successfully demonstrated that the proposed χ-Sim algorithm does perform bi-clustering and its results are better than other contemporary traditional techniques. The quality of the results was verified by bi-clustering several publicly available cancer data sets, and analyzed the results of both the gene and sample clusters.

Our work is also significant in that we have used an algorithm that was fundamentally developed for text clustering and adapted it to perform bi-clustering of gene expression data. This is an interesting scenario and one needs to further investigate if other co-clustering algorithms can be adapted for bi-clustering problem and vice versa and this will form a major part of our future work.

References

[1] Tamayo, P., et al.: Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proceedings of the National Academy of Sciences of the United States of America 96(6), 2907 (1999)

[2] Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., Church, G.M.: Systematic determination of genetic network architecture. Nature Genetics 22, 281–285 (1999)

[3] Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences 95(25), 14863 (1998)

[4] Barkow, S., Bleuler, S., Prelic, A., Zimmermann, P., Zitzler, E.: BicAT: a biclustering analysis toolbox, vol. 22. Oxford Univ. Press (2006)

[5] Cheng, Y., Church, G.M.: Biclustering of expression data, pp. 93–103 (2000)

[6] Cho, H., Dhillon, I.S.: Coclustering of human cancer microarrays using minimum sum-squared residue coclustering. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 385–400 (2008)

[7] Wee-Chung Liew, A., Law, N.F., Yan, H.: Recent Patents on Biclustering Algorithms for Gene Expression Data Analysis. Recent Patents on DNA &# 38; Gene Sequences 5(2), 117–125 (2011)

[8] Gu, J., Liu, J.: Bayesian biclustering of gene expression data. BMC Genomics 9(1), S4 (2008)

[9] Prelic, A., et al.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22(9), 1122–1129 (2006)

[10] Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., Modha, D.S.: A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 509–514 (2004)

[11] Hussain, S.F., Bisson, G., Grimal, C.: An improved co-similarity measure for document clustering. In: Ninth International Conference on Machine Learning and Applications (ICMLA), pp. 190–197 (2010)
[12] Giannakidou, E., Koutsonikola, V., Vakali, A., Kompatsiaris, Y.: Co-clustering tags and social data sources. In: The Ninth International Conference on Web-Age Information Management, pp. 317–324 (2008)
[13] Bisson, G., Hussain, F.: Chi-Sim: A New Similarity Measure for the Co-clustering Task. In: International Conference on Machine Learning and Applications, pp. 211–217 (2008)
[14] Lemaire, B., Denhière, G.: Effects of high-order co-occurrences on word semantic similarities, Arxiv preprint arXiv:0804.0143 (2008)
[15] Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM 18(11), 620 (1975)
[16] Dudoit, S., Fridlyand, J., Speed, T.P.: Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association 97(457), 77–88 (2002)
[17] Alon, U., et al.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences 96(12), 6745 (1999)
[18] Golub, T.R., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5439), 531 (1999)
[19] Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE Transactions on computational Biology and Bioinformatics, 24–45 (2004)
[20] Tanay, A., Sharon, R., Shamir, R.: Biclustering gene expression data. In: International Conference on Intelligent Systems for Molecular Biology (2002)
[21] Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering local structure in gene expression data: the order-preserving submatrix problem. Journal of Computational Biology 10, 373–384