Biclustering of *DNA* Microarray Data with Early Pruning

Ahmed H. Tewfik and Alain B. Tchagang
Department of Electrical and Computer Engineering
University of Minnesota, 200 Union St. SE Minneapolis, MN 55455

Abstract

Uncovering genetic pathways is equivalent to finding clusters of genes with expression levels that evolve coherently under subsets of conditions. This can be done by applying a biclustering procedure to gene expression data. We propose a new biclustering procedure that derives biclusters from candidate subsets of conditions. These candidate subsets of conditions are identified by comparing pairs of gene expression data. To reduce complexity, the procedure discards early in the candidate subset of conditions formation stage any subset that is predicted to have less than a desired minimum number of conditions. When the biclusters are required to have more than a minimum number of genes, we show that further reduction in complexity can be achieved with no loss of performance by comparing each gene with only a subset of all genes. The proposed approach finds all genes expression levels that evolve coherently under each of the candidate subsets of conditions using a fast approximate pattern matching technique. This approximate pattern matching procedure can find a pattern in a list even if instances of the pattern in the list have random insertions of characters between consecutive characters in the pattern. As compared to prior techniques, the approach finds all maximum size biclusters with a number of conditions greater than a specified minimum. It has a run time equivalent to the fastest of these techniques, even though the fastest biclustering techniques are not guaranteed to find all biclusters.

1. INTRODUCTION

One of the major goals of gene expression data analysis is to uncover genetic pathways, i.e., chains of genetic interactions. For example, a researcher may be interested in identifying the genes that contribute to a disease. This task is difficult because subgroups of genes display similar activation patterns *only* under certain experimental conditions. Genes that are coregulated or coexpressed under a subset of conditions will behave differently under other conditions. Finding genetic pathways therefore requires identifying clusters of genes that are coexpressed under subsets of conditions as opposed to all conditions.

Gene expression data is typically arranged in a data matrix, with rows corresponding to genes and columns to experimental conditions. Conditions can be different environmental conditions or different time points corresponding to one or more environmental condition. The (i,j)th entry of the gene expression matrix represents the expression level of the gene corresponding to row i under the specific condition corresponding to column j. The numerical value of the entry is usually the logarithm of the relative amount of the mRNA of the gene under the specific condition. Finding the genetic pathways is therefore equivalent to simultaneously clustering the rows and columns of the gene expression matrix.

Cheng and Church [1] introduced the term biclustering to denote simultaneous row-column clustering of gene expression data. Biclustering algorithms are also known as bidimensional clustering, subspace clustering and coclustering in other application fields. It should be clear that biclustering techniques produce local models whereas clustering approaches compute global models. If we use a clustering algorithm on the rows of the gene expression matrix, a given gene cluster is defined using all the conditions. In contrast, a biclustering technique will assign a gene to a bicluster based on a subset of conditions. Furthermore, when a clustering algorithm is applied to the rows of the gene expression matrix, it assigns each gene to a single cluster. Biclustering techniques on the other hand identify clusters that are not mutually exclusive or exhaustive. A gene may belong to no cluster, one or more clusters [2].

In this paper we describe a novel biclustering approach for gene expression data. We focus on biclusters with coherent evolutions, wherein gene expression levels stay constant or increase coherently across the subset of conditions selected. To deal with the noise in the gene expression data as well as provide the researcher with flexibility in defining biclusters, we begin by quantizing the entries of the gene expression matrix before applying our biclustering approach. We construct biclusters using a two step procedure. First, we identify candidate subsets of conditions under which pairs of genes display coherent evolutions of expression levels. Next, we use these candidate subsets of conditions to construct the biclusters. We restrict our attention to subsets with more than a pre-specified number of conditions.

To reduce the complexity of the candidate subset of condition identification step, we generate several intermediary ordered lists of conditions for each gene and gene pair to allow the procedure to discard early in the formation of candidate subsets of conditions, any subset that will have less than the desired minimum number of conditions after it is fully completed. Furthermore, if we only seek biclusters with K or more genes, we show that we can reduce the complexity of the procedure by a factor of K with no loss of performance. This is achieved by comparing each gene with only a subset of all genes while identifying candidate subsets of conditions. The benefit of this reduction in complexity cannot be over-emphasized: it allows the procedure to identify all possible biclusters in reasonable time for the large gene expression matrices common in practice.

Finally, a quick approximate pattern matching technique is used to identify all genes that display coherent evolutions of expression levels for all the candidate subsets of conditions identified in the previous step. The proposed pattern matching technique is called "approximate" because it takes into account the fact that the target pattern may appear in the list of ordered conditions for each gene (row in the gene expression matrix) with random insertions of conditions between any two conditions in the pattern. That is, the approximate pattern matching technique can quickly recognize that the list {8, 3, 5, 4, 1, 2, 7, 6} contains the pattern {3, 1, 7}.

Note that subsets of a bicluster also form a valid bicluster. By construction, our procedure will find the maximum size biclusters and will not generate spurious biclusters that are proper subsets of other biclusters. Note also that the worst case complexity of the proposed biclustering procedure is approximately $O(M^2N)$ where M is the number of vector codewords or genes used to identify candidate subsets of conditions and N is the number of conditions.

2. BICLUSTERING APPROACHES

As mentioned in [2], there exists an extensive literature on biclustering techniques, e.g., [3-7]. Almost all of the proposed methods search for one or two types of biclusters among four types that have been identified in the literature [2]: biclusters with constant values, biclusters with constant values on rows and columns, biclusters with coherent values, and biclusters with coherent evolution.

Most techniques are greedy and will miss meaningful biclusters. Some, such as [8], are exhaustive. To ensure a reasonable run time, exhaustive techniques will restrict the maximum size of the bicluster. For example, [8] limits the number of genes that can appear in a bicluster.

Almost all techniques use a cost function to define biclusters. For example, the cost function can measure the square deviation from the sum of the mean value of expression levels in the entire bicluster, and the mean values of expression levels along each row and column in the bicluster. In contrast, we used in our approach a definition of bicluster similar to that of [9]. Specifically, we define a bicluster as a group of genes with expression levels that are non-decreasing across a subset of conditions. As mentioned above, and unlike prior work, we proceed to identify all biclusters by first identifying candidate subsets of condition using a pre-processing of the gene expression data and then constructing the biclusters corresponding to these subsets. This approach avoids the need for exhaustive enumeration or heuristic cost functions that can miss some pertinent biclusters.

3. BICLUSTERING WITH EARLY PRUNING

3.1 Overview

Our biclustering approach identifies groups of genes with expression levels that are non-decreasing across a subset of conditions. The approach is summarized in Algorithm 1 and performs the following steps sequentially:

1. *Quantize gene expression data*. The goal of this step is to reduce the effect of the noise in the gene expression values. In our experimental work, we use the k-means algorithm to quantize the raw expression values when conditions correspond to environmental conditions only (no time data). The number of dictionary entries is typically selected to ensure that the root mean square quantization error is close to the root mean square noise value.

   Note that by changing the size of the dictionary it also possible to define hierarchical biclusters as we move from a dictionary of small size to one with a larger size. The practical significance of such an approach is still being studied.

   When the conditions correspond to time measurements for different environmental conditions, we use a vector quantization approach to reduce the time data to the index of an appropriate dictionary entry. We then substitute the index of the dictionary entry in the cell that corresponds to the underlying condition. The labeling of the vector codewords of the dictionary is assumed to be meaningful.
2. *Re-order quantized gene expression data.* The goal of this step is to produce several intermediate lists that help speed up subsequent steps. First, for each gene, we re-order the conditions in order of non-decreasing expression levels. Conditions with equal quantized expression levels are ordered lexicographically, i.e., a condition corresponding to a smaller column index appears to the left of one corresponding to a larger column index. This step yields a new data matrix in which entries correspond to conditions and columns to the relative order of the conditions. Thus a {3} in cell (4,5) indicates that the 3rd condition corresponds to the 5th smallest quantized expression level for gene 4. We will refer to the resulting matrix as the *gene ordered condition matrix*. Note that in this representation, the quantized expression levels are lost. Only the relative ordering of expression levels versus conditions is maintained.

   The procedure also constructs another matrix, referred to as the *gene condition rank matrix*. Rows in this latter matrix correspond to genes while columns correspond to conditions ordered lexicographically. The entries of the matrix correspond to the relative position of a given condition when conditions are ordered in order of non-decreasing expression levels for a given gene. Thus a {3} in cell (4,5) of the gene condition rank matrix indicates that the 5th condition corresponds to the 3rd smallest quantized expression level for gene 4.
3. *Identify candidate subsets of conditions.* The goal of this step is to extract from the gene expression matrix the candidate subsets of conditions that will be used to define biclusters. This is done through pairwise examination of genes and extraction of ordered subsets of conditions over which the two genes display coherent evolutions of expression levels. This step is the most computationally expensive step of the procedure.

   To reduce the complexity of this step, we use an intermediary condition ordering procedure. Specifically, when comparing two genes, we produce for each condition a list of other conditions with expression levels that are larger in both genes than that of the condition at hand. We refer to the list corresponding to a given condition as the *proto-cluster* corresponding to that condition. We then produce an intermediate list of all conditions ordered according to the cardinality of their corresponding proto-clusters. This pre-processing step allows us to use a sphere decoding like algorithm for identifying candidate subsets of conditions under which the two genes display coherent evolutions of expression levels. As explained below, by examining the cardinality of the proto-clusters corresponding to various conditions, the algorithm can discard early in the subset formation stage, subsets of conditions that are unlikely to have more than the minimum pre-specified number of conditions.

   If we only seek biclusters with K or more genes, the complexity of this step can be reduced by a factor of K. Specifically, we arrange all genes in a circular list and note that, at worst, all pairs of genes appearing in a target cluster will be separated by K/M other genes. Hence, we compare each gene *only* with the M/K genes that follow it. It also appears possible to further reduce complexity by first using a vector quantization step on all genes (rows of the gene expression matrix) and then running this step on the resulting vector codewords rather than the raw data. Note that this could potentially work because the candidate subsets of conditions are generated from the gene ordered condition matrix with no reference to the absolute value of the raw or quantized expression levels. Therefore, as long as the

distance measure and error thresholds used in the vector quantization routines are properly selected, this step will dramatically reduce the number of gene pairs that need to be examined to produce candidate subsets of conditions, without introducing the risk of missing pertinent candidate condition clusters. We are currently studying such an approach.

4. *Generate biclusters.* Once the candidate ordered subsets of conditions are identified a quick approximate pattern matching approach is used to uncover all genes that display expression levels that are non-decreasing across that ordered condition pattern. The complexity of this step is proportional to the number of biclusters that are produced by step 3.
5.

In the remainder of this Section we describe some of the details of steps 3 and 4.

3.2 Candidate subset of conditions identification with early pruning

The two most important procedures used to identify candidate subsets of conditions are the generation of condition proto-clusters and the generation of the actual subsets of conditions.

For a given pair of rows in the gene ordered condition matrix, we generate proto-clusters as follows. For each entry (condition) in the first row, we merge the list of conditions that appear to its right with that corresponding to those that appear to the right of the same condition in the second row. Next, using a quick sort we determine which conditions appear twice in the merged list. This analysis is then used to generate the proto-cluster matrix with rows and columns corresponding to conditions and entries equal to 0 or 1. A 1 in cell (i,j) in the proto-cluster matrix indicates that condition j is to the right of condition i in both genes under consideration. By summing up all entries along a row i, we have an estimate of the maximum size of any candidate subset of conditions that starts with condition i .

Generation of the candidate subsets of conditions starts a list with the condition that has the largest number of conditions to its right as captured by the proto-cluster matrix. Next, it scans all conditions in the proto-cluster corresponding to the condition appearing in the first position of the list in order of decreasing proto-cluster size. Let j denote an entry from that proto-cluster under consideration. For each list that has already been initialized, the procedure determines whether j is in the proto-clusters of all entries already in the list. If it is, it appends it to the list. If it is not, the procedure determines whether j is in the proto-clusters of a sublist of sequential entries in the list, starting with the first entry. Suppose it is and let S_j be the longest such sublist. The procedure then determines whether the sum of the size of S_j plus the size of the *jth* proto-cluster is larger than the minimum subset of conditions size specified. If it is, a new list is created by appending j to S_j.

Note that if the size of the proto-cluster corresponding to j is larger than the minimum subset of conditions size specified, this step will also create a new list with j as its first element.

Note also that determining whether j is in the proto-clusters of all entries already in a given list can easily be evaluated by multiplying all the entries corresponding to the elements of the list in the *jth* column of the proto-cluster matrix. If the product is 1, j is indeed in the proto-clusters of all entries in the list.

Finally, to avoid duplicates, the procedure must merge candidate subsets of conditions generated by comparing a given pair of genes with all candidates generated up to that point, eliminating duplicates.

It can be shown that by construction, the procedure will generate subsets of maximum size. It will not generate subsets of these maximum size subsets, unless the larger subsets do not appear when comparing certain pairs of genes, while the smaller subsets do.

3.3 Bicluster generation and approximate pattern matching

Bicluster generation is relatively easy once the candidate condition clusters have been identified. Each candidate condition cluster is a list of conditions, or columns in the quantized gene expression level and the gene condition rank matrices. To find all genes that have expression levels that are non-decreasing across the candidate subset of conditions, we select the columns of gene condition rank matrix corresponding to the conditions in the subset under consideration. The columns are ordered in the same order that the conditions appear in the candidate subset. Next, the procedure calculates the first order difference of the entries across each row (gene). The resulting rows that have no negative entry correspond to genes in the bicluster defined by the given subset of conditions.

4. RESULTS

We applied the proposed biclustering technique to the yeast gene microarray data that can be found at [10]. The data consists of 8224 genes and 17 conditions. Each gene expression level is represented as a 4 byte real number. We quantized the expression data using a dictionary with 200 codewords determined by the k-means algorithm. We determined all biclusters with 7 or more conditions.

A complete discussion of the results can be found in [11]. Because of the large number of biclusters found, we will present here a few illustrative results that will help the reader grasp the magnitude of the problem and the nature of the results produced by the algorithm. For example, Fig. 1 shows a histogram of the number of conditions in the first 59,000 biclusters produced by the procedure. Figs. 2 and 3 show the gene expression levels of 11 genes in a bicluster with 11 genes and 10 conditions, across all conditions and only the conditions in the bicluster respectively. Note that across the conditions in the bicluster, the gene expression levels are indeed non-decreasing. The behavior displayed in Fig. 3 is characteristic of all biclusters.

Finally note that the proposed technique has performance advantages over previously reported approaches. Its running time is comparable to that of [4] for the user selected parameters given in that reference even though, unlike [4], it finds all biclusters with more than a given number of conditions. Its running time is much better than that of [1] which reportedly takes 300-400s to find a single bicluster.

5. REFERENCES

[1] Y. Cheng, G.M. Church, "Biclustering of Expression Data," In *Proc. ISMB'00*, pages 93-103. AAA *I* Press, 2000.

[2] S. C. Madeira, A. L. Oliveira, "Biclustering Algorithms for Biological Data Analysis: A Survey," *IEEE Transactions on computational Biology and Bioinformatics*, Vol. 1, No. 1, Jan-March 2004.

[3] G. Getz, E. Levine, E. Domany, "Coupled Two-way Clustering Analysis of Microarray Data," *Proc. Natl. Acad. Sci. USA*, 97(22): 12079-84, 2000.

[4] H. Wang, W. Wang, J. Yang, and P.S. Yu, "Clustering by Pattern Similarity in Large Data Sets," *Proc. 2002 ACM SIGMOD Int'l Conf. Management of Data*, pp. 394-405, 2002.

[5] R. Sharan, A. Maron-Katz, N. Arbili, R. Shamir, "CLICK and EXPANDER: a System for Clustering and Visualizing Gene Expression Data," *Bioinformatics*, 2003.

[6] Y. Kluger, R. Barsi, JT. Cheng, M. Gerstein, "Spectral Biclustering of Microarray Data: Coclustering Genes and Conditions," *Genome Res.*, 13(4): 703-16, 2003.

[7] L. Lazzeron *i*, A. Owen, "Plaid Models for Gene Expression Data," *Statistica Sinica*, 12: 61-86, 2002.

[8] A. Tanay, R. Sharan, and R. Shamir, "Discovering Statistically Significant Biclusters in Gene Expression Data," *Bioinformatics*,vol. 18, pp. S136-S144, 2002.

[9] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini, "Discovering Local Structure in Gene Expression Data: The Order-Preserving Submatrix Problem," *Proc. Sixth Int'l Conf. Computational Biology (RECOMB '02)*, pp. 49-57, 2002.

[10] S. Tavazoie, J. Hughes, M. Campbell, R. Cho, and G. Church. Yeast micro data set. At http://arep.med.harvard.edu/biclustering.

[11] A. B. Tchagang and A. H. Tewfik, *Robust Biclustering Algorithm*s, Technical Report, University of Minnesota, 2005.

.

Input: MxN gene expression matrix
Output: Identify all biclusters with K or more genes and L or more conditions
Do:

1. Quantize gene expression matrix
2. Reorder gene expression data
   - 2.1. Generate gene condition rank matrix
   - 2.2. Generate gene ordered condition matrix
3. Arrange genes in a circular list.
4. Compare each gene in the circular list with the M/K genes that follow it:
   - 4.1. generate proto-clusters for all conditions
   - 4.2. scan conditions in order of decreasing proto-cluster size
     - 4.2.1. Append condition to existing candidate subsets of conditions if feasible, or
     - 4.2.2. start new subset of conditions if the subset is predicted to have more than L conditions when completed
5. Use approximate pattern matching approach to identify all genes with expression levels that are non-decreasing across each candidate subset of conditions.

Algorithm 1: Proposed biclustering approach

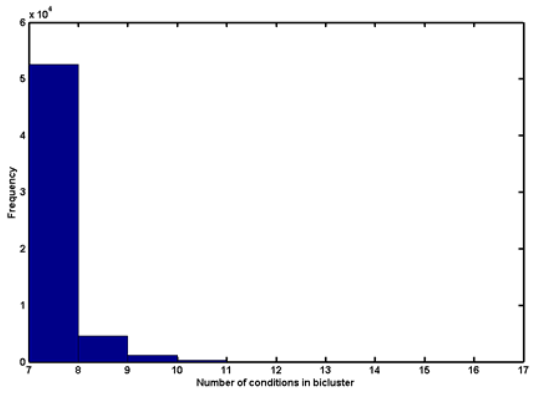

Fig. 1 Histogram of number of conditions in a subset of 59,000 biclusters.

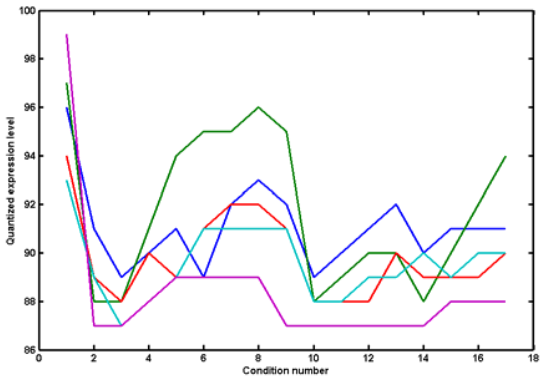

Fig. 2: Quantized gene expression levels for genes in a bicluster with 10 conditions for all conditions. Different lines correspond to different genes.

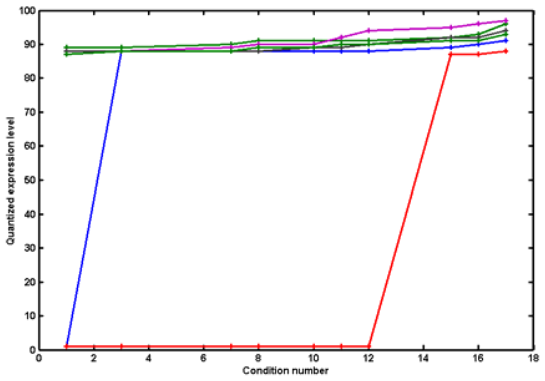

Fig 3: Quantized gene expression levels for genes in a bicluster with 10 conditions for conditions in bicluster only. Different lines correspond to different genes. Crosses indicate data plotted.