Algorithms for Bounded-Error Correlation of High Dimensional Data in Microarray Experiments *

Mehmet Koyutürk, Ananth Grama, and Wojciech Szpankowski
Department of Computer Sciences, Purdue University
West Lafayette, IN, 47907, USA.
{koyuturk, ayg, spa}@cs.purdue.edu

Abstract

The problem of clustering continuous valued data has been well studied in literature. Its application to microarray analysis relies on such algorithms as k-means, dimensionality reduction techniques, and graph-based approaches for building dendrograms of sample data. In contrast, similar problems for discrete-attributed data are relatively unexplored. An instance of analysis of discrete-attributed data arises in detecting co-regulated samples in microarrays. In this paper, we present an algorithm and a software framework, PROXIMUS*, for error-bounded clustering of high-dimensional* discrete *attributed datasets in the context of extracting co-regulated samples from microarray data. We show that* PROXIMUS *delivers outstanding performance in extracting accurate patterns of gene-expression.*

1. Introduction

Analysis of large datasets from microarray experiments traditionally takes the form of clustering high-dimensional data with a view to correlating samples. This is traditionally done using eigenvalue/singular value decomposition (PCA/rank reduction), k-means clustering, least squares methods, etc. These analysis techniques view individual samples as points in high dimensional space and build dendrograms that cluster spatially proximate points together in a hierarchy. One problem with this approach is that all dimensions in these datasets are treated identically and local up- or down-regulation can be masked by gross behavior over the entire experiment. While this can be addressed by scaling dimensions, determining scaling coefficients is itself difficult.

Another challenge associated with analyzing microarray data is to determine samples that are co-regulated (up- and down-regulated together) and to detect motifs responsible for this co-regulation. The objective function in this case is defined over a discrete space of up- and down-regulation and is therefore not amenable to clustering techniques that treat it as continuous data. The starting point of such analysis is a discretized vector derived from continuous microarray (expression) data. A down-regulation during an interval (with respect to previous interval) is discretized to value 0, and to value 1, otherwise. A large set of such discrete vectors must now be correlated to determine (i) sets of samples that are co-regulated, and (ii) regions within a vector that display strong correlations.

In order to overcome the computational requirements of this problem while providing efficient analysis of data we propose a new technique – binary($\{0,1\}$) non-orthogonal matrix transformation to extract dominant patterns. In this technique, elements of singular vectors of a discrete, positive valued matrix are constrained to binary entries with an associated singular value of 1. In contrast, in a related technique called Semi-Discrete Decomposition (SDD), elements of singular vectors are in the set $\{-1,0,1\}$ and the associated singular value is continuous. We show here that our variant results in an extremely efficient algorithm and powerful framework within which large discretized microarray datasets can be analyzed.

PROXIMUS is a non-orthogonal matrix transform based on recursive partitioning of a dataset depending on the distance of a specific (discretized) gene expression pattern from the dominant pattern. The dominant pattern is computed as a binary singular vector of the matrix of discretized vectors. PROXIMUS computes only the first singular vector and consequently, each discovered pattern has a physical interpretation at all levels in the hierarchy of the recursive process. For the discovery of the dominant singular vector, we adopt an iterative alternating heuristic. Due to the discrete nature of the problem, initialization of singular vectors is critical for convergence to desirable local optima. We derive effective initialization strategies, along with algorithms for a multiresolution analysis of discretized mi-

*The authors would like to acknowledge support from National Institutes of Health and the National Science Foundation.

croarray datasets. We demonstrate excellent accuracy and runtime of our methods on four microarray datasets.

2. Background and Related Work

Much of the existing literature on microarray analysis focuses on clustering high-dimensional datasets. These clustering techniques range from k-means methods to matrix transformations such as truncated singular value decomposition (SVD) and rank-reduction, semi-discrete decomposition (SDD), centroid decomposition, and principal direction divisive partitioning (PDDP) [1, 3, 5]. SDD is a variant of SVD in which the values of the entries in matrices U and V are constrained to be in the set $\{-1, 0, 1\}$ [5]. Centroid Decomposition (CD) is an approximation to SVD that is widely used in factor analysis. It has been shown empirically that CD provides a measurement of second order statistical information of the original data [3].

Our approach differs from these methods in that it discovers naturally occurring patterns in discretized microarray data with no constraint on cluster sizes or number of clusters. Thus, it provides a generic interface to the microarray analysis problem. Furthermore, the superior execution characteristics of our approach make it particularly suited to extremely high-dimensional attribute sets (well beyond those currently encountered in high-throughput microarray experiments).

3. Non-Orthogonal Decomposition of Binary Matrices

PROXIMUS is a collection of novel algorithms and data structures that rely on modified SDD to find error-bounded correlations of binary attributed datasets. While relying on the idea of non-orthogonal matrix transforms, PROXIMUS provides a framework that captures the properties of discrete datasets more accurately and takes advantage of their binary nature to improve both the quality and efficiency of the analysis. Our approach is based on recursively computing discrete rank-one approximations to the matrix to extract dominant patterns hierarchically [6].

A binary rank-one approximation for a matrix is defined as an outer product of two binary vectors that is at minimum Hamming distance from the matrix over all outer products. In other words, the rank-one approximation problem for matrix A with m columns and n rows is one of finding two vectors x and y that maximize the number of zeros in the matrix $(A - xy^T)$, where x and y are of dimensions m and n, respectively. Here, vector y is the *pattern vector* which is the best approximation for the objective (error) function specified. Vector x is the *presence vector* representing the rows of A that are well approximated by the pattern described by y.

Conventional singular value decompositions (SVDs) can be viewed as summations of rank-one approximations to a sequence of matrices. Here, the first matrix is the original matrix itself and each subsequent matrix is a residual matrix, *i.e.*, the difference between the given matrix and the matrix produced by sum of previous rank-one approximations. However, the application of SVDs to binary matrices has two drawbacks. First, the resulting decomposition contains non-integral vector values, which is generally hard to interpret for binary datasets. SDD partially solves this problem by restricting the entries of singular vectors to the set {-1, 0, 1}. However, the second drawback is associated with the idea of orthogonal decomposition, and therefore, SDD also suffers from this problem: if the underlying data consists of non-overlapping (orthogonal) patterns only, SVD successfully identifies these patterns. However, if the patterns with similar strengths overlap, then, because of the orthogonality constraint, the features contained in some of the previously discovered patterns are extracted from each pattern. Furthermore, in orthogonalizing the second singular vector with respect to the first, SVD introduces negative values into the second vector. There is no easy interpretation of these negative values in the context of up- or down-regulation of genes (recall that a 0 corresponds to a down-regulation and 1 otherwise).

Based on these observations, our modification to SDD for binary matrices has two major components: (i) pattern and presence vectors are restricted to binary elements; and (ii) the matrix is partitioned based on the presence vector after each computation of rank-one approximation, and the procedure is applied recursively to each partition. This method provides a hierarchical representation of dominant patterns.

3.1. Discrete Rank-one Approximation of Binary Matrices

The problem of finding the optimal discrete rank-one approximation for a binary matrix can be stated as follows.

Definition 3.1 Rank-one approximation
Given matrix $A \in \{0,1\}^m \times \{0,1\}^n$, find $x \in \{0,1\}^m$ and $y \in \{0,1\}^n$ to minimize the error:

$$||A - xy^T||_F^2 = |\{a_{ij} \in (A - xy^T) : |a_{ij}| = 1\}|. \quad (1)$$

In other words, the error for a rank-one approximation is the number of nonzero entries in the residual matrix. This problem is closely related to finding maximal cliques in graphs. This problem is known to be NP-hard and there exist no known approximation algorithms or effective heuristics in literature. As a matter of fact, if we view the problem as one of discovering significant patterns in the matrix, the optimal solution is not necessarily the desired rank-one approximation [6].

Alternating Iterative Heuristic Since the objective (error) function can be written as

$$||A - xy^T||_F^2 = ||A||_F^2 - 2x^T Ay + ||x||_2^2 ||y||_2^2, \quad (2)$$

minimizing the error is equivalent to maximizing

$$C_d(x, y) = 2x^T Ay - ||x||_2^2 ||y||_2^2. \quad (3)$$

If we fix y and set $s = Ay$, the corresponding x that maximizes this function is given by the following equation.

$$x(i) = \begin{cases} 1, & if\ 2s(i) \geq ||y||_2^2 \\ 0, & otherwise \end{cases} \quad (4)$$

This equation follows from the idea that a nonzero element of x can have a positive contribution to $C_d(x, y)$ if and only if at least half of the nonzero elements of y match with the nonzero entries on the corresponding row of A. Clearly, this equation leads to a linear time algorithm in the number of nonzeros of A to compute x, as computation of s requires $O(nz(A))$ time and Equation 4 can be evaluated in $O(m)$ time. Similarly, we can compute vector y that maximizes $C_d(x, y)$ for a fixed x in linear time. This leads us to an alternating iterative algorithm based on the computation of SDD [5], namely initialize y, then solve for x. Now, we solve for y based on updated value of x. We repeat this process until there is no improvement in the objective function. Indeed, this technique is distantly related to expectation maximization, which is a commonly used technique in statistical analysis [4].

Although the objective function of Equation 3 leads to a linear time algorithm and guarantees convergence to a local maximum, it has a significant drawback due to the discrete nature of the domain. Specifically, this algorithm does not have any global awareness, *i.e.*, it always converges to the local maximum closest to initialization. This leaves the task of solving the problem to suitable initialization of the pattern vector. A continuous objective function approximating $C_d(x, y)$, addresses this problem, since it is more successful in forcing convergence to desired local maxima, especially for sparse matrices.

Approximate Continuous Objective Function In the case of decomposing continuous valued matrices, it has been shown that the objective function of rank-one approximation is equivalent to maximizing

$$C_c(x, y) = \frac{(x^T Ay)^2}{||x||_2^2 ||y||_2^2}. \quad (5)$$

Although this function is not equivalent to the objective function in the case of binary matrices, *i.e.*, $C_d(x, y)$ and $C_c(x, y)$ do not have their global maximum at the same point, the behavior of these two functions is highly correlated. Thus, we can use $C_c(x, y)$ as a continuous approximation to $C_d(x, y)$. Fixing y and letting $s = Ay/||y||_2^2$ as above, the objective becomes one of maximizing $\frac{(x^T s)^2}{||x||_2^2}$. This can be done in linear time by sorting elements of s via counting sort and visiting elements of x in the resulting order until no improvement in the objective function is possible.

This continuous function has the desirable property of having a broader range of convergence compared to the discrete objective function. Furthermore, since the rate of growth of this function declines less rapidly with increasing number of nonzeros in x, it favors discovery of sparser patterns. Although a local maximum of $C_c(x, y)$ does not necessarily correspond to a local maximum of the objective function, it may correspond to a point that is close to a local maximum and has a higher objective value than many undesirable local maxima. Note that although this metric provides more flexibility in initialization, selection of the initial pattern vector still has a significant impact on the quality of the solution due to the discrete nature of the domain.

3.2. Recursive Decomposition of Binary Vectors

We use the rank-one approximation of the given matrix to partition the gene regulation vectors into two groups.

Definition 3.2 Partitioning based on rank-one approximation:
Given rank-one approximation $A \approx xy^T$, a partition of A with respect to this approximation is defined by two submatrices A_1 and A_0, where

$$A(i) \in \begin{cases} A_1, & if\ \ x(i) = 1 \\ A_0, & otherwise \end{cases}$$

for $1 \leq i \leq m$. Here, $A(i)$ denotes the i^{th} row of A.

The intuition behind this approach is that the rows corresponding to 1's in the presence vector are the rows of a maximally connected submatrix of A. Therefore, these rows have more similar non-zero structures among each other compared to the rest of the matrix. This partitioning can also be interpreted as creating two new groups of genes, A_0 and A_1. Since the rank-one approximation for A gives no information about A_0, we further find a rank-one approximation and partition this matrix recursively. On the other hand, we use the representation of the rows in A_1 given by the pattern vector y and check if this representation is adequate via a stopping criterion. If so, we decide that matrix A_1 is adequately represented by matrix xy^T and stop; else, we recursively apply the procedure for A_1 as for A_0.

The partitioning-and-approximation process continues until the matrix cannot be further partitioned or the resulting approximation adequately represents the entire group.

We define a metric, called normalized Hamming radius, to measure the adequacy of the representation in terms of the Hamming distances of rows to the underlying pattern vector.

Definition 3.3 Normalized Hamming distance
Given two binary vectors x and y, the normalized Hamming distance between x and y is defined as:

$$\hat{h}(x,y) = \frac{x^T x + y^T y - 2x^T y}{n},$$

where $||x|| = ||x||_2^2 = ||x||_1$ is the number of nonzeros in an n-dimensional binary vector x.

Normalized Hamming distance measures the fraction of unmatched nonzeros between x and y among all nonzeros of x and y. Note that $0 \leq \hat{h}(x,y) \leq 1$. The normalized Hamming distance between a row of the matrix and a pattern vector measures the fraction of the row that is not represented by the pattern as well as the fraction of the pattern that does not exist in the row. Thus, the normalized Hamming distance provides a measure for detecting mismatched patterns as well as underrepresentation of a row by the underlying pattern.

Definition 3.4 Normalized Hamming radius
Given a set of binary vectors $X = \{x_1, x_2, \ldots, x_n\}$ and a binary vector y, the normalized Hamming radius of X centered around y is defined as:

$$\hat{r}(X,y) = \max_{1 \leq i \leq n} \hat{h}(x,y).$$

We use the normalized Hamming radius as the major stopping criterion for the algorithm to determine when a group of regulation patterns is sufficiently correlated. The recursive algorithm does not partition subgroup A_i further if one of the following two conditions holds for the rank-one approximation $A_i \approx x_i y_i^T$.

- $\hat{r}(A_{i1}, y_i) < \epsilon$, where ϵ is the prescribed bound on the normalized Hamming radius of identified clusters.
- $x_i(j) = 1\, \forall j$, *i.e.*, all the rows of A_i are present in A_{i1}.

If one of the above conditions holds, the pattern vector y_i is identified as a dominant regulation pattern of group A.

3.3. Initialization of Iterative Process

While finding a rank-one approximation, initialization is crucial for not only the rate of convergence but also the quality of the solutions since a wrong choice can result in poor local minima. In order to find a feasible solution, the initial pattern vector should have magnitude greater than zero, *i.e.*, at least one of the entries in the initial pattern vector should be equal to one. One possible initialization scheme is to select a separator column and to identify rows that have a nonzero on that column. We then initialize the pattern vector to the centroid of these rows. In our implementation, we select the dimension that yields the most balanced partition in order to increase the probability of partitioning along a significant dimension.

4. Experimental Results

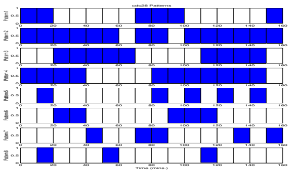
We demonstrate the use of PROXIMUS in the context of analysis of microarray data. Conventional analysis techniques have focused on clustering techniques for building dendrograms for gene expression data. While this is useful for grouping gene expression based on gross behavior over the experiment, our objective is to examine co-regulation (up- and down-regulation) in groups of genes. With this goal, we convert expression data for each gene into a binary vector of length equal to number of samples. Each component of the vector is assigned a value 0 if expression was down-regulated during the period (w.r.t. previous period) and 1 otherwise. We then apply PROXIMUS to this set of discrete-valued vectors to determine a suitable set of representative patterns along with a partitioning (and assignment) of the genes to these patterns. Each partition represents a set of genes that are co-regulated to within specified tolerance. This partitioning can then be used to identify motifs in genes that control regulation.

Experiment	No. of patterns	# samples in each pattern
alpha	13	[200, 46, 41, 54, 41, 48, 50, 52, 32, 111, 60, 30, 34]
cdc15	10	[174, 58, 69, 35, 58, 65, 73, 134, 80, 53]
cdc28	8	[322, 29, 257, 24, 88, 36, 31, 12]
elu	7	[433, 173, 104, 32, 31, 14, 12]

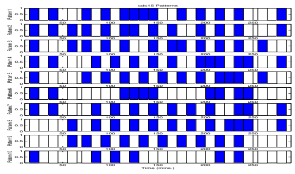
Table 1. Summary of regulation patterns discovered by PROXIMUS in each experiment.

We apply our method to microarray data from four experiments on yeast cultures synchronized by the following methods: α-factor arrest (dataset Alpha), elutriation (dataset Elu), and arrest of cdc15/cdc28 (datasets cdc15/cdc28) temperature-sensitive mutants (Spellman et al. [7], Cho et al. [2]). Dataset Alpha corresponds to samples taken at 7-minute intervals for 140 minutes, dataset cdc15 contains samples taken every 10 minutes for 300 minutes, dataset cdc28 contains samples taken every 10 minutes for 160 minutes, and dataset Elu contains samples taken every 30 minutes for 330 minutes.

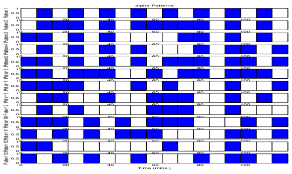
The first step in our analysis is the determination of appropriate threshold (error with respect to representative pat-



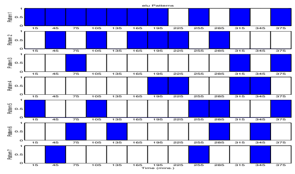
(a) Dataset CDC28.



(b) Dataset CDC15.



(a) Dataset Alpha.



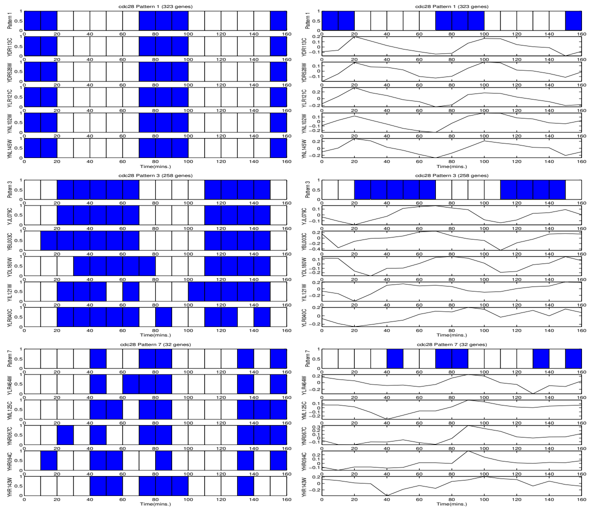
(b) Dataset Elu.

Figure 1. Up-regulation and down-regulation patterns extracted from four datasets. The shaded regions indicate clusters that are up-regulated and empty regions indicate down-regulation. Each of these patterns correspond to clusters of genes that exhibit this behavior. Some of these clusters along with individual up- and down-regulation are illustrated in Figures 2 and 3.

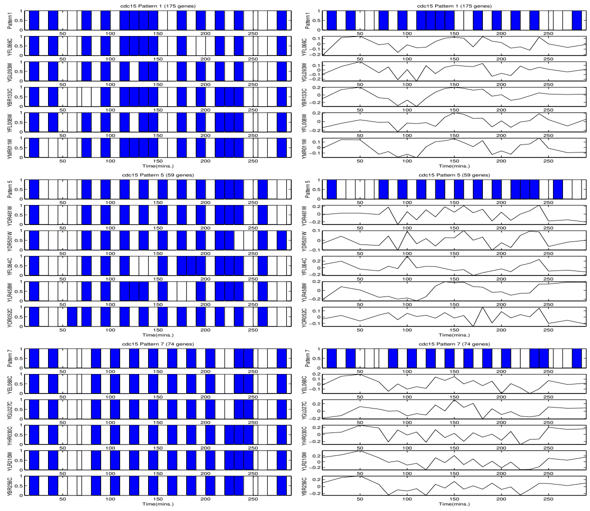
tern) for partitioning data into correlated sets. The number of partitions, along with the number of samples in each partition is illustrated in Table 1. For example, dataset alpha is partitioned into 13 groups, with groups containing 200, 46, ..., samples, respectively. The selection of appropriate error threshold is important because a low threshold results in each sample being identified as a pattern, itself. Conversely, a high threshold results in poor patterns.

In Figure 1, we illustrate the patterns extracted from the four datasets (8, 10, 13, and 7 patterns from cdc28, cdc15, alpha, and elu, respectively). The dark (blue) regions represent periods of up-regulation and unshaded regions represent periods of down-regulation. In Figures 2 and 3, we select some of the patterns from each dataset and demonstrate the excellent clustering properties of PROXIMUS. For example, in the top panel of Figure 2, we illustrate 3 patterns from dataset cdc28. The top pattern in each case is the representative pattern and the following five rows correspond to five randomly chosen samples from the data. The left panel illustrates the pattern in comparison to actual up- and down-regulation data (0/1 discretized expression data) and the right panel illustrates the pattern along with actual regulation data. It is evident that with very high accuracy, PROXIMUS captures patterns in up- and down-regulation of expression. This is reflected both in the discretized data, as well as continuous sampled data.

We illustrate three randomly selected patterns along with

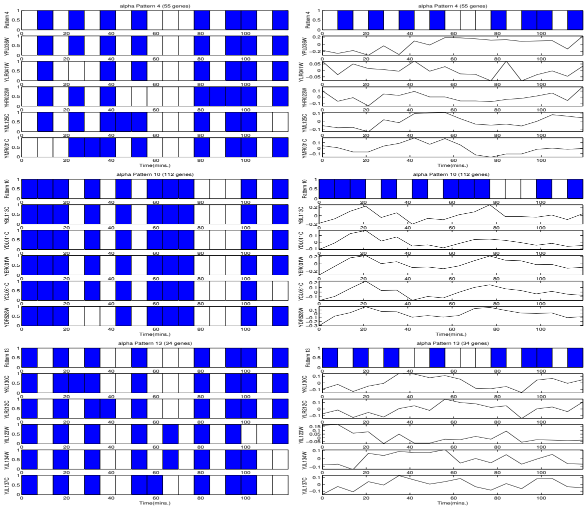


Clusters 1, 3, 7 of dataset CDC28.

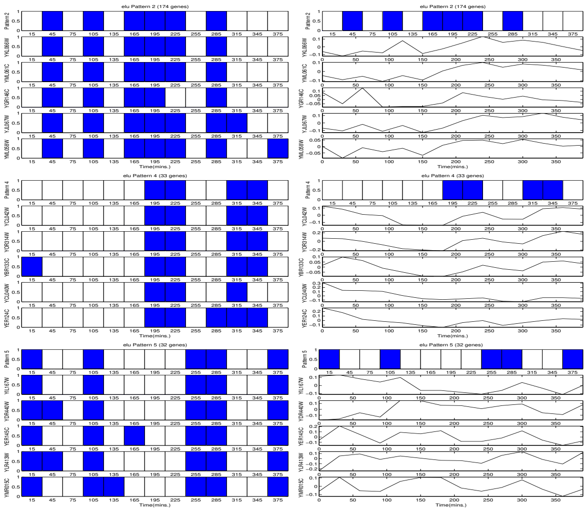


Clusters 1, 5, 7 of dataset CDC15.

Figure 2. Selected clusters from datasets CDC28 and CDC15, the representative patterns of these clusters and some members of the clusters illustrating excellent co-regulation properties. The left column shows up/down regulation and right column illustrates individual values. In each case, the first row is the representative pattern computed by PROXIMUS and subsequent rows correspond to experimental data input to PROXIMUS.



Clusters 4, 10, 13 of dataset Alpha.



Clusters 2, 4, 5 of dataset Elu.

Figure 3. (Figure 2 Continued) Selected clusters from datasets Alpha and Elu, the representative patterns of these clusters and some members of the clusters illustrating excellent co-regulation properties. The left column shows up/down regulation and right column illustrates individual values. In each case, the first row is the representative pattern computed by PROXIMUS **and subsequent rows correspond to experimental data input to** PROXIMUS**.**

five samples corresponding to each of these three patterns (along with the actual sample data in right panel) for all four experiments in Figures 2 and 3. In each case the correlation within each cluster with respect to up- and down- regulation is observed to be very strong. We are currently in the process of identifying motifs in all of these samples and to correlate motifs in clusters to their up- and down-regulation behavior. We expect to present this data in the final version of this paper.

5. Conclusions and Ongoing Work

In this paper, we have presented and used a novel technique, PROXIMUS, for analyzing discrete attributed data. We use this technique to identify co-regulated samples in microarray experiments and demonstrated excellent results. We are currently in the process of identifying motifs in clusters induced by PROXIMUS and to relate these motifs to underlying regulatory mechanisms. PROXIMUS is available for free download at `http://www.cs.purdue.edu/homes/koyuturk/proximus/`.

References

[1] D. Boley. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4):325–344, 1998.

[2] R. J. Cho, M. J. Campbell, E. A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. G. Wolfsberg, A. E. Gabrielian, D. Landsman, D. J. Lockhart, and R. W. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2(1), July 1998.

[3] M. T. Chu and R. E. Funderlic. The centroid decomposition: Relationships between discrete variational decompositions and SVDs. *SIAM J. Matrix Anal. Appl.*, 23(4):1025–1044, 2002.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39 (1):1–38, 1977.

[5] T. G. Kolda and D. P. O'Leary. Computation and uses of the semidiscrete matrix decomposition. *ACM Transactions on Information Processing*, 1999.

[6] M. Koyutürk, A. Grama, and N. Ramakrishnan. Algebraic techniques for analysis of large discrete-valued datasets. In *Proc. 6th European Conf. on Principles of Data Mining and Knowledge Discovery (PKDD 2002)*, pages 311–324, 2002.

[7] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.