Biclustering of Microarray Data based on Modular Singular Value Decomposition

Manjunath Aradhya[1,2], Francesco Masulli[1,3], and Stefano Rovetta[1]

(1) Dept of Computer and Information Sciences University of Genova, Via Dodecaneso 35, 16146 Genova, Italy - email: {aradhya|masulli|ste}@disi.unige.it,

(2) Dept of ISE, Dayananda Sagar College of Engg, Bangalore, India - 560078

(3) Sbarro Institute for Cancer Research and Molecular Medicine, Center for Biotechnology, Temple University, BioLife Science Bldg., 1900 N 12th Street Philadelphia, PA 19122 USA

Keywords: Gene expression data, Microarray data, SVD, Biclustering.

Abstract. Unsupervised machine learning methods are widely used in the analysis of gene expression data obtained from microarray experiments. Clustering of data is one of the most popular approaches of analyzing gene expression data. Recently, biclustering approach which has shown to be remarkably effective in a variety of applications that perform simultaneous clustering on the row and column dimension of the data matrix. In this paper, we present a new approach to biclustering called the Modular Singular Value Decomposition (M-SVD-BC) for gene expression. Experimental study on standard datasets demonstrated the effectiveness of the algorithm in gene expression data.

1 Introduction

DNA microarray technology is recent throughput and parallel platform that can provide expression profiling of thousands of genes in different biological conditions [19]. These samples may correspond to different environmental condition, time points, organ and individuals. Examining and analyzing this kind of Bio-informatics data is a strong challenge that can allow us to obtain a depended knowledge on biological phenomena.

DNA microarray data are usually arranged in a matrix, where each row corresponds to a gene and each column an experimental condition. Each entry in the matrix records the expression level of a gene as a real number, which is usually derived by taking the logarithmic of the relative abundance of the mRNA of that genes in a specific condition [14]. An important objective of analyzing this kind of data is the classification of genes and conditions and the identification of regulatory process. With the aim of analyzing such groups and samples, clustering has an important role in the exploratory analysis of microarray data. Techniques derived by clustering can be applied to either genes or conditions to investigate the underlying structure. The resultant clusters produce by these methods reflect the global pattern of expression data, but an interesting cellular process for most cases may be only involved in a subset of genes co-expressed only under a subset of conditions. In order to obtain this kind of structure it is highly desirable to move further and to develop approaches capable of discovering local pattern in microarray data [4].

The term biclustering in gene expression analysis was first introduced in [4], which inspired by Hartigan's [8] so called direct clustering. In the last few years, research on biclustering has gaining popularity for its various potential applications. A detailed survey on biclustering algorithms for biological data analysis can be found in [13]; the paper presents a comprehensive survey on the models, methods and applications in the field of biclustering algorithms. Another interesting survey on biclustering algorithms is also in [17].

Many algorithms have been proposed in literature for biclustering gene expression data. Spectral biclustering of microarray data is proposed in [9], which is based on the observation that checkerboard structures in matrices can be found in eigenvectors corresponding to characteristic expression patterns across genes or conditions. Biclustering of gene expression data by tendency is described in [12]; that proposes a deterministic biclustering model, namely Order Preserving (OP) clustering to capture the set of general tendencies exhibited by a subset of genes along a subset of conditions. A linear time biclustering algorithm for time series gene expression data has been proposed in [14], by finding all maximal consecutive column biclusters under specific assumptions. Experiments are conducted on synthetic and real data of yeast. Improved biclustering of microarray data is presented in [18]; this approach is based on accelerating individual differences clustering and apply binary least squares to update the cluster membership parameters. Genetic algorithm based methods with local search strategy for identifying overlapped biclusters in gene expression data is presented in [15]. An approach to the biclustering problem using the Possibilistic Clustering paradigm is described in [6]; this method finds one bicluster at a time, assigning a membership to the bicluster for each gene and for each condition. The possibilistic clustering is tested on the Yeast database, obtaining fast convergence and good quality solutions. A geometric biclustering algorithm based on the Hough transform for analysis of large scale microarray data is presented in [19]. A method on discovering biclusters in gene expression data based on high-dimensional linear geometries is described in [7].

SVD based methods has also been used in order to obtain biclusters in gene expression data and also in many potential applications [5, 11]. Applying SVD directly on the data may obtain biclusters, but obtaining efficient biclusters on data is still a challenging problem. Hence in this paper we propose modular SVD based method for biclustering in gene expression data. The standard SVD based method may not be very effective under different conditions of gene, since it considers the global information of gene and conditions and represents them with a set of weights. While applying SVD on sub data, local features of genes and conditions can be extracted efficiently in order to obtain better biclusters.

The organization of the paper is as follows: in Sect 2, we explain proposed Modular SVD based method. In Sect 3, we perform experiment on synthetic and real dataset. Finally conclusions are drawn at the end.

2 M-SVD Biclustering Algorithm

In this section we describe our proposed method which is based on modular(sub-data) SVD. The SVD is one of the most important and powerful tool used in numerical signal processing. It is employed in a variety of signal processing applications, such as spectrum analysis, filter design, system identification, etc. SVD based methods has also been used in order to obtain biclusters in gene expression data and also in many potential applications [5, 11]. Applying SVD directly on the data may obtain biclusters, but obtaining efficient biclusters on data is still a challenging problem. The standard SVD based method may not be very effective under different conditions of gene, since it considers the global information of gene and conditions and represents them with a set of weights.

Hence in this work, we made an attempt by overcoming the aforementioned problem by partitioning a gene expression data into several smaller sub-data and then SVD is applied to each of the sub-data separately. The three main steps involved in our method, named M-SVD Biclustering, are:

1. An original whole pattern denoted by a matrix is partitioned into a set of equally sized sub-data in a non-overlapping way.

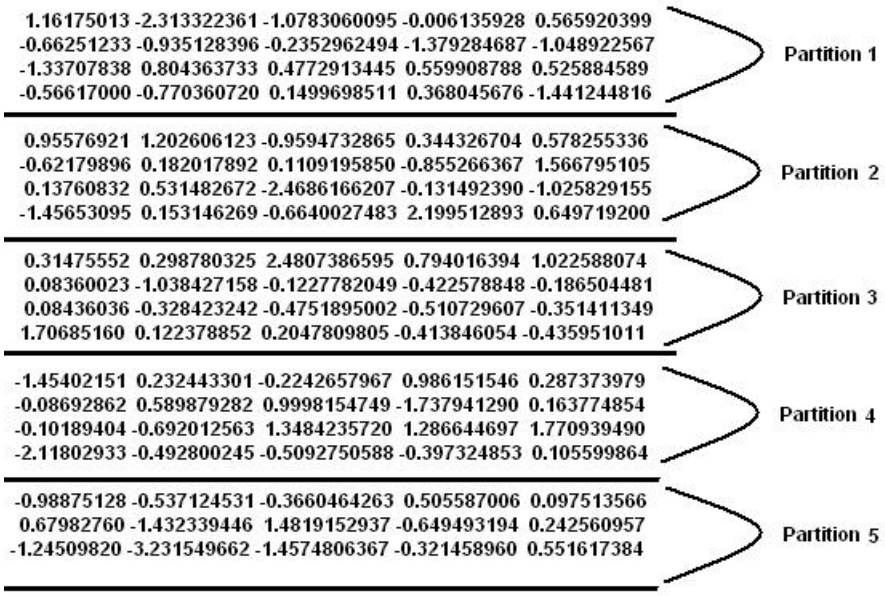

Figure 1: Procedure of applying SVD on partitioned data

2. SVD is performed on each of such sub-data.

3. At last, a single global feature is synthesized by concatenating each sub-data′s.

In order to partition the data, we experimented in two ways. In the first type, we choose rows which are similar by computing mean of the row data. In the second type, we partitioned data in non-overlapping way, which is shown in figure 1. After thorough study, we decided to work with second type of partitioning the data, which leads in better result compared to first type.

Each step of the algorithm is explained in detail as follows:

Data Partition: Let us consider a $m \times n$ matrix A, which contain m genes and n conditions. Now, this matrix A is partitioned into K d-dimensional sub-matrices of similar sizes in a non-overlapping way, where $A = (A_1, A_2, ..., A_K)$ with A_k being the sub-data of A. Figure 1 shows a partition procedure for a given data matrix. Note that a partition of A can be obtained in many different ways e.g., selection groups of continuous rows or groups of continuous columns, or also randomly sampling some rows or some columns.

Apply SVD on K sub-data: Now according to the second step, conventional SVD is applied on K sub-pattern. The SVD provides a factorization for all matrices, even matrices that are not square or have repeated eigenvalues. In general, the theory of SVD states that any matrix A of size $m \times n$ can be factorized into a product of unitary matrices and a diagonal matrix, as follows [10]:

$$A = U\Sigma V^T \tag{1}$$

where $U \in \Re^{m \times m}$ is unitary, $V \in \Re^{n \times n}$ is unitary, and $\Sigma \in \Re^{m \times n}$ has the form $\Sigma = diag(\lambda_1, \lambda_2, ...\lambda p)$, where p is the minimum value of m or n. The diagonal elements of Σ are called the singular values of A and are usually ordered in descending manner. The SVD has the eigenvectors of AA^T in U and of A^TA in V.

The SVD are known to be more robust than usual eigen vectors of covariance matrix. This is because, the robustness are determined by the directional vectors rather than mere scalar quantity like magnitudes (Singular values stored in Σ). Since U and V

matrices are inherently orthogonal in nature, these directions are encoded in U and V matrices. This is unlike the eigenvectors which need not be orthogonal. Hence, a small perturbations like noise have very little effect to disturb the orthogonal properties encoded in the U and V matrices. This we believe could be the main reason for the robust behavior of the SVD.

Finally, from each of the data partitions, we would expect that the eigenvectors corresponding to the largest eigenvalue would provide the optimal clusters. But we also observed that an eignevector with with a small eigenvalue could yield clusters. In our final experiment, instead of clustering each eignevector individually, we perform final clustering step by applying the k-means to the data projected to the best three or four eigenvectors. Finally, we will consider each bicluster size obtained from partitioned data in order to find the final result of Homogeneity H and maximum bicluster size n.

More formally, the proposed method is presented in the form of Algorithm as shown below.

Algorithm: Modular SVD

- **Input: Gene Expression Data**
- **Output: Homogeneity H and Bicluster's size n**
- **Steps:**
  - 1: Acquire gene expression matrix and generate K number of d-dimensional sub-data in a non-overlapping way and reshaped into $K \times n$ matrix $A = (A_1, A_2, ..., A_K)$ with A_k being the sub-data of A.
  - 2: Apply standard SVD method to each sub-data obtained from the Step 1.
  - 3: Perform final clustering step by applying the k-means to the data projected to get the best three or four eigenvectors.
  - 4: Repeat this procedure for all the partition present in the gene expression data.
  - 5: At last, computation of Homogeneity H and size n, are done using the resultant bicluster's obtained from each partition matrix.
- **Algorithm ends**

3 Experiment Results and Comparative Study

In this section we experimentally evaluate the proposed method with pertaining to synthetic and standard dataset. The proposed algorithm has been coded in R language on Pentium IV 2 GHz with 756 MB of RAM under Windows platform. In order to show the performance of the system, we considered two parameters, such as homogeneity *H* and the maximum biclusters size *n*. The size n of a biclusters is usually defined as the number of cells in the gene expression matrix X belonging to it that is the product of cardinality $n_g = |g|$ and $n_c = |c|$:

$$n = n_g \cdot n_c \tag{2}$$

We can define *H* as the mean square residual, a quantity that measures the biclusters homogeneity:

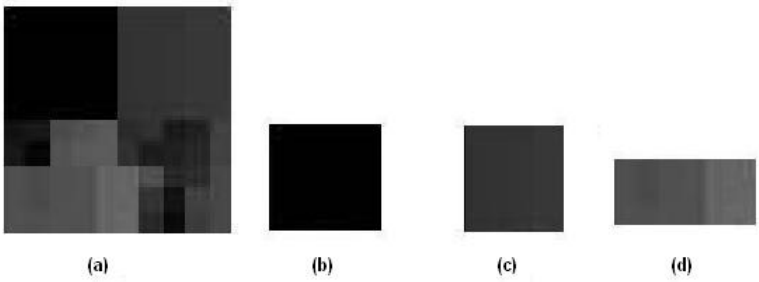

Figure 2: (a):A synthetic dataset with multiple biclusters of different patterns (b-d):and the biclusters extracted

$$H = \sum_{i \in g} \sum_{j \in c} d_{ij}^2 \tag{3}$$

Where $d_{ij}^2 = (x_{ij} + x_{IJ} - x_{iJ} - x_{Ij})/n$, x_{IJ}, x_{Ij} and x_{iJ} are the biclusters mean, the row mean and the column mean of *X* for the selected genes and conditions respectively.

We compared our results with standard spectral method [9] using synthetic and standard dataset. We generated matrices with random values and the size of the matrices varied from $100 \times 10(rows \times columns)$. Table 1 shows the results obtained from the synthetic dataset. From the results it is clear that proposed method based on Modular SVD performs better compared to standard Spectral method which uses SVD for computation.

We also tested our method on another synthetic data to this aim we generated matrices with random numbers, on which 3 biclusters were similar, with dimensions ranging from 3-5 rows and 5-7 columns. Homogeneity H and biclusters size n are tabulated in Table 2. From this table it is ascertained that, the proposed modular approach performs better results compared to standard technique. Figure 2 shows the example of biclusters extracted using synthetic dataset.

Table 1: Homogeneity H and size n for Synthetic Dataset of size 100×10

Methods	Homogeneity (H)	Size (n)
Spectral[9]	—	—
M-SVD-BC	0.90	70

Table 2: Homogeneity H and size n for Synthetic Dataset of size 10×10

Methods	Homogeneity (H)	Size (n)
Spectral[9]	7.1	30
M-SVD-BC	7.21	51

We also tested our proposed method on the standard dataset of Bicat Yeast and SyntrenEcoli. Data structure with information about the expression levels of 419 probesets over 70 conditions follow Affymetrix probeset notation is resulted in Bicat Yeast dataset [2]. Affymetrix data files are normally available in DAT, CEL, CHP and EXP files. Date containing in CDF files can also be used and containd the information about which probes belong to which probe set. For more information on affymetrix can be found in [1]. Results pertaining to this dataset is reported in Table 3. From the results it is clear that the proposed method yields better bicluster size compared to standard method.

Table 3: H and n for standard Bicat Yeast Dataset

Methods	Homogeneity (H)	Size (n)
Spectral[9]	0.721	680
M-SVD-BC	0.789	2840

Another data structure with information about the expression levels of 200 genes over 20 conditions from transcription regulatory network is also used in our experiment [16]. Detail description about Syntren can be found in [3]. The results of Homogeniety and bicluster size is tabulated in Table 4. From these results, it is clear that applying SVD on modular approach yields better performance compared to standard approach.

Table 4: H and n for standard Syntren E. coli Dataset

Methods	Homogeneity (H)	Size (n)
Spectral[9]	19.95	16
M-SVD-BC	7.07	196

4 Conclusions

In this paper, we described biclustering method for gene expression data based on Modular SVD. The proposed method computes SVD on each partitioned data of a given matrix. The standard SVD based method may not be very effective under different conditions of gene, since it considers the global information of gene and conditions and represents them with a set of weights. While applying SVD on modular way, local features of genes and conditions can be extracted efficiently in order to obtain better biclusters. Experiments on synthetic and standard dataset demonstrated the effectiveness of the algorithm.

References

[1] *www.affymetrix.com/analysis/index.affix*.

[2] S. Barkow, S. Bleuler, A. Prelic, P. Zimmermann, and E. Zitzler. Bicat: A biclustering analysis toolbox. *Bioinformatics*, 19:1282–1283, 2006.

[3] Blucke, Leemput, Naudts, Remortel, Ma, Verschoren, Moor, and Marchal. Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithm. *BMC Bioinformatics*, 7:1–16, 2006.

[4] Y. Cheng and Church. Biclustering of expression data. In *Proceedings of the Intl Conf on intelligent Systems and Molecular Biology*, pages 93–103, 2000.

[5] I.S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the 7th ACM SIGKDD*, pages 269–274, 2001.

[6] M. Filippone, F. Masulli, and R. Stefano. Possibilistic approach to biclustering: An application to oligonucleotide microarray data analysis. In *Proceedings of the Computational Methods in System Biology*, pages 312–322, 2006.

[7] X. Gan, Alan, and H. Yan. Discovering biclsuters in gene expression data based on high dimensional linear geometries. *BMC Bioinformatics*, 9:209–223, 2008.

[8] J.A. Hartigan. direct clustering of a data matrix. *Journal of the American Statistical Association*, 67:123–129, 1972.

[9] Y Kluger, Basri, Chang, and Gerstein. Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research*, 13:703–716, 2003.

[10] David C Lay. *Linear Algebra and its Applications*. Addison-Wesley, 2002.

[11] Z. Li, X. Lu, and W. Shi. Process variation dimension reduction based on svd. In *Proceedings of the Intl Symposium on Circuits and Systems*, pages 672–675, 2003.

[12] J. Liu, J. Yang, and W. Wang. Biclustering in gene expression data by tendency. In *Proceedings of the Computational Systems Bioinformatics*, pages 182–193, 2004.

[13] S.C. Madeira and A.L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE & ACM Trans on Computational Biology and Bioinformatics*, 1:24–45, 2004.

[14] S.C. Madeira and A.L Oliveira. A linear time biclustering algorithm for time series gene expression data. In *Proceedings of WABI*, pages 39–52, 2005.

[15] S. Mitra and H. Banka. Mulit-objective evolutionary biclustering of gene expression data. *Pattern Recognition*, 39:2464–2477, 2006.

[16] S. Orr. Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, 31:64–68, 2002.

[17] A. Tanay and R. Sharan R. Shamir. *Biclustering Algorithms : A Survey*. Handbook of Computational Molecular Biology, 2004.

[18] H. Turner, T. Bailey, and W. Krzanowski. Improved biclustering of microarray data demonstrated through systematic performance tests. *Computational Statistics and Data Analysis*, 48:235–254, 2005.

[19] H. Zhao, Alan, X. Xie, and H. Yan. A new geometric biclustering algorithm based on the Hough transform for analysis of large scale microarray data. *Journal of Theoretical Biology*, 251:264–274, 2008.