Text Mining Biomedical Literature for Discovering Gene-to-Gene Relationships: A Comparative Study of Algorithms

Ying Liu, Shamkant B. Navathe, Jorge Civera, Venu Dasigi, Ashwin Ram, Brian J. Ciliax, and Ray Dingledine

Abstract—Partitioning closely related genes into clusters has become an important element of practically all statistical analyses of microarray data. A number of computer algorithms have been developed for this task. Although these algorithms have demonstrated their usefulness for gene clustering, some basic problems remain. This paper describes our work on extracting functional keywords from MEDLINE for a set of genes that are isolated for further study from microarray experiments based on their differential expression patterns. The sharing of functional keywords among genes is used as a basis for clustering in a new approach called BEA-PARTITION in this paper. Functional keywords associated with genes were extracted from MEDLINE abstracts. We modified the Bond Energy Algorithm (BEA), which is widely accepted in psychology and database design but is virtually unknown in bioinformatics, to cluster genes by functional keyword associations. The results showed that BEA-PARTITION and hierarchical clustering algorithm outperformed k-means clustering and self-organizing map by correctly assigning 25 of 26 genes in a test set of four known gene groups. To evaluate the effectiveness of BEA-PARTITION for clustering genes identified by microarray profiles, 44 yeast genes that are differentially expressed during the cell cycle and have been widely studied in the literature were used as a second test set. Using established measures of cluster quality, the results produced by BEA-PARTITION had higher purity, lower entropy, and higher mutual information than those produced by k-means and self-organizing map. Whereas BEA-PARTITION and the hierarchical clustering produced similar quality of clusters, BEA-PARTITION provides clear cluster boundaries compared to the hierarchical clustering. BEA-PARTITION is simple to implement and provides a powerful approach to clustering genes or to any clustering problem where starting matrices are available from experimental observations.

Index Terms—Bond energy algorithm, microarray, MEDLINE, text analysis, cluster analysis, gene function.

1 Introduction

DNA microarrays, among the most rapidly growing tools for genome analysis, are introducing a paradigmatic change in biology by shifting experimental approaches from single gene studies to genome-level analyses [1], [2]. Increasingly accessible microarray platforms allow the rapid generation of large expression data sets [3]. One of the key challenges of microarray studies is to derive biological insights from the unprecedented quantities of data on gene-expression patterns [5]. Partitioning genes into closely related groups has become an element of practically all analyses of microarray data [4].

A number of computer algorithms have been applied to gene clustering. One of the earliest was a hierarchical algorithm developed by Eisen et al. [6]. Other popular algorithms, such as k-means [7] and Self-Organizing Maps (SOM) [8] have also been widely used. These algorithms have demonstrated their usefulness in gene clustering, but some basic problems remain [2], [9]. Hierarchical clustering organizes expression data into a binary tree, in which the leaves are genes and the interior nodes (or branch points) are candidate clusters. True clusters with discrete boundaries are not produced [10]. Although SOM is efficient and simple to implement, studies suggest that it typically performs worse than the traditional techniques, such as k-means [11].

Based on the assumption that genes with the same function or in the same biological pathway usually show similar expression patterns, the functions of unknown genes can be inferred from those of the known genes with similar expression profile patterns. Therefore, expression profile gene clustering by all the algorithms mentioned above has received much attention; however, the task of finding functional relationships between specific genes is left to the investigator. Manual scanning of the biological literature (for example, via MEDLINE) for clues regarding potential functional relationships among a set of genes is not feasible when the number of genes to be explored rises above approximately 10. Restricting the scan (manual or automatic) to annotation fields of GenBank, SwissProt, or LocusLink is quicker but can suffer from the ad hoc relationship of keywords to the research interests of whoever submitted the entry. Moreover, keeping annotation fields current as new

- *Y. Liu, S.B. Navathe, J. Civera, and A. Ram are with the College of Computing, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30322. E-mail: {yingliu, sham, ashwin}@cc.gatech.edu, jorcisai@iti.upv.es.*
- *V. Dasigi is with the Department of Computer Science, School of Computing and Software Engineering, Southern Polytechnic State University, Marietta, GA 30060. E-mail: vdasigi@spsu.edu.*
- *B.J. Ciliax is with the Department of Neurology, Emory University School of Medicine, Atlanta, GA 30322. E-mail: bciliax@emory.edu.*
- *R. Dingledine is with the Department of Pharmacology, Emory University School of Medicine, Atlanta, GA 30322. E-mail: rdingledine@pharm.emory.edu.*

Manuscript received 4 Apr. 2004; revised 1 Oct. 2004; accepted 10 Feb. 2005; published online 30 Mar. 2005.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-0043-0404.

information appears in the literature is a major challenge that is rarely met adequately.

If, instead of organizing by expression pattern similarity, genes were grouped according to shared function, investigators might more quickly discover patterns or themes of biological processes that were revealed by their microarray experiments and focus on a select group of functionally related genes. A number of clustering strategies based on shared functions rather than similar expression patterns have been devised. Chaussabel and Sher [3] analyzed literature profiles generated by extracting the frequencies of certain terms from the abstracts in MEDLINE and then clustered the genes based on these terms, essentially applying the same algorithm used for expression pattern clustering. Jenssen et al. [12] used co-occurrence of gene names in abstracts to create networks of related genes automatically. Text analysis of biomedical literature has also been applied successfully to incorporate functional information about the genes in the analysis of gene expression data [1], [10], [13], [14] without generating clusters de novo. For example, Blaschke et al. [1] extracted information about the common biological characteristics of gene clusters from MEDLINE using Andrade and Valencia's statistical text mining approach, which accepts user-supplied abstracts related to a protein of interest and returns an ordered set of keywords that occur in those abstracts more often than would be expected by chance [15].

We expanded and extended Andrade and Valencia's approach [15] to functional gene clustering by using an approach that applies an algorithm called the Bond Energy Algorithm (BEA) [16], [17], which, to our knowledge, has not been used in bioinformatics. We modified it so that the "affinity" among attributes (in our case, genes) is defined based on the sharing of keywords between them and we came up with a scheme for partitioning the clustered affinity matrix to produce clusters of genes. We call the resulting algorithm BEA-PARTITION. BEA was originally conceived as a technique to cluster questions in psychological instruments [16], has been used in operations research, production engineering, marketing, and various other fields [18], and is a popular clustering algorithm in distributed database system (DDBS) design. The fundamental task of BEA in DDBS design is to group attributes based on their affinity, which indicates how closely related the attributes are, as determined by the inclusion of these attributes by the same database transactions. In our case, each gene was considered as an attribute. Hence, the basic premise is that two genes would have higher affinity, thus higher bond energy, if abstracts mentioning these genes shared many informative keywords. BEA has several useful properties [16], [19]. First, it groups attributes with larger affinity values together, and the ones with smaller values together (i.e., during the permutation of columns and rows, it shuffles the attributes towards those with which they have higher affinity and away from those with which they have lower affinity). Second, the composition and order of the final groups are insensitive to the order in which items are presented to the algorithm. Finally, it seeks to uncover and display the association and interrelationships of the clustered groups with one another.

In order to explore whether this algorithm could be useful for clustering genes derived from microarray experiments, we compared the performance of BEA-PARTITION, hierarchical clustering algorithm, self-organizing map, and the k-means algorithm for clustering functionally-related genes based on shared keywords, using purity, entropy, and mutual information as metrics for evaluating cluster quality.

2 Methods

2.1 Keyword Extraction from Biomedical Literature

We used statistical methods to extract keywords from MEDLINE citations, based on the work of [15]. This method estimates the significance of words by comparing the frequency of words in a given gene-related set (Test Set) of abstracts with their frequency in a background set of abstracts. We modified the original method by using a 1) different background set, 2) a different stemming algorithm (Porter's stemmer), and 3) a customized stop list. The details were reported by Liu et al. [20], [21].

For each gene analyzed, word frequencies were calculated from a group of abstracts retrieved by an SQL (structured query language) search of MEDLINE for the specific gene name, gene symbol, or any known aliases (see LocusLink, ftp://ftp.ncbi.nih.gov/refseq/LocusLink/LL_tmpl.gz for gene aliases) in the TITLE field. The resulting set of abstracts (the Test Set) was processed to generate a specific keyword list.

Test Sets of Genes. We compared BEA-PARTITION and other clustering algorithms (k-means, hierarchical, and SOM) on two test sets.

1. Twenty-six genes in four well-defined functional groups consisting of 10 glutamate receptor subunits, seven enzymes in catecholamine metabolism, five cytoskeletal proteins, and four enzymes in tyrosine and phenylalanine synthesis. The gene names and aliases are listed in Table 1. This experiment was performed to determine whether keyword associations can be used to group genes appropriately and whether the four gene families or clusters that were known a priori would also be predicted by a clustering algorithm simply using the affinity metric based on keywords.
2. Forty-four yeast genes involved in the cell cycle of budding yeast (*Saccharomyces cerevisiae*) that had altered expression patterns on spotted DNA microarrays [6]. These genes were analyzed by Cherepinsky et al. [4] to demonstrate their Shrinkage algorithm for gene clustering. A master list of member genes for each cluster was assembled according to a combination of 1) common cell-cycle functions and regulatory systems and 2) the corresponding transcriptional activators for each gene [4] (Table 2).

Keyword Assessment. Statistical formulae from [15] for word frequencies were used without modification. These calculations were repeated for all gene names in the test

TABLE 1
Twenty-Six Genes Manually Clustered Based on Functional Similarity

Group	Genes	Functions
1	*GluR1,GluR2,GluR3,GluR4,GluR6, KA1,KA2,NMDA-R1,NMDA-R2A, NMDA-R2B*	Glutamate receptor channels
2	*Tyrosine hydroxylase, DOPA decarboxylase, Dopamine beta-hydroxylase, Phenethanolamine N-methyltransferase, Monoamine oxidase A, Monoamine oxidase B, Catechol-O-methyltransferase*	Catecholamine synthetic enzymes
3	*Actin, Alpha-tubulin, Beta-tubulin, Alpha-spectrin, Dynein*	Cytoskeletal proteins
4	*Chorismate mutase, Prephenate dehydratase, Prephenate dehydrogenase, Tyrosine transaminase*	Enzymes in tyrosine and phenylalanine synthesis

TABLE 2
Forty-Four Yeast Genes Grouped by Transcriptional Activators and Cell Cycle Functions [4]

Group	Activators	Genes	Functions
1	Swi4, Swi6	*Cln1, Cln2, Gic1, Gic2, Msb2, Rsr1, Bud9, Mnn1, Och1, Exg1, Kre6, Cwp1*	Budding
2	Swi6, Mbp1	*Clb5, Clb6, Rnr1, Rad27, Cdc21, Dun1, Rad51, Cdc45, Mcm2*	DNA replication and repair
3	Swi4, Swi6	*Htb1, Htb2, Hta1, Hta2, Hta3, Hho1*	Chromatin
4	Fkh1	*Hhf1, Hht1, Tel2, Apr7*	Chromatin
5	Fkh1	*Tem1*	Mitosis control
6	Ndd1, Fkh2, Mcm1	*Clb2, Ace2, Swi5, Cdc20*	Mitosis control
7	Ace2, Swi5	*Cts1, Egt2*	Cytokinesis
8	Mcm1	*Mcm3, Mcm6, Cdc6, Cdc46*	Prereplication complex formation
9	Mcm1	*Ste2, Far1*	Mating

set, a process that generated a database of keywords associated with specific genes, the strength of the association being reflected by a z-score. The z-score of word a for gene g is defined as:

$$Z_g^a = \frac{F_g^a - \overline{F}^a}{\sigma^a}, \quad (1)$$

where F_g^a equals the frequency of word a in Test Set g (i.e., in the Test set g, the number of abstracts where the word a occurs divided by the total number of abstracts) and, $\bar{F}^a$ and σ^a are the average frequency and standard deviation, respectively, of word a in the background set. Intuitively, the score Z compares the "importance" or "discriminatory relevance" of a keyword in the test set of abstract with the background set that represents the expected occurrence of that word in the literature at large.

Keyword Selection for Gene Clustering. We used z-score thresholds to select the keywords used for gene clustering. Those keywords with z-scores less than the threshold were discarded. The z-score thresholds we tested were 0, 5, 8, 10, 15, 20, 30, 50, and 100. The database generated by this algorithm is represented as a sparse word (rows) $\times$ gene (columns) matrix with cells containing z-scores. The matrix is characterized as "sparse" because each gene only has a fraction of all words associated with it. The output of the keyword selection for all genes in each Test Set is represented as a sparse keyword (rows) $\times$ gene (columns) matrix with cells containing z-scores.

2.2 BEA-PARTITION: Detailed Working of the Algorithm

The BEA-PARTITION takes a symmetric matrix as input, permutes its rows and columns, and generates a sorted matrix, which is then partitioned to form a clustered matrix.

Constructing the Symmetric Gene $\times$ Gene Matrix. The sparse word $\times$ gene matrix, with the cells containing the z-scores of each word-gene pair, was converted to a gene $\times$ gene matrix with the cells containing the sum of products of z-scores for shared keywords. The z-score value was set to zero if the value was less than the threshold. Larger values reflect stronger and more extensive keyword associations between gene-gene pairs. For each gene pair (Gi, Gj) and every word a they share in the sparse word $\times$ gene matrix, the $Gi \times$ Gj cell value $(aff(Gi, Gj))$ in the gene $\times$ gene matrix represents the affinity of the two genes for each other and is calculated as:

$$aff(G_i, G_j) = \frac{\sum_{a=1}^{N} (Z_{Gi}^a \times Z_{Gj}^a)}{1,000}. \quad (2)$$

Dividing the sum of the z-score product by 1,000 was done to reduce the typically large numbers to a more readable format in the output matrix.

Sorting the Matrix [19]. The sorted matrix is generated as follows:

1. *Initialization*. Place and fix one of the columns of symmetric matrix arbitrarily into the clustered matrix.
2. *Iteration*. Pick one of the remaining n-i columns (where i is the number of columns already in the sorted matrix). Choose the placement in the sorted matrix that maximizes the change in bond energy as described below (3). Repeat this step until no more columns remain.
3. *Row ordering*. Once the column ordering is determined, the placement of the rows should also be changed correspondingly so that their relative positions match the relative position of the columns. This restores the symmetry to the sorted matrix.

To calculate the change in bond energy for each possible placement of the next $(i+1)$ column, the bonds between that column (k) and each of two newly adjacent columns (i, j) are added and the bond that would be broken between the latter two columns is subtracted. Thus, the "bond energy" between these three columns i, j, and k (representing gene i (Gi); gene j (Gj); gene k (Gk))) is calculated by the following interaction contribution measure:

$$\begin{aligned} &energy(Gi, Gj, Gk) = \\ &2 \times [bond(Gi, Gk) + bond(Gk, Gj) - bond(Gi, Gj)], \end{aligned} \tag{3}$$

where bond (Gi, Gj) is the bond energy between gene Gi and gene Gj and

$$bond(Gi, Gj) = \sum_{r=l}^{N} aff(Gr, Gi) \times aff(Gr, Gj) \tag{4}$$

$$\begin{aligned} aff(G0, Gi) &= aff(Gi, G0) \\ &= aff(G(n+1), Gi) = aff(Gi, G(n+1)) = 0. \end{aligned} \tag{5}$$

The last set of conditions (5) takes care of cases where a gene is being placed in the sorted matrix to the left of the leftmost gene or to the right of the rightmost gene during column permutations, and prior to the topmost row and following the last row during row permutations.

Partitioning the Sorted Matrix. The original BEA algorithm [16] did not propose how to partition the sorted matrix. The partitioning heuristic was added by Navathe et al. [17] for the problems in the distributed database design. These heuristics were constructed using the goals of design: to minimize access time and storage costs. We do not have the luxury of such a clear cut objective function in our case. Hence, to partition the sorted matrix into submatrices, each representing a gene cluster, we experimented with different heuristics and, finally, derived a heuristic that identifies the boundaries between clusters by sequentially finding the maximum sum of the quotients for corresponding cells in adjacent columns across the matrix. With each successive split, only those rows corresponding to the remaining columns were processed, i.e., only the remaining symmetrical portion of the submatrix was used for further iterations of the splitting algorithm. The number of clusters into which the gene affinity matrix was partitioned was determined by AUTOCLASS (described below), however, other heuristics might be useful for this determination. The boundary metric (B) for columns Gi and Gj used for placement of new column k between existing columns i and j was defined as:

$$B(G_i, G_j) = \max_{p-1 \le q \le p} \sum_{k=p-1}^{p} \frac{\max(aff(k,q), aff(k,q+1))}{\min(aff(k,q), aff(k,q+1))}, \tag{6}$$

where q is the new splitting point (for simplicity, we use the number of the leftmost column in the new submatrix that is to the right of the splitting point), which will split the submatrix defined between two previous splitting points, p and $p-1$ (which do not necessarily represent contiguous columns). To partition the entire sorted matrix, the following initial conditions are set, $p = \mathrm{N}, p-1 = 0$.

2.3 K-Means Algorithm and Hierarchical Clustering Algorithm

K-means and hierarchical clustering analysis were performed using Cluster/Treeview programs available online (http://bonsai.ims.u-tokyo.ac.jp/~mdehoon/software/cluster/software.htm).

2.4 Self-Organizing Map

Self-organizing map was performed using GeneCluster 2.0 (http://www.broad.mit.edu/cancer/software/software.html).

Euclidean distance measure was used when gene × keyword matrix as input. When gene × gene matrix was used as input, the gene similarity was calculated by (2).

2.5 Number of Clusters

In order to apply BEA-PARTITION and k-means clustering algorithms, the investigator needs to have a priori knowledge about the number of clusters in the test set. We determined the number of clusters by applying AUTOCLASS, an unsupervised Bayesian classification system developed by [22]. AUTOCLASS, which seeks a maximum posterior probability classification, determines the optimal number of classes in large data sets. Among a variety of applications, AUTOCLASS has been used for the discovery of new classes of infra-red stars in the IRAS Low Resolution Spectral catalogue, new classes of airports in a database of all US airports, and discovery of classes of proteins, introns and other patterns in DNA/protein sequence data [22]. We applied an open source implementation of AUTOCLASS (http://ic.arc.nasa.gov/ic/projects/bayes-group/autoclass/autoclass-c-program.html). The resulting number of clusters was then used as the endpoint for the partitioning step of the BEA-PARTITION algorithm. To determine whether AUTOCLASS could discover the number of clusters in the test sets correctly, we also tested different number of clusters other than the ones AUTOCLASS predicted.

2.6 Evaluating the Clustering Results

To evaluate the quality of our resultant clusters, we used the established metrics of Purity, Entropy, and Mutual Information, which are briefly described below [23]. Let us assume that we have C classes (i.e., C expert clusters, as shown in Tables 1 and 2), while our clustering algorithms produce K clusters, $\pi_1, \pi_2, \ldots, \pi_k$.

Purity. Purity can be interpreted as classification accuracy under the assumption that all objects of a cluster are classified to be members of the dominant class for that cluster. If the majority of genes in cluster A are in class X, then class X is the dominant class. Purity is defined as the ratio between the number of items in cluster π_{i} from dominant class j and the size of cluster π_{i}, that is:

$$P(\pi_i) = \frac{1}{n_i} \max_j (n_i^j), i = 1, 2 \ldots, k, \tag{7}$$

where $\mathrm{n_i} = |\pi_{\mathrm{i}}|$, that is, the size of cluster i and n_i^j is the number of genes in π_{i} that belong to class $\mathrm{j}, \mathrm{j} = 1, 2, \ldots, \mathrm{C}$. The closer to 1 the purity value is, the more similar this cluster is to its dominant class. Purity is measured for each cluster and the average purity of each test gene set cluster result was calculated.

Entropy. Entropy denotes how uniform the cluster is. If a cluster is composed of genes coming from different classes, then the value of entropy will be close to 1. If a cluster only contains one class, the value of entropy will be close to 0. The ideal value for entropy would be zero. Lower values of entropy would indicate better clustering. Entropy is also measured for each cluster and is defined as:

$$E(\pi_i) = -\frac{1}{\log C} \sum_{j=1}^{C} \frac{n_i^j}{n_i} \log\left(\frac{n_i^j}{n_i}\right). \tag{8}$$

The average entropy of each test gene set cluster result was also calculated.

Mutual Information. One problem with purity and entropy is that they are inherently biased to favor small clusters. For example, if we had one object for each cluster, then the value of purity would be 1 and entropy would be zero, no matter what the distribution of objects in the expert classes is.

Mutual information is a symmetric measure for the degree of dependency between clusters and classes. Unlike correlation, mutual information also takes higher order dependencies into account. We use mutual information because it captures how related clusters are to classes without bias towards small clusters. Mutual information is a measure of the discordance between the algorithm-derived clusters and the actual clusters. It is the measure of how much information the algorithm-derived clusters can tell us to infer the actual clusters. Random clustering has mutual information of 0 in the limit. Higher mutual information indicates higher similarity between the algorithm-derived clusters and the actual clusters. Mutual information is defined as:

$$M(\pi) = \frac{2}{N} \sum_{i=1}^{K} \sum_{j=1}^{C} n_i^j \frac{\log \frac{n_i^j \times N}{\sum_{t=1}^{K} n_i^t \sum_{t=1}^{C} n_i^t}}{\log(K \times C)}, \tag{9}$$

where N is the total number of genes being clustered and K is the number of clusters the algorithm produced, and C is the number of expert classes.

2.7 Top-Scoring Keywords Shared among Members of a Gene Cluster

Keywords were ranked according to their highest shared z-scores in each cluster. The keyword sharing strength metric (K^a) is defined as the sum of z-scores for a shared keyword a within the cluster, multiplied by the number of genes (M) within the cluster with which the word is associated; in this calculation z-scores less than a user-selected threshold are set to zero and are not counted.

$$K^a = \sum_{g=1}^{M} (z_g^a) \times \sum_{g=1}^{M} Count(z_g^a). \tag{10}$$

Thus, larger values reflect stronger and more extensive keyword associations within a cluster. We identified the 30 highest scoring keywords for each of the four clusters and provided these four lists to approximately 20 students, postdoctoral fellows, and faculty, asking them to guess a major function of the underlying genes that gave rise to the four keyword lists.

3 Results

3.1 Keywords and Keyword × Gene Matrix Generation

A list of keywords was generated for each gene to build the keyword × gene matrix. Keywords were sorted according to their z-scores. The keyword selection experiment (see below) showed that a z-score threshold of 10 generally produced better results, which suggests that keywords with z-scores lower than 10 have less information content, e.g., "cell," "express." The relative values of z-scores depended on the size of the background set (data not shown). Since we used 5.6 million abstracts as the background set, the z-scores of most of the informative keywords were well above 10 (based on smaller values of standard deviation in the definition of z-score). The keyword × gene matrices were used as inputs to k-means, hierarchical clustering algorithm, self-organizing map, while as required by the BEA approach, they were first converted to a gene × gene matrix based on common shared keywords and these gene × gene matrices were used as inputs to BEA-PARTITION. An overview of the gene clustering by shared keyword process is provided in Fig. 1.

3.2 Effect of Keyword Selection on Gene Clustering

The effect of using different z-score thresholds for keyword selection on the quality of resulting clusters is shown in Figs. 2A1 and 2B1. For both test sets, BEA-PARTITION produced clusters with higher mutual information when z-score thresholds were within a range of 10 to 20. For the 44-gene set, K-means produced clusters with the highest

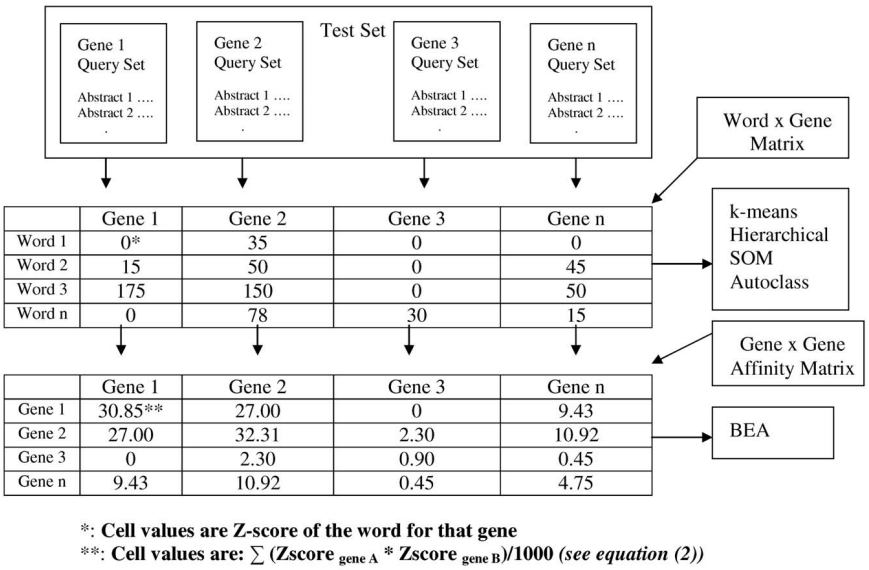


Fig. 1. Procedure for clustering genes by the strength of their associated keywords.

mutual information when the z-score threshold was 8, while, for the 26-gene set, mutual information was highest when z-score threshold was 15. For the remaining studies, we chose to use a z-score threshold of 10 to keep as many functional keywords as possible.

3.3 Number of Clusters

We then used AUTOCLASS to decide the number of clusters in the test sets. AUTOCLASS took the keyword × gene matrix as input and predicted that there were five clusters in the set of 26 genes and nine clusters in the set of 44 yeast genes. The effect of the numbers of clusters on the algorithm performance was shown in Figs. 2A2 and 2B2. BEA-PARTITION again produced a better result regardless of the number of clusters used. BEA-PARTITION had the highest mutual information when the numbers of clusters were five (26-gene set) and nine (44-gene set), whereas k-means worked marginally better when the numbers of clusters were 8 (26-gene set) and 10 (44-gene set). Based on these results we chose to use five and nine clusters, respectively, because the probabilities were higher than the other choices.

3.4 Clustering of the 26-Gene Set by Keyword Associations

To determine whether keyword associations could be used to group genes appropriately, we clustered the 26-gene set with either BEA-PARTITION, k-means, hierachical algorithm, SOM, and AUTOCLASS. Keyword lists were generated for each of these 26 genes, which belonged to one of four well-defined functional groups (Table 1). The resulting word × gene matrix had 26 columns (genes) and approximately 8,540 rows (words with z-scores $>= 10$ appearing in any of the query sets). The BEA-PARTITION, with z-score threshold = 10, correctly assigned 25 of 26 genes to the appropriate cluster based on the strength of keyword associations (Fig. 3). Tyrosine transaminase was the only outlier. As expected from the BEA-PARTITION, cells inside clusters tended to have much higher values than those outside. Hierarchical clustering algorithm, with the gene × keyword matrix as the input, generated similar result as BEA-PARTITION (five clusters and TT was the outlier) (Fig. 4a). The results, with gene × gene matrix as the input, were shown in tables in the supplementary materials which can be found at www.computer.org/publications/dlib.

While BEA-PARTITION and hierarchical clustering algorithm produced clusters very similar to the original functional classes, those produced by k-means (Table 4), self-organizing map (Table 5), and AUTOCLASS (Table 6), with gene × keyword matrix as input, were heterogeneous and, thus, more difficult to explain. The average purity,

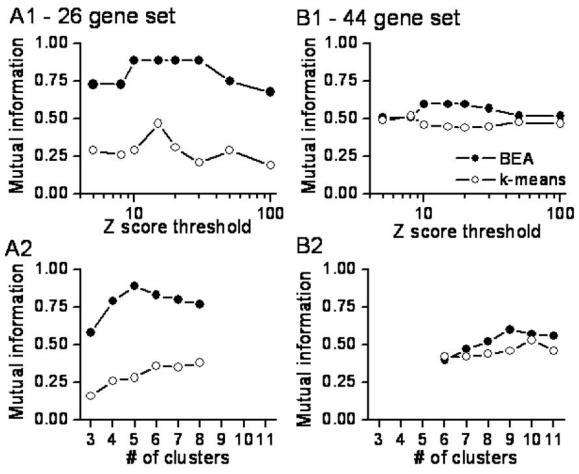
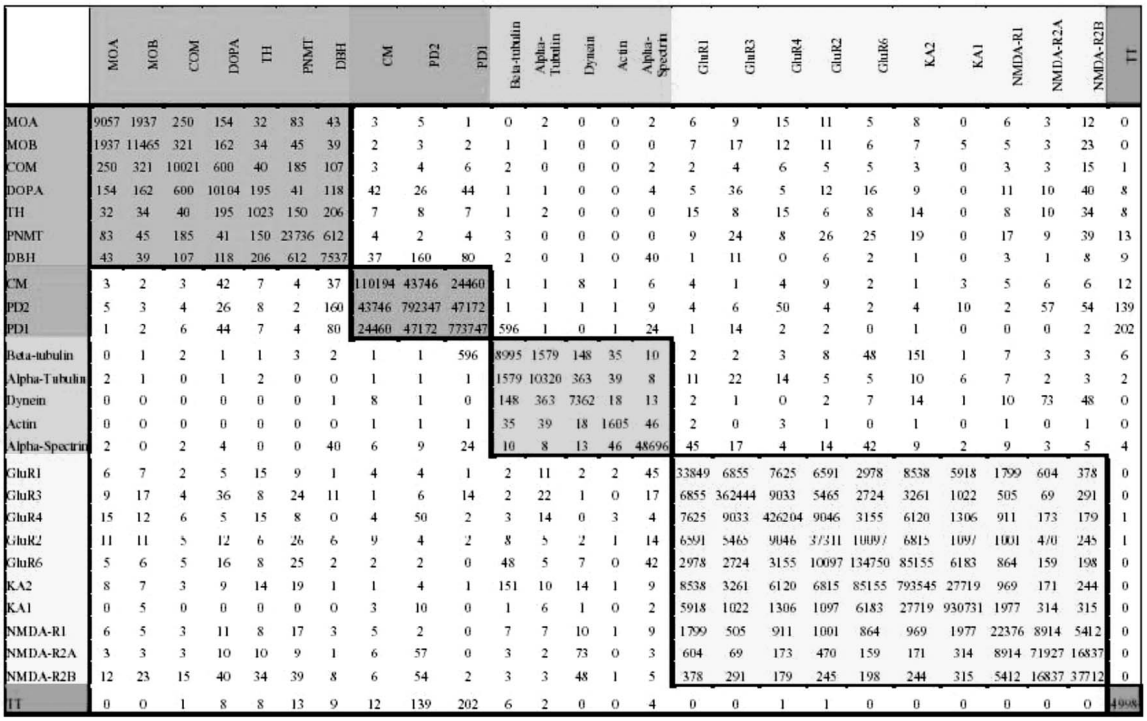


Fig. 2. Effect of keyword selection by z-score thresholds (A1 and B1) and different number of clusters (A2 and B2) on the cluster quality. Z-score thresholds were used to select the keywords for gene clustering. Those keywords with z-scores less than the threshold were discarded. To determine the effect of keyword selection by z-score thresholds on cluster quality, we tested z-score thresholds 0, 5, 8, 10, 15, 20, 30, 50, and 100. To determine whether AUTOCLASS could be used to discover the number of clusters in the test sets correctly, we tested a different number of clusters other than the ones AUTOCLASS predicted (four for the 26-gene set and nine for the 44-gene set).



	MOA	MOB	COM	DOPA	TH	PNMT	DBH	CM	PD2	PD1	Beta-tubulin	Alpha-Tubulin	Dynein	Actin	Alpha-Spectrin	GluR1	GluR3	GluR4	GluR2	GluR6	KA2	KA1	NMDA-R1	NMDA-R2A	NMDA-R2B	TT
MOA	9057	1937	250	154	32	83	43	3	5	1	0	2	0	0	2	6	9	15	11	5	8	0	6	3	12	0
MOB	1937	11465	321	162	34	45	39	2	3	2	1	1	0	0	0	7	17	12	11	6	7	5	5	3	23	0
COM	250	321	10021	600	40	185	107	3	4	6	2	0	0	0	2	2	4	6	5	5	3	0	3	3	15	1
DOPA	154	162	600	10104	195	41	118	42	26	44	1	1	0	0	4	5	36	5	12	16	9	0	11	10	40	8
TH	32	34	40	195	1023	150	206	7	8	7	1	2	0	0	0	15	8	15	6	8	14	0	8	10	34	8
PNMT	83	45	185	41	150	23736	612	4	2	4	3	0	0	0	0	9	24	8	26	25	19	0	17	9	39	13
DBH	43	39	107	118	206	612	7537	37	160	80	2	0	1	0	40	1	11	0	6	2	1	0	3	1	8	9
CM	3	2	3	42	7	4	37	110194	43746	24460	1	1	8	1	6	4	1	4	9	2	1	3	5	6	6	12
PD2	5	3	4	26	8	2	160	43746	792347	47172	1	1	1	1	9	4	6	50	4	2	4	10	2	57	54	139
PD1	1	2	6	44	7	4	80	24460	47172	773747	596	1	0	1	24	1	14	2	2	0	1	0	0	0	2	202
Beta-tubulin	0	1	2	1	1	3	2	1	1	596	8995	1579	148	35	10	2	2	3	8	48	151	1	7	3	3	6
Alpha-Tubulin	2	1	0	1	2	0	0	1	1	1	1579	10320	363	39	8	11	22	14	5	5	10	6	7	2	3	2
Dynein	0	0	0	0	0	0	1	8	1	0	148	363	7362	18	13	2	1	0	2	7	14	1	10	73	48	0
Actin	0	0	0	0	0	0	0	1	1	1	35	39	18	1605	46	2	0	3	1	0	1	0	1	0	1	0
Alpha-Spectrin	2	0	2	4	0	0	40	6	9	24	10	8	13	46	48696	45	17	4	14	42	9	2	9	3	5	4
GluR1	6	7	2	5	15	9	1	4	4	1	2	11	2	2	45	33849	6855	7625	6591	2978	8538	5918	1799	604	378	0
GluR3	9	17	4	36	8	24	11	1	6	14	2	22	1	0	17	6855	362444	9033	5465	2724	3261	1022	505	69	291	0
GluR4	15	12	6	5	15	8	0	4	50	2	3	14	0	3	4	7625	9033	426204	9046	3155	6120	1306	911	173	179	1
GluR2	11	11	5	12	6	26	6	9	4	2	8	5	2	1	14	6591	5465	9046	37311	10097	6815	1097	1001	470	245	1
GluR6	5	6	5	16	8	25	2	2	2	0	48	5	7	0	42	2978	2724	3155	10097	134750	85155	6183	864	159	198	0
KA2	8	7	3	9	14	19	1	1	4	1	151	10	14	1	9	8538	3261	6120	6815	85155	793545	27719	969	171	244	0
KA1	0	5	0	0	0	0	0	3	10	0	1	6	1	0	2	5918	1022	1306	1097	6183	27719	930731	1977	314	315	0
NMDA-R1	6	5	3	11	8	17	3	5	2	0	7	7	10	1	9	1799	505	911	1001	864	969	1977	22376	8914	5412	0
NMDA-R2A	3	3	3	10	10	9	1	6	57	0	3	2	73	0	3	604	69	173	470	159	171	314	8914	71927	16837	0
NMDA-R2B	12	23	15	40	34	39	8	6	54	2	3	3	48	1	5	378	291	179	245	198	244	315	5412	16837	37712	0
TT	0	0	1	8	8	13	9	12	139	202	6	2	0	0	4	0	0	1	1	0	0	0	0	0	0	4998

Fig. 3. Gene clusters by keyword associations using BEA-PARTITION. Keywords with z-scores $>= 10$ were extracted from MEDLINE abstracts for 26 genes in four functional classes. The resulting word $\times$ gene sparse matrix was converted to a gene $\times$ gene matrix. The cell values are the sum of z-score products for all keywords shared by the gene pair. This value is divided by 1,000 for purpose of display. A modified bond energy algorithm [16], [17] was used to group genes into five clusters based on the strength of keyword associations, and the resulting gene clusters are boxed.

average entropy, and mutual information of the BEA-PARTITION and hierarchical algorithm result were 1, 0, and 0.88, while those of k-means result were 0.53, 0.65, and 0.28, respectively, those of SOM result were 0.76, 0.35, and 0.18, respectively, and those of AUTOCLASS result were 0.82, 0.28, and 0.56 (Table 3) (gene $\times$ keyword matrix as input). When gene $\times$ gene matrix was used as input to hierarchical algorithm, k-means, and SOM, the results were even worse as measured by purity, entropy, and mutual information (Table 3).

3.5 Yeast Microarray Gene Clustering by Keyword Association

To determine whether our test mining/gene clustering approach could be used to group genes identified in microarray experiments, we clustered 44 yeast genes taken from Eisen et al. [6] via Cherepinsky et al. [4], again using BEA-PARTITION, hierarchical algorithm, SOM, AUTOCLASS, and k-means. Keyword lists were generated for each of the 44 yeast genes (Table 2) and a 3,882 (words appearing in the query sets with z-score greater or equal 10) $\times$ 44 (genes) matrix was created. The clusters produced by the BEA-PARTITION, k-means, SOM, and AUTOCLASS are shown in Tables 7, 8, 9, and 10, respectively, whereas those produced by hierarchical algorithm are shown in Fig. 4b. The average purity, average entropy, and mutual information of the BEA-PARTITION result were 0.74, 0.24, and 0.60, whereas those of hierarchical algorithm, SOM, k-means, and AUTOCLASS results (gene $\times$ keyword matrix as input) were 0.86, 0.12, and 0.58; 0.60, 0.37, and 0.46; 0.61, 0.33, and 0.39; 0.57, 0.39, and 0.49, respectively (Table 3).

3.6 Keywords Indicative of Major Shared Functions with a Gene Cluster

Keywords shared among genes (26-gene set) within each cluster were ranked according to a metric based on both the degree of significance (the sum of z-scores for each keyword) and the breadth of distribution (the sum of the number of genes within the cluster for which the keyword has a z-score greater than a selected threshold). This double-pronged metric obviated the difficulty encountered with keywords that had extremely high z-scores for single genes within the cluster but modest z-scores for the remainder. The 30 highest scoring keywords for each of the four clusters were tabulated (Table 11). The respective keyword lists appeared to be highly informative about the general function of the original, preselected clusters when shown to medical students, faculties, and postdoctoral fellows.

4 Discussion

In this paper, we clustered the genes by shared functional keywords. Our gene clustering strategy is similar to the document clustering in information retrieval. Document clustering, defined as grouping documents into clusters according to their topics or main contents in an unsupervised manner, organizes large amounts of information into a small number of meaningful clusters and improves the information retrieval performance either via cluster-driven dimensionality reduction, term-weighting, or query expansion [9], [24], [25], [26], [27].

Term vector-based document clustering has been widely studied in information retrieval [9], [24], [25], [26], [27]. A

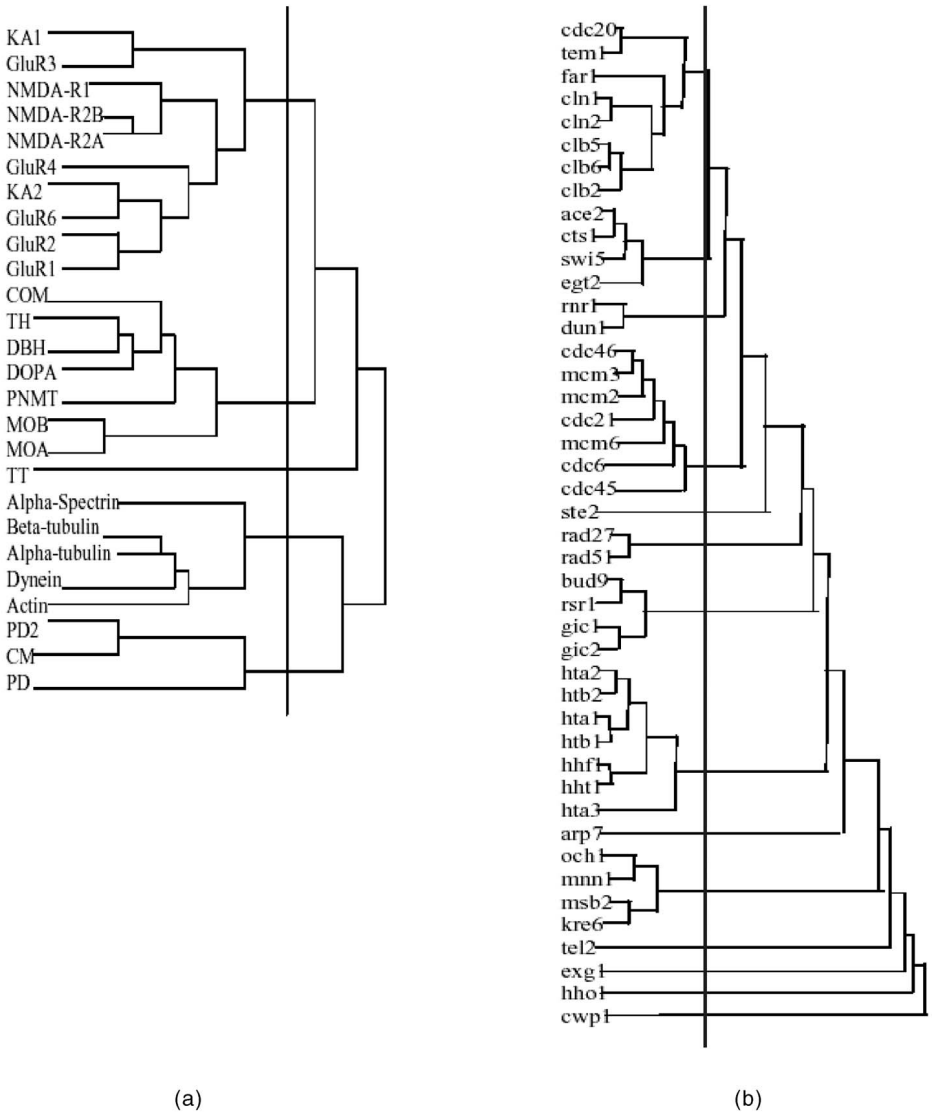


Fig. 4. Gene clusters by keyword associations using hierarchical clustering algorithm. Keywords with z-scores $>= 10$ were extracted from MEDLINE abstracts for (a) 26 genes in four functional classes and (b) 44 gene in nine classes. The resulting word $\times$ gene sparse matrix was used as input to the hierarchical algorithm.

number of clustering algorithms have been proposed and many of them have been applied to bioinformatics research. In this report, we introduced a new algorithm for clustering genes, BEA-PARTITION. Our results showed that BEA-PARTITION, in conjunction with the heuristic developed for partitioning the sorted matrix, outperforms the k-means algorithm and SOM in two test sets. In the first set of genes (26-gene set), BEA-PARTITION, as well as hierarchical algorithm, correctly assigned 25 of 26 genes in a test set of four known gene groups with one outlier, whereas k-means and SOM mixed the genes into five more evenly sized but less well functionally defined groups. In the 44-gene set, the result generated by BEA-PARTITION had the highest mutual information, indicating that BEA-PARTITION outperformed all the other four clustering algorithms.

4.1 BEA-PARTITION versus k-Means

In this study, the z-score thresholds were used for keyword selection. When the threshold was 0, all words, including noise (noninformative words and misspelled words), were used to cluster genes. Under the tested conditions, clusters produced by BEA-PARTITION had higher quality than those produced by k-means. BEA-PARTITION clusters genes based on their shared keywords. It is unlikely that genes within the same cluster shared the same noisy words with high z-scores, indicating that BEA-PARTITION is less sensitive to noise than k-means. In fact, BEA-PARTITION performed better than k-means in the two test gene sets under almost all test conditions (Fig. 2). BEA-PARTITION performed best when z-score thresholds were 10, 15, and 20, which indicated 1) that the words with z-score less than 10 were less informative and 2) few words with z-scores between 10 and 20 were shared by at least two genes and did not improve the cluster quality. When z-score thresholds were high (> 30 in the 26-gene set and > 20 in the 44-gene set), more informative words were discarded, and as a result, the cluster quality was degraded.

TABLE 3
The Quality of the Gene Clusters Derived by Different Clustering Algorithms, Measured by Purity, Entropy, and Mutual Information

Input Matrix	Test gene set	Clustering algorithm	Average Purity	Average Entropy	Mutual Information
Gene X keyword matrix	26-gene set	Hierarchical	1	0	0.88
		k-means	0.53	0.65	0.28
		SOM	0.76	0.35	0.18
		Autoclass	0.82	0.28	0.56
	44-gene set	Hierarchical	0.86	0.12	0.58
		k-means	0.60	0.37	0.46
		SOM	0.61	0.33	0.39
		Autoclass	0.57	0.39	0.49
Gene X Gene matrix	26-gene set	BEA-PARTITION	1	0	0.88
		Hierarchical	1	0	0.88
		k-means	0.87	0.19	0.16
		SOM	0.81	0.28	0.20
		Autoclass	0.89	0.13	0.78
	44-gene set	BEA-PARTITION	0.74	0.24	0.60
		Hierarchical	0.84	0.16	0.56
		k-means	0.84	0.12	0.30
		SOM	0.71	0.27	0.35
		Autoclass	0.72	0.26	0.51

BEA-PARTITION is designed to group cells with larger values together, and the ones with smaller values together. The final order of the genes within the cluster reflected deeper interrelationships. Among the 10 glutamate receptor genes examined, *GluR1*, *GluR2*, and *GluR4* are AMPA receptors, while *GluR6*, *KA1*, and *KA2* are kainate receptors. The observation that BEA-PARTITION placed gene *GluR6* and gene *KA2* next to each other, confirms that the literature associations between *GluR6* and *KA2* are higher than those between *GluR6* and AMPA receptors. Furthermore, the association and interrelationships of the clustered groups with one another can be seen in the final clustering matrix. For example, TT was an outlier in Fig. 3, however, it still had higher affinity to *PD1* (affinity = 202) and *PD2* (affinity = 139) than to any other genes. Thus, TT appears to be strongly related to genes in the tyrosine and phenylalanine synthesis cluster, from which it originated.

BEA-PARTITION has several advantages over the k-means algorithm: 1) while k-means generally produces a locally optimal clustering [2], BEA-PARTITION produces

TABLE 4
Twenty-Six Gene Set k-Means Result (Gene $\times$ Keyword Matrix as Input)

Cluster	Gene	Function
1	*Dynein, Alpha-Tublin*	Cytoskeletal proteins
	MOB (Monoamine oxidase B),	Catecholamine synthetic enzymes
	MOA (Monoamine oxidase A)	
2	*GluR1,GluR2,GluR6,KA2,NMDA-R1*	Glutamate receptor channels
	PNMT (Phenethanolamine N-methyltransferase)	Catecholamine synthetic enzymes
3	*Actin,Beta-Tublin*	Cytoskeletal proteins
	DBH (Dopamine beta-hydroxylase),	Catecholamine synthetic enzymes
	DOPA (DOPA decarboxylase)	
	NMDA-R2B	Glutamate receptor channels
4	*COM (Catechol-O-methyltransferase)*	Catecholamine synthetic enzymes
	GluR3,GluR4,KA1	Glutamate receptor channels
	PD1 (Prephenate dehydratase),	Enzymes in tyrosine synthesis
	PD2 (Prephenate dehydrogenase)	
5	*Alpha-Spectrin*	Cytoskeletal proteins
	TH (Tyrosine hydroxylase)	Catecholamine synthetic enzymes
	NMDA-R2A	Glutamate receptor channels
	CM (Chorismate mutase),	Enzymes in tyrosine synthesis
	TT (tyrosine transaminase)	

TABLE 5
Twenty-Six Gene SOM Result (Gene × Keyword Matrix as Input)

Cluster	Gene	Function
1	*Actin,Alpha-Spectrin,Alpha-Tubulin Beta-tubulin,Dynein*	Cytoskeletal proteins
	GluR1,GluR2,GluR3,NMDA-R1,NMDA-R2A,NMDA-R2B	Glutamate receptor channels
	DBH (Dopamine beta-hydroxylase), COM (Catechol-O-methyltransferase) DOPA (DOPA decarboxylase) MOB (Monoamine oxidase B), MOA (Monoamine oxidase A) TH (Tyrosine hydroxylase) PNMT (Phenethanolamine N-methyltransferase	Catecholamine synthetic enzymes
	TT (tyrosine transaminase) CM (Chorismate mutase)	Enzymes in tyrosine synthesis
2	*GluR6*	Glutamate receptor channels
3	*GluR4*	
	KA2	Glutamate receptor channels
4	*KA1*	Glutamate receptor channels
	PD2 (Prephenate dehydrogenase)	Enzymes in tyrosine synthesis
	PD1 (Prephenate dehydratase)	

the globally optimal clustering by permuting the columns and rows of the symmetric matrix; 2) the k-means algorithm is sensitive to initial seed selection and noise [9].

4.2 BEA-PARTITION versus Hierarchical Algorithm

Hierarchical clustering algorithm, as well as k-means, and Self-Organizing Maps, have been widely used in microarray expression profile analysis. Hierarchical clustering organizes expression data into a binary tree without providing clear indication of how the hierarchy should be clustered. In practice, investigators define clusters by a manual scan of the genes in each node and rely on their biological expertise to notice shared functional properties of genes. Therefore, the definition of the clusters is subjective, and as a result, different investigators may interpret the same clustering result differently. Some have proposed automatically defining boundaries based on statistical properties of the gene expression profiles; however, the same statistical criteria may not be generally applicable to identify all relevant biological functions [10]. We believe that an algorithm that produces clusters with clear boundaries can provide more objective results and possibly new discoveries, which are beyond the experts' knowledge. In this report, our results showed that BEA-PARTITION can have similar performance as a hierarchical algorithm, and provide distinct cluster boundaries.

4.3 K-Means versus SOM

The k-means algorithm and SOM can group objects into different clusters and provide clear boundaries. Despite its

TABLE 6
Twenty-Six Gene AUTOCLASS Result (Gene × Keyword Matrix as Input)

Cluster	Gene	Function
1	*Alpha-Spectrin*	Cytoskeletal proteins
	DBH (Dopamine beta-hydroxylase), DOPA (DOPA decarboxylase) TH (Tyrosine hydroxylase)	Catecholamine synthetic enzymes
	NMDA-R1	Glutamate receptor channels
2	*GluR2,GluR3,GluR4,GluR6,NMDA-R2A,NMDA-R2B*	Glutamate receptor channels
3	*GluR1,KA1,KA2*	Glutamate receptor channels
	PD2 (Prephenate dehydrogenase) PD1 (Prephenate dehydratase) TT (tyrosine transaminase)	Enzymes in tyrosine synthesis
	CM (Chorismate mutase) PNMT (Phenethanolamine N-methyltransferas)	Catecholamine synthetic enzymes
4	*Actin,Alpha-Tubulin,Beta-tubulin Dynein*	Cytoskeletal proteins
5	*MOB (Monoamine oxidase B), MOA (Monoamine oxidase A) COM (Catechol-O-methyltransferase)*	Catecholamine synthetic enzymes

TABLE 7
Forty-Four Yeast Genes BEA-PARTITION Result (Gene × Keyword Matrix as Input)

Clusters	Activators	Genes
1	Swi4, Swi6	*Cwp1, Exg1, Mnn1, Och1*
2	Fkh1	*Arp7*
3	Ndd1, Fkh2, Mcm1	*Cdc20, Swi5, Ace2, Clb2*
	Ace2, Swi5	*Egt2, Cts1*
4	Swi4, Swi6	*Bud9, Rsr1, Gic1, Gic2*
	Mcm1	*Far1*
	Fkh1	*Tem1*
5	Swi4, Swi6	*Cln1, Cln2*
	Swi6, Mbp1	*Clb5, Clb6, Rnr1, Dun1*
6	Swi4, Swi6	*Hta1, Hta3, Hta2, Htb2, Htb1*
	Fkh1	*Hhf1, Hht1*
	Swi6, Mbp1	*Rad51*
7	Swi4, Swi6	*Kre6, Msb2*
	Swi4, Swi6	*Hho1*
	Mcm1	*Ste2*
8	Fkh1	*Tel2*
9	Swi6, Mbp1	*Rad27, Cdc45, Mcm2, Cdc21*
	Mcm1	*Cdc46, Mcm3, Mcm6, Cdc6*

simplicity and efficiency, the SOM algorithm has several weaknesses that make its theoretical analysis difficult and limit its practical usefulness. Various studies have suggested that it is hard to find any criteria under which the SOM algorithm performs better than the traditional techniques, such as k-means [11]. Balakrishnan et al. [28] compared the SOM algorithm with k-means clustering on 108 multivariate normal clustering problems. The results showed that the SOM algorithm performed significantly worse than the k-means clustering algorithm. Our results also showed that k-means performed better than SOM by generating clusters with higher mutual information.

4.4 Computing Time

The computing time of BEA-PARTITION, same as that of hierarchical algorithm and SOM, is in the order of N^2, which means that it grows proportionally to the square of the number of genes and commonly denoted as $O(\mathrm{N}^2)$, and that of k-means is in the order of N*K*T (O(NKT)), where N is the number of genes tested, K is the number of clusters, and T is the number of improvement steps (iterations) performed by k-means. In our study, the number of improvement steps was 1,000. Therefore, when the number of genes tested is about 1,000, BEA-PARTITION runs $(a^*\mathrm{K} + \mathrm{b})$ times faster than k-means, where a, and b are constants. As long as the number of genes to be clustered is less than the product of the number

TABLE 8
Forty-Four Yeast Gene SOM Result (Gene × Keyword as Input)

Clusters	Activators	Genes
1	Swi4, Swi6	*Gic1,Gic2,Msb2*
2	Fkh1	*Hhf1,Hht1*
	Swi4, Swi6	*Hta2,Hta3,Htb2*
3	Swi4, Swi6	*Hta1,Htb1*
4	Ndd1, Fkh2, Mcm1	*Cdc20,Clb2,Cln1,Cln2,Cwp1,Exg1,Mnn1,Och1,Rsr1*
	Swi6, Mbp1	*Cdc21,Cdc45,Clb5,Clb6,Dun1,Mcm2,Rad27,Rad51,Rnr1*
	Mcm1	*Cdc46,Cdc6,Mcm3,Mcm6*
	Mcm1	*Far1,Ste2*
	Fkh1	*Tem1*
5	Ndd1, Fkh2, Mcm1	*Ace2,Swi5*
	Ace2, Swi5	*Cts1*
	Swi4, Swi6	*Kre6*
6	Ace2, Swi5	*Egt2*
	Swi4, Swi6	*Hho1*
	Fkh1	*Tel2*
7	Fkh1	*Arp7*
	Swi4, Swi6	*Bud9*

TABLE 9
Forty-Four Yeast Gene k-Means Result (Gene $\times$ Keyword Matrix as Input)

Clusters	Activators	Genes
1	Ndd1, Fkh2, Mcm1	*Ace2, Swi5*
	Ace2, Swi5	*Cts1, Egt2*
	Swi6, Mbp1	*Rad51*
	Fkh1	*Tel2*
2	Swi6, Mbp1	*Cdc21, Cdc45, Mcm2*
	Mcm1	*Cdc46, Mcm3, Mcm6*
	Mcm1	*Ste2*
3	Swi4, Swi6	*Hho1, Hta3*
4	Swi4, Swi6	*Gic1, Gic2*
	Swi6, Mbp1	*Rad27*
5	Swi4, Swi6	*Bud9, Mnn1, Rsr1*
	Swi6, Mbp1	*Rnr1*
6	Swi4, Swi6	*Exg1, Kre6, Och1,*
	Fkh1	*Tem1*
7	Fkh1	*Arp7*
	Swi4, Swi6	*Cwp1, Msb2*
8	Swi6, Mbp1	*Dun1,*
	Fkh1	*Hhf1, Hht1*
	Swi4, Swi6	*Hta1, Hta2, Htb1, Htb2*
9	Ndd1, Fkh2, Mcm1	*Cdc20, Clb2*
	Mcm1	*Cdc6*
	Swi6, Mbp1	*Clb5, Clb6*
	Swi4, Swi6	*Cln1, Cln2*
	Mcm1	*Far1*

of clusters and the number of iterations, BEA-PARTITION will run faster than k-means.

4.5 Number of Clusters

One disadvantage of BEA-PARTITION and k-means compared to hierarchical clustering is that the investigator needs to have a priori knowledge about the number of clusters in the test set, which may not be known. We approached this problem by using AUTOCLASS to predict the number of clusters in the test sets. BEA-PARTITION performed best when it grouped the genes into five clusters (26-gene set) and nine clusters (44-gene set), which were predicted by AUTOCLASS with higher probabilities. Therefore, AUTOCLASS

TABLE 10
Forty-Four Yeast Gene AUTOCLASS Result (Gene $\times$ Keyword Matrix as Input)

Clusters	Activators	Genes
1	Swi4, Swi6	*Cwp1, Exg1, Mnn1, Och1*
	Swi4, Swi6	*Hhf1, Hht1*
	Fkh1	*Hta1, Hta3, Hta2, Htb2, Htb1*
2	Fkh1	*Arp7*
	Swi4, Swi6	*Bud9, Msb2, Rsr1*
	Swi4,Swi6	*Hho1*
	Mcm1	*Mcm3*
3	Ndd1, Fkh2, Mcm1	*Cdc20, Clb2*
	Swi6,Mbp1	*Clb5,Clb6*
	Fkh1	*Tem1*
4	Ndd1, Fkh2, Mcm1	*Ace2,Swi5*
	Swi6,Mbp1	*Cdc21*
	Ace2, Swi5	*Cts1,Egt2*
5	Mcm1	*Cdc6,Mcm6*
	Swi6, Mbp1	*Rad27,Rad51,Mcm2*
6	Swi4,Swi6	*Exg1,Kre6,Mnn1,Och1*
	Mcm1	*Ste2*
7	Swi6,Mbp1	*Cdc45*
	Mcm1	*Cdc46*
	Swi4,Swi6	*Gic1,Gic2*
8	Swi6,Mbp1	*Dun1,Rnr1*
	Fkh1	*Tel2*
9	Swi4,Swi6	*Cln1,Cln2*
	Mcm1	*Far1*

TABLE 11
Top Ranking Keywords Associated with Each Gene Cluster

Cluster 1 (Catecholamine Biosynthesis)	Cluster 2 (Tyrosine/Phenylalanine Metabolism)	Cluster 3 (Cytoplasmic Proteins)	Cluster 4 (Glutamate Receptors)
mao	mutase	tubulin	ampa
clorgyline	monofunct	dynein	ionotrop
phenylethanolamine	dehydratase	spectrin	kainate
methyltransferase	bifunct	microtubule	glutam
monoamine	phenylalanine	axonem	isoxazole
hydroxylase	tyrosine	axoneme	subunit
deprenyl	phenylpyruv	chlamydomona	glutamaterg
catechol	herbicola	demembran	homomer
dopamine	fluorophenylalanine	flagellar	receptor
oxidase	tryptophan	flagella	methyl
chromaffin	erwinia	cytoskeleton	propion
selegiline	catalyt	isotype	hydroxi
dihydroxyphenyl	brevibacterium	cytoskelet	neuron
catecholamine	substrate	microtubular	domoate
tyrosine	enzyme	protofila	hippocampu
phenylethylamine	dehydrogenase	tetrahymena	gyru
adrenomedullari	decarboxyl	depolymer	hippocamp
dopa	biosynthet	subunit	synapt
tyramine	flavum	isoform	methylisoxazole
medulla	aromat	cilia	hek
pargyline	hcl	polymer	aspart
inhibitor	subtili	sequence	postsynapt
homovanill	ammonium	mutant	cerebellum
catecholaminerg	sulfate	tyrosin	cortex
adren	monom	diverg	isoxazolepropion
enzyme	molecular	kinesin	cyclothiazide
dihydroxyphenylalanine	arg	pvuii	ca
coeruleu	mutant	intron	heteromer
parkinson	nicotinamide	codon	bergmann
moclobemide	subunit	multigene	coloc
noradrenerg	tyr	encod	forebrain
mptp	effector	cytoplasm	purkinje
neuron	inhibitor	physarum	cerebellar

appears to be an effective tool to assist the BEA-PARTITION in gene clustering.

5 Conclusions and Future Work

There are several aspects of the BEA approach that we are currently exploring with more detailed studies. For example, although the BEA-PARTITION described here performs relatively well on small sets of genes, the larger gene lists expected from microarray experiments need to be tested. Furthermore, we derived a heuristic to partition the clustered affinity matrix into clusters. We anticipate that this heuristic, which is simply based on the sum of ratios of corresponding values from adjacent columns, will generally work regardless of the type of items being clustered. Generally, optimizing the heuristic to partition a sorted matrix after BEA-based clustering will be valuable. Finally, we are developing a Web-based tool that will include a text mining phase to identify functional keywords, and a gene clustering phase to cluster the genes based on the shared functional keywords. We believe that this tool should be useful for discovering novel relationships among sets of genes because it links genes by shared functional keywords rather than just reporting known interactions based on published reports. Thus, genes that never co-occur in the same publication could still be linked by their shared keywords.

The BEA approach has been applied successfully to other disciplines, such as operations research, production engineering, and marketing [18]. The BEA-PARTITION algorithm represents our extension to the BEA approach specifically for dealing with the problem of discovering functional similarity among genes based on functional keywords extracted from literature. We believe that this important clustering technique, which was originally proposed by [16] to cluster questions on psychological instruments and later introduced by [17] for clustering of data items in database design, has promise for application to other bioinformatics problems where starting matrices are available from experimental observations.

Acknowledgments

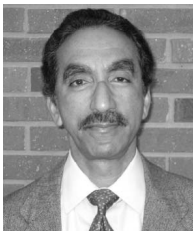
This work was supported by NINDS (RD) and the Emory-Georgia Tech Research Consortium. The authors would like to thank Brian Revennaugh and Alex Pivoshenk for research support.

References

[1] C. Blaschke, J.C. Oliveros, and A. Valencia, "Mining Functional Information Associated with Expression Arrays," *Functional & Integrative Genomics*, vol. 1, pp. 256-268, 2001.
[2] Y. Xu, V. Olman, and D. Xu, "EXCAVATOR: A Computer Program for Efficiently Mining Gene Expression Data," *Nucleic Acids Research*, vol. 31, pp. 5582-5589, 2003.
[3] D. Chaussabel and A. Sher, "Mining Microarray Expression Data by Literature Profiling," *Genome Biology*, vol. 3, pp. 1-16, 2002.
[4] V. Cherepinsky, J. Feng, M. Rejali, and B. Mishra, "Shrinkage-Based Similarity Metric for Cluster Analysis of Microarray Data," *Proc. Nat'l Academy of Sciences USA*, vol. 100, pp. 9668-9673, 2003.
[5] J. Quackenbush, "Computational Analysis of Microarray Data," *Nature Rev. Genetics*, vol. 2, pp. 418-427, 2001.

[6] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein, "Cluster Analysis and Display of Genome-Wide Expression Patterns," *Proc. Nat'l Academy of Sciences USA,* vol. 95, pp. 14863-14868, 1998.
[7] R. Herwig, A.J. Poustka, C. Mller, C. Bull, H. Lehrach, and J. O'Brien, "Large-Scale Clustering of cDNA-Fingerprinting Data," *Genome Research,* vol. 9, pp. 1093-1105, 1999.
[8] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E.S. Lander, and T.R. Golub, "Interpreting Patterns of Gene Expression with Self-Organizing Maps: Methods and Application to Hematopoietic Differentiation," *Proc. Nat'l Academy of Sciences USA,* vol. 96, pp. 2907-2912, 1999.
[9] A.K. Jain, M.N. Murty, and P.J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys,* vol. 31, pp. 264-323, 1999.
[10] S. Raychaudhuri, J.T. Chang, F. Imam, and R.B. Altman, "The Computational Analysis of Scientific Literature to Define and Recognize Gene Expression Clusters," *Nucleic Acids Research,* vol. 15, pp. 4553-4560, 2003.
[11] B. Kegl, "Principle Curves: Learning, Design, and Applications," PhD dissertation, Dept. of Computer Science, Concordia Univ., Montreal, Quebec, 2002.
[12] T.K. Jenssen, A. Laegreid, J. Komorowski, and E. Hovig, "A Literature Network of Human Genes for High-Throughtput Analysis of Gene Expression," *Nat'l Genetics,* vol. 178, pp. 139-143, 2001.
[13] D.R. Masys, J.B. Welsh, J.L. Fink, M. Gribskov, I. Klacansky, and J. Corbeil, "Use of Keyword Hierarchies to Interprate Gene Expression Patterns," *Bioinformatics,* vol. 17, pp. 319-326, 2001.
[14] S. Raychaudhuri, H. Schutze, and R.B. Altman, "Using Text Analysis to Identify Functionally Coherent Gene Groups," *Genome Research,* vol. 12, pp. 1582-1590, 2002.
[15] M. Andrade and A. Valencia, "Automatic Extraction of Keywords from Scientific Text: Application to the Knowledge Domain of Protein Families," *Bioinformatics,* vol. 14, pp. 600-607, 1998.
[16] W.T. McCormick, P.J. Schweitzer, and T.W. White, "Problem Decomposition and Data Reorganization by a Clustering Technique," *Operations Research,* vol. 20, pp. 993-1009, 1972.
[17] S. Navathe, S. Ceri, G. Wiederhold, and J. Dou, "Vertical Partitioning Algorithms for Database Design," *ACM Trans. Database Systems,* vol. 9, pp. 680-710, 1984.
[18] P. Arabie and L.J. Hubert, "The Bond Energy Algorithm Revisited," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 20, pp. 268-274, 1990.
[19] A.T. Ozsu and P. Valduriez, *Principles of Distributed Database Systems,* second ed. Prentice Hall Inc., 1999.
[20] Y. Liu, M. Brandon, S. Navathe, R. Dingledine, and B.J. Ciliax, "Text Mining Functional Keywords Associated with Genes," *Proc. Medinfo 2004,* pp. 292-296, Sept. 2004.
[21] Y. Liu, B.J. Ciliax, K. Borges, V. Dasigi, A. Ram, S. Navathe, and R. Dingledine, "Comparison of Two Schemes for Automatic Keyword Extraction from MEDLINE for Functional Gene Clustering," *Proc. IEEE Computational Systems Bioinformatics Conf. (CSB 2004),* pp. 394-404, Aug. 2004.
[22] P. Cheeseman and J. Stutz, "Bayesian Classification (Autoclass): Theory and Results," *Advances in Knowledge Discovery and Data Mining,* pp. 153-180, AAAI/MIT Press, 1996.
[23] A. Strehl, "Relationship-Based Clustering and Cluster Ensembles for High-Dimensional Data Mining," PhD dissertation, Dept. of Electric and Computer Eng., The University of Texas at Austin, 2002.
[24] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval.* New York: Addison Wesley Longman, 1999.
[25] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys,* vol. 34, pp. 1-47, 1999.
[26] P. Willett, "Recent Trends in Hierarchic Document Clustering: A Critical Review," *Information Processing and Management,* vol. 24, pp. 577-597, 1988.
[27] J. Aslam, A. Leblanc, and C. Stein, "Clustering Data without Prior Knowledge," *Proc. Algorithm Eng.: Fourth Int'l Workshop,* 1982.
[28] P.V. Balakrishnan, M.C. Cooper, V.S. Jacob, and P.A. Lewis, "A Study of the Classification Capabilities of Neural Networks Using Unsupervised Learning: A Comparison with K-Means Clustering," *Psychometrika,* vol. 59, pp. 509-525, 1994.

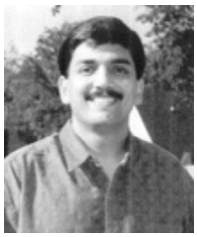
Ying Liu received the BS degree in environmental biology from Nanjing University, China. He received Master's degrees in bioinformatics and computer science from Georgia Institute of Technology in 2002. He is a PhD candidate in College of Computing, Georgia Institute of Technology, where he works on text mining biomedical literature to discover gene-to-gene relationships. His research interests include bioinformatics, computational biology, data mining, text mining, and database system. He is a student member of IEEE Computer Society.



Shamkant B. Navathe received the PhD degree from the University of Michigan in 1976. He is a professor in the College of Computing, Georgia Institute of Technology. He has published more than 130 refereed papers in database research; his important contributions are in database modeling, database conversion, database design, conceptual clustering, distributed database allocation, data mining, and database integration. Current projects include text mining of medical literature databases, creation of databases for biological applications, transaction models in P2P and Web applications, and data mining for better understanding of genomic/proteomic and medical data. His recent work has been focusing on issues of mobility, scalability, interoperability, and personalization of databases in scientific, engineering, and e-commerce applications. He is an author of the book, *Fundamentals of Database Systems,* with R. Elmasri (Addison Wesley, fourth edition, 2004) which is currently the leading database text-book worldwide. He also coauthored the book *Conceptual Design: An Entity Relationship Approach* (Addison Wesley, 1992) with Carlo Batini and Stefano Ceri. He was the general cochairman of the 1996 International VLDB (Very Large Data Base) Conference in Bombay, India. He was also program cochair of ACM SIGMOD 1985 at Austin, Texas. He is also on the editorial boards of *Data and Knowledge Engineering* (North Holland), *Information Systems* (Pergamon Press), *Distributed and Parallel Databases* (Kluwer Academic Publishers), and *World Wide Web Journal* (Kluwer). He has been an associate editor of *IEEE Transactions on Knowledge and Data Engineering*. He is a member of the IEEE.

Jorge Civera received the BSc degree in computer science from the Universidad Politécnica de Valencia in 2002, and the Msc degree in computer science from Georgia Institute of Technology in 2003. He is currently a PhD student at Departamento de Sistemas Informáticos y Computación and a research assistant in the Instituto Tecnológico de Informática. He is also with a fellowship from the Spanish Ministry of Education and Culture. His research interests include bioinformatics, machine translation, and text mining.

Venu Dasigi received the BE degree in electronics and communication engineering from Andhra University in 1979, the MEE degree in electronic engineering from the Netherlands Universities Foundation for International Cooperation in 1981, and the MS and PhD degrees in computer science from the University of Maryland, College Park in 1985 and 1988, respectively. He is currently professor and chair of computer science at Southern Polytechnic State University in Marietta, Georgia. He is also an honorary professor at Gandhi Institute of Technology and Management in India. He held research fellowships at the Oak Ridge National Laboratory and the Air Force Research Laboratory. His research interests include text mining, information retrieval, natural language processing, artificial intelligence, bioinformatics, and computer science education. He is a member of ACM and the IEEE Computer Society.



Ashwin Ram received the PhD degree from Yale University in 1989, the MS degree from the University of Illinois in 1984, and the BTech degree from IIT Delhi in 1982. He is an associate professor in the College of Computing at the Georgia Institute of Technology, an associate professor of Cognitive Science, and an adjunct professor in the School of Psychology. He has published two books and more than 80 scientific papers in international forums. His research interests lie in artificial intelligence and cognitive science, and include machine learning, natural language processing, case-based reasoning, educational technology, and artificial intelligence applications.

Brian J. Ciliax received the BS degree in biochemistry from Michigan State University in 1981, and the PhD degree in pharmacology from the University of Michigan in 1987. He is currently an assistant professor in the Department of Neurology at Emory University School of Medicine. His research interests include the functional neuroanatomy of the basal ganglia, particularly as it relates to hyperkinetic movement disorders such as Tourette's Syndrome. Since 2000, he has collaborated with the coauthors on the development of a system to functionally cluster genes (identified by high-throughput genomic and proteomic assays) according to keywords mined from relevant MEDLINE abstracts.

Ray Dingledine received the PhD degree in pharmacology from Stanford. He is currently professor and chair of pharmacology at Emory University and serves on the Scientific Council of NINDS at NIH. His research interests include the application of microarray and associated technologies to identify novel molecular targets for neurologic disease, the normal functions and pathobiology of glutamate receptors, and the role of COX2 signaling in neurologic disease.