ACADEMIC PRESS

Journal of Biomedical Informatics 35 (2002) 25–36

Journal of Biomedical Informatics

www.academicpress.com

Visualization and evaluation of clusters for exploratory analysis of gene expression data

Ju Han Kim,[a,b,*] Isaac S. Kohane,[b] and Lucila Ohno-Machado[c]

[a] *SNUBI: Seoul National University Biomedical Informatics, Seoul National University School of Medicine, 28 Yongon-dong Chongno-gu, Seoul 110-799, Republic of Korea*

[b] *Children's Hospital Informatics Program, The Children's Hospital, Harvard Medical School, 300 Longwood Avenue, Boston, MA 02115, USA*

[c] *Decision Systems Group, Brigham and Women's Hospital, Harvard Medical School, 75 Francis St., Boston, MA 02115, USA*

Received 11 July 2001

Abstract

Clustering algorithms have been shown to be useful to explore large-scale gene expression profiles. Visualization and objective evaluation of clusters are two important considerations when users are selecting different clustering algorithms, but they are often overlooked. The developments of a framework and software tools that implement comprehensive data visualization and objective measures of cluster quality are crucial. In this paper, we describe a theoretical framework and formalizations for consistently developing clustering algorithms. A new clustering algorithm was developed within the proposed framework. We demonstrate that a theoretically sound principle can be uniformly applied to the developments of cluster-optimization function, comprehensive data-visualization strategy, and objective cluster-evaluation measures as well as actual implementation of the principle. Cluster consistency and quality measures of the algorithm are rigorously evaluated against those of popular clustering algorithms for gene expression data analysis (*K*-means and self-organizing maps), in four data sets, yielding promising results. © 2002 Elsevier Science (USA). All rights reserved.

Keywords: DNA microarray; Gene expression profile; Visualization; Cluster analysis; Optimization

1. Introduction

A general question in many research areas is how to organize observed data into meaningful structures. In a typical DNA microarray experiment, expression levels of thousands of genes are systematically recorded over tens of different samples (i.e., cell lines or tissues). Cluster analysis can be applied to explore the underlying similarity structures of observations and generate hypothetical clusters. Clustering algorithms have been didactically classified into two major categories, hierarchical and partitional. The former results in nested clusters, and the latter results in non-nested clusters. Both hierarchical and partitional algorithms can be implemented using an agglomerative or a divisive paradigm.

Hierarchical clustering algorithm transforms a pairwise (dis)similarity matrix of objects (or patterns) into a sequence of nested partitions, for example, a phylogenetic-type hierarchical tree (or a dendrogram). An agglomerative algorithm joins similar objects together into successively larger clusters in a bottom-up fashion (i.e., from the leaves to the root of the tree) by relaxing the threshold for joining objects or sets of objects [1]. This type of algorithm has been used extensively in gene expression analysis [2,3]. A variety of heuristics such as single-link, complete-link, and minimal spanning strategies can be used to determine when to merge objects and/or clusters.

A divisive algorithm forms clusters in the reverse order. For example, threshold-based clustering [4] starts with a completely connected graph from which edges (i.e., the "glue" that connects one object to a cluster or another object) are successively deleted to reveal

[*] Corresponding author. Fax: +822-747-4830.

E-mail addresses: juhan@snu.ac.kr (J.H. Kim), isaac_kohane@harvard.edu (I.S. Kohane), machado@dsg.harvard.edu (L. Ohno-Machado).

PII: S1532-0464(02)00001-1

'naturally emerging' clusters (i.e., sets of objects) at a certain threshold. This algorithm has also been used in gene expression analysis [5,6]. Creating a hierarchical-tree structure in a divisive top-down fashion (i.e., from the root to the leaves of the tree) by defining successive 'optimal' binary partitions was also successfully applied in functional genomics. Some authors used graph theory [7] and others used geometric space-partitioning principles [8] to accomplish this task.

Partitional clustering algorithms define a partition of objects into K clusters, such that the objects in a cluster are more similar to each other than to objects in different clusters. The value of K may or may not be given a priori. The clusters are not nested. A clustering criterion may be adopted to minimize within-cluster scatter or maximize between-cluster scatter [9]. A wide range of partitional algorithms have been successfully used to analyze gene expression data, including K-means [10,11], self-organizing maps (SOM) [12,13], CAST [14], and MCLUST [15].

The widely accepted dichotomy between the hierarchical and partitional clustering is misleading because it does not refer to a fundamental difference in the clustering principle. Rather, it describes mere 'procedural' aspects of clustering algorithms (so does the dichotomy between agglomerative and divisive categories). When there is a clear 'descriptive' definition, given the context of particular data analysis, of what the 'optimal' clustering is, it should not matter whether the actual implementation is hierarchical or partitional (or agglomerative or divisive).

A hierarchical classification can be constructed as a special nested sequence of partitions. It is possible to develop clustering algorithms in which both strategies are used. These strategies may strongly complement each other in the analysis of complex data.

2. Prior work

We have previously reported on the matrix incision tree (MITree) algorithm, a divisive hierarchical clustering algorithm that was able to reveal plausible clusters in gene expression data [8]. Using an intuitive geometric space-partitioning principle, MITree aims at iteratively determining the hyperplanes that 'optimally' partition a high-dimensional data space into two lower-dimensional subspaces, thereby creating a bifurcating hierarchical tree. Global optimization strategies such as evolution strategy [16] (i.e., a genetic algorithm) and deterministic annealing (i.e., a simulated annealing) were also successfully applied to the original algorithm.

From our previous experience with MITree, we verified that the hierarchical tree structure works best when the data structure is intrinsically hierarchical. Complex data like gene expression profiles tend to have mixed structures and nested substructures of different types that can best be captured by a proper combination of hierarchical and partitional structures. This paper describes a framework and formalizations for the consistent development of both hierarchical and partitional clustering algorithms based on the same principle.

2.1. Visualization of clustering structures

An important objective of hierarchical clustering approach is to provide a graphical overview of the data so that it can easily be interpreted. The color-coded expression patterns in accordance with a dendrogram support visual analysis [2]. The relevance networks approach creates a proximity graph (i.e., a threshold graph in which each edge is weighted according to its proximity) that graphically represents the similarity structure of data [5,6]. On the other hand, a threshold graph in general can be converted to the corresponding dendrogram [9]. If a dendrogram is drawn from a proximity graph and weighted with the proximity values, we call it a proximity dendrogram. Therefore, any hierarchical clustering algorithm can be seen as a method for transforming a proximity matrix into a (proximity) dendrogram or a threshold (or proximity) graph.

Partitional algorithms impose no or less structure on their clustering solutions than their hierarchical counterparts, partly due to their simple output, a cluster membership function. SOM tends to put clusters of similar patterns in neighboring cells and those of different patterns in distant cells. Tamayo et al. [13] argue that the geometric-grid structure imposed on the clusters by SOM is superior to the non-structure of popular partitional clustering algorithms such as K-means. The geometric-grid structure, however, represents only qualitative (i.e., non-quantitative) inter-cluster associations that are difficult to interpret.

Because a hierarchical classification can be viewed as a special nested sequence of partitions and a proximity graph can be converted to a proximity dendrogram (and vice versa), a unified view of the graphical representation of clustering structures having both hierarchical and partitional components can be developed.

3. Method

3.1. Data hyperspace and incisional hyperplane

Fig. 1 illustrates data hyperspace and *incisional* hyperplanes. In the completely connected graphs, the vertices represent objects and the edges represent associations (or similarity measures) between objects. In general, N objects and their geometric relationships can be fully represented in $(N-1)$-dimensional hyperspace and are separable into two lower-dimensional subspaces

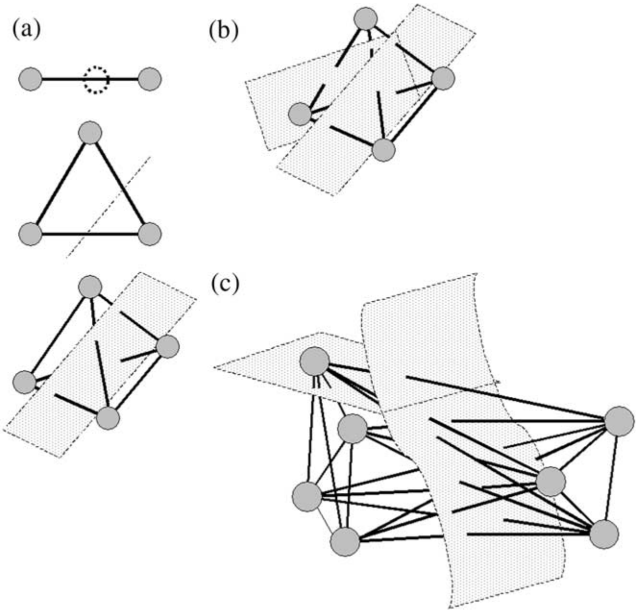


Fig. 1. Data hyperspace and *incisional* hyperplanes. (a) N objects can be arranged in $(N-1)-D$ space and are separable by $(N-2)-D$ bi-*incisional* plane (i.e., dot, line, or plane). (b) A multiple-*incisional* hyperplane that partitions the vertices into more than two groups can be viewed as the union of a set of bi-*incisional* hyperplanes that have the same dimensionality. In this example, four objects and their edges can be arranged in a 3-D space and separable into three subgroups by a set of two 2-D bi-*incisional* hyperplanes. (c) Seven objects in 6-D hyperspace are separated by two 5-D bi-*incisional* hyperplanes (or a multiple-*incisional* hyperplane, which is the union of the two bi-*incisional* hyperplanes) into three groups of one, three, and three members. Note that this is a severely distorted 3-D representation of the 6-D hyperspace, where all the 21 $((7*6)/2)$ links can have any Euclidean length. The multiple-*incisional* hyperplane deletes 15 $(3*3+1*3+1*3)$ edges and 6 $(3*2/2+3*2/2+1*0/2)$ edges will remain in the separated three lower-dimensional sub-spaces.

by a set of $(N-2)$-dimensional *incisional* hyperplanes (Fig. 1a). When a hyperplane separates N objects into two subgroups with m and n objects $(N=m+n)$, the plane deletes $m*n$ edges among the total $N(N-1)/2$ edges, and there are $2^{N-1}-1$ such binary *incisional* hyperplanes. We call such binary *incisional* hyperplanes as bi-*incisional* hyperplanes.

As shown in Figs. 1b and c, a high-dimensional data space can be partitioned into more than two sub-spaces by the union of bi-*incisional* hyperplanes, all with the same dimensionality. We call the union of the bi-*incisional* hyperplanes the multiple-*incisional* hyperplane. When a multiple-*incisional* hyperplane partitions N objects into K subgroups of $m_1, m_2, \ldots, m_K$ members, the hyperplane deletes $\sum_{i<j}|m_i||m_j|$ links among the total $N(N-1)/2$ links and the multiple-*incisional* hyperplane can be represented as the union of $K-1$ bi-*incisional* hyperplanes that partially overlap (Figs. 1b and c).

It is worth noting that we intentionally used the somewhat odd term, '*incisional*' hyperplane, to distinguish it from the separating hyperplane of support vector machines (SVM), which typically means maximum margin hyperplane characterized by the kernel and soft-margin-penalty functions in the statistical learning theory [17]. Instead of creating a lower-dimensional soft margin hyperplane with the small number of support vectors selected from the total training examples for supervised classification with SVM, we treated all N objects as marginal cases just adjacent to the $(N-2)$-dimensional bi-*incisional* hyperplane for unsupervised clustering.

3.2. Object similarity matrix

To manage such complex observations as gene expression data, we can view each gene or array as an *object* and the associations between genes (or arrays) as *connecting edges* between objects. In that way, we can create a comprehensive N-by-N object similarity matrix for N genes (or arrays).

Let $M=(O,E)$ be an object similarity matrix consisting of a set of objects and a set of connecting edges. We denote the object set of M by $O(M)$, the edge set by $E(M)$, and an edge between two objects by $e(o_i,o_j)\in E(M)$, where $o_i,o_j\in O(M)$. Any non-empty subset of M is called a cluster (including the object similarity matrix M itself).

The (within-cluster) average similarity measure of a cluster (or an object similarity matrix), $S_{\text{cluster}}(M)$, is defined as the mean similarity value of all the connecting edges in the corresponding cluster. If $S(x,y)$ is a similarity measure between objects x and y, then

$$S_{\text{cluster}}(M)=\frac{1}{|E(M)|}_{o\in O(M),i<j}\sum S(o_i,o_j). \tag{1}$$

Similarly, the between-cluster average similarity measure, $S_{\text{between}}(M_i,M_j)$, is defined as the mean similarity value of all between-cluster edges,

$$S_{\text{between}}(M_i,M_j)=\frac{1}{|O(M_i)||O(M_j)|}\sum_{o_u\in O(M_i),o_v\in O(M_j)}S(o_u,o_v). \tag{2}$$

3.3. Matrix incision index

Fig. 2 is an equivalent but much more manageable matrix representation of the hyperspace-partitioning problem discussed in Fig. 1. An $(N-1)$-dimensional space containing completely connected N objects with $N(N-1)/2$ connecting edges can be represented as an N-by-N object similarity matrix. The rectangular area, H, in Fig. 2a represents a bi-*incisional* hyperplane that separates the high-dimensional space of N objects into two lower-dimensional sub-spaces of $|O(M_1)|$ and $|O(M_2)|$ objects represented by the triangular areas, M_1 and M_2, respectively (Fig. 2a).

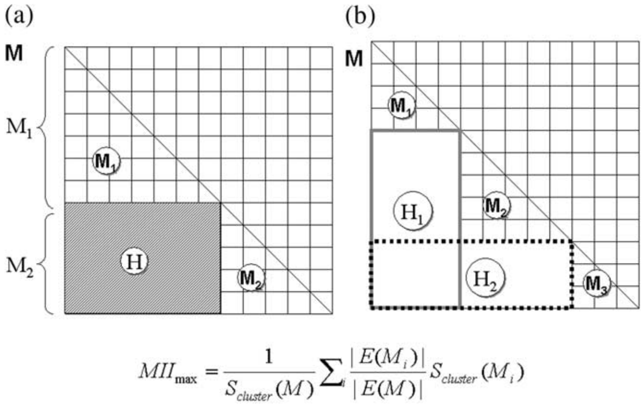


Fig. 2. Matrix representation of data hyperspace, *incisional* hyperplanes, and matrix incision indices (MIIs). (a) A bi-*incisional* hyperplane can be viewed as the rectangular area (H) that splits of an object similarity matrix (M) into two sub-matrices (M_1, M_2). (b) A multiple-*incisional* hyperplane that partitions M into K sub-matrices ($M_1, \ldots, M_k$) can be viewed as a union of $(K-1)$ bi-*incisional* hyperplanes ($H_1(M) \cup \cdots \cup H_{K-1}(M)$). The maximum matrix incision index, $\mathrm{MII}_{\max}$, for optimizing multiple-*incisional* hyperplane was defined as the fraction of gain (i.e., the weighted means of within-cluster average similarity measures of the K resulting clusters ($M_1, \ldots, M_k$)) from the parent matrix (i.e., the average similarity measure of the similarity matrix (M)).

Let $H(M) = \{e(o_1, o_1), \ldots, e(o_i, o_j)\}$, $H(M) \subset E(M)$, be a bi-*incisional* hyperplane, whose removal splits M into two disjoint non-empty subsets, A and B, such that $A, B \subset M$, $O(A) \cup O(B) = O(M)$, $E(A) \cup E(B) \cup H(M) = E(M)$, and $M = (O(A) \cup O(B), E(A) \cup E(B) \cup H(M))$. Therefore, the 'optimal' binary partitioning problem of this high-dimensional data space becomes a matrix incision problem of finding the 'optimal' bi-*incisional* hyperplane, $H(M)$, that minimizes the loss (a weighted function of $H(M)$) and/or maximizes the gain (a weighted function of $E(A)$ and $E(B)$) of the partitioning.

Let $P(M) = H_1(M) \cup H_2(M) \cdots \cup H_{k-1}(M)$ be a multiple-*incisional* hyperplane, whose removal splits M into K disjoint non-empty subsets, $M_1 \cdots M_K \subset M$, such that $O(M_1) \cup \cdots \cup O(M_K) = O(M)$, $E(M_1) \cup \cdots \cup E(M_K) \cup P(M) = E(M)$, and $M = (O(M_1) \cup \cdots \cup O(M_K)$, $E(M_1) \cup \cdots \cup E(M_K) \cup P(M))$. Then, the problem of 'optimal' multiple matrix incision (or the 'optimal' K partitioning) is to find the 'optimal' multiple-*incisional* hyperplane, $P(M)$, that minimizes a certain weight function of $P(M)$ and/or maximizes a certain weight function of $E(M_1), \ldots, E(M_K)$.

We defined $\mathrm{MII}_{\max}$ of a multiple-*incisional* hyperplane as the ratio of the weighted S_{cluster} of a set of clusters ($M_1, \ldots, M_k$) over the $S_{\mathrm{cluster}}(M)$. Therefore, the 'optimal' multiple partitioning can be obtained by searching the multiple-*incisional* hyperplane of a similarity matrix with the maximum $\mathrm{MII}_{\max}$,

$$\mathrm{MII}_{\max} = \frac{1}{S_{\mathrm{cluster}}(M)} \sum_i \frac{|E(M_i)|}{|E(M)|} S_{\mathrm{cluster}}(M_i). \tag{3}$$

3.4. MITree-K: the K-partitioning matrix incision tree algorithm

The classical minimum graph quotient problem, which is NP-complete (for review, see [18,19]), can be viewed as a special case of the geometric space-partitioning problem, which can therefore be solved using these (generalizations of) approximation algorithms. The geometric space-partitioning problem can be reduced to the classical minimum graph quotient problem by adding, between every pair of vertices, an edge of infinitesimal weight that has no effect on the values of cuts but changes the number of edges crossing any cut to the partitions. Multiple partitioning by determining the 'optimal' multiple-*incisional* hyperplane is obviously a harder problem than the binary partitioning problem.

Our cluster-optimization strategies based on deterministic annealing and evolution strategy [16] that were successfully applied to the binary partitioning problem were either computationally very expensive or inadequate in finding global optima for the harder problem of multiple partitioning applied to complex gene expression data. Thus, we have developed an efficient heuristic approximation algorithm by combining modified K-medoids, which can be viewed as a special case of EM (Expectation–Maximization) algorithm, and an intervening incremental trimming-and-reassignment strategy, which was introduced in our prior work [8], to facilitate local search.

The MITree-K algorithm (Fig. 3) tries to find the 'optimal' multiple-*incisional* hyperplane, $P(M)$, having the maximum $\mathrm{MII}_{\max}$. It first creates a set of 'candidate' clusters using classical Voroni-type partitioning by a set of iteratively converging cluster centers. Let S_{obj} be the average similarity measure between an object and all the others in its corresponding cluster,

$$S_{\mathrm{obj}}(x) = \frac{1}{|O(M)| - 1} \sum_{x, y \in O(M), x \neq y} S(x, y). \tag{4}$$

The center of a cluster (or an object similarity matrix) is defined as the object with the highest S_{obj} in the corresponding cluster.

MITree-K tries to harmoniously maximize $S_{\mathrm{cluster}}(M_i)$'s by means of incremental trimming-and-reassignment strategy. It trims objects with smaller S_{obj} in the 'candidate' clusters to create a set of dense 'core' clusters. It then incrementally reassigns the trimmed objects to the nearest 'core' clusters. As the result of the step may depend on the reassignment order, MITree-K repeats the trimming-and-reassignment step by gradually increasing the sizes of the trimmed 'core' clusters and updating cluster centers at each step. For simplicity, we increased the sizes of 'core' clusters from 50% of the corresponding 'candidate' clusters gradually up to 80, 90, 95, and then to 100% in the present study.

Step 1. Randomly select a set of initial cluster centers ($C = \{c_i, \ldots, c_k\}$).

Repeat steps 2 through 4 until the cluster-center membership stabilizes (i.e., no change in C).

Step 2. Generate a new partition by assigning each object (x) to its closest cluster center (c_i) with the highest similarity measure $S(x,c_i)$.

Step 3. Compute new cluster center (with the highest S_{obj}) for each candidate cluster.

Step 4. Repeat steps 2 and 3 until the convergence of the cluster centers.

Step 5. Apply incremental trimming-and-reassignment strategy for candidate clusters.

Step 6. Choose the set of clusters that shows the highest MII_{max} by repeating steps 1 to 5.

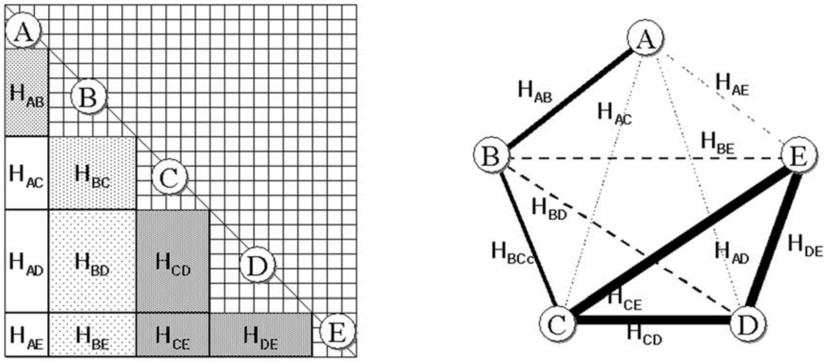
Fig. 3. MITree-*K* algorithm.

3.5. Systematic matrix decomposition and reconstruction for quantitative visualization of clustering structures

Clustering can be viewed as the decomposition process of a similarity matrix M into a set of clusters $(M_1, \ldots, M_k)$ and a multiple-*incisional* hyperplane, $P(M) = H_1(M) \cup \cdots \cup H_{k-1}(M)$. The hyperplane can also be decomposed into a set of $K(K-1)/2$ 'component' bi-*incisional* hyperplanes between each pair of clusters, $\{H(M_i \cup M_j) \,|\, i < j,\ 1 < j < K\}$, as shown in Fig. 4. Therefore, if we consider each cluster, M_i, as an object and each 'component' bi-*incisional* hyperplane, $(H(M_i \cup M_j))$, as a connecting edge between each pair of clusters, a similarity matrix of the clusters, $M(\{M_1, \ldots, M_k\}, \{H(M_i \cup M_j) \,|\, i < j,\ 1 < j < K\})$, can be created to capture the quantitative relationships among clusters.

The quantitative relationships among clusters can be graphically represented as a cluster proximity graph, where vertices are clusters and edges elementary bi-*incisional* hyperplanes weighted by the between-cluster average similarity measures of the corresponding hyperplanes, $S_{\text{between}}(M_i, M_j)$ (see Fig. 4).

One may argue that it is just an obvious graphical representation for any clustering solution. This is true to a certain degree. However, defining (within- and



$$P(M) = H_1(M) \cup H_2(M) \cup H_3(M) \cup H_4(M)$$
$$H_1(M) = H_{AB}(A \cup B) \cup H_{AC}(A \cup C) \cup H_{AD}(A \cup D) \cup H_{AE}(A \cup E)$$
$$H_2(M) = H_{AC}(A \cup C) \cup H_{BC}(B \cup C) \cup H_{AD}(A \cup D) \cup H_{BD}(B \cup D)\ U\ H_{AE}(A \cup E) \cup H_{BE}(B \cup E)$$
$$H_3(M) = H_{AD}(A \cup D) \cup H_{BD}(B \cup D) \cup H_{CD}(C \cup D) \cup H_{AE}(A \cup E)\ U\ H_{BE}(B \cup E) \cup H_{CE}(C \cup E)$$
$$H_4(M) = H_{AE}(A \cup E) \cup H_{BE}(B \cup E) \cup H_{CE}(C \cup E) \cup H_{DE}(D \cup E)$$

$$S_{between}(M_1, M_2) = \frac{1}{|O(M_1)||O(M_2)|} \sum_{u \in O(M_1), v \in O(M_2)} S(u,v)$$

Fig. 4. Quantitative visualization of clustering structure by systematic matrix decomposition and reconstruction. The algorithm decomposes the input similarity matrix into a set of clusters (A, ..., E) and a multiple-*incisional* hyperplane, $P(M)$, which can be decomposed into a set of 'component' bi-*incisional* hyperplanes (i.e., $\mathrm{H}_{\mathrm{AB}}, \mathrm{H}_{\mathrm{AC}}, \ldots, \mathrm{H}_{\mathrm{DE}}$). A cluster-proximity graph that regards clusters as objects and bi-*incisional* hyperplanes as edges provides quantitative visualization of clustering structure. Between-cluster distance is represented by cell darkness and line thickness in matrix and graphical representations, respectively.

between-) cluster similarity measures for the graphical representation may not be clear and requires arbitrary choices of measures in several other algorithms. It is not always the case that similarity measures conform to the corresponding cluster-optimization principles. This gets even more complicated when one tries to further partition the initial clusters to reveal substructures (see Fig. 7). Notice that in our framework the (within- and between-) cluster similarity measures, $S_{\text{cluster}}(M)$ and $S_{\text{between}}(M_i, M_j)$ (i.e., all the triangular and rectangular areas in Fig. 4), as well as the object similarity measure are all uniformly defined as the mean edge weights of all involved pairs of objects in a completely connected graph model with no assumption on the data distribution. The systematic matrix decomposition and reconstruction is a natural way of defining cluster similarity measures in our framework for the measures are also consistent in multilevel partitioning.

3.6. Evaluation data sets

Four well-studied data sets were used to evaluate the MITree-*K* algorithm: Fisher's iris data set [22], Golub's leukemia gene expression data set [20], Cho's yeast cell-cycle data set [21], and Iyer's human fibroblast gene expression data set [22]. The former two are tagged with known class labels and the latter two are not.

Fisher's iris data set consists of 150 observations of the three species of iris flowers (50 *Iris setosa*, 50 *Iris versicolor*, and 50 *Iris virginica*) and four discriminating measurements (petal and septal length, and petal and septal width). Golub's leukemia data set has 6817 human gene expression profiles of 74 cell lines (38 training and 34 test sets) of acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). The authors selected 50 genes that were most highly correlated with the AML/ALL class distinction by a supervised learning algorithm.

Cho's yeast and Iyer's human fibroblast data sets were used as real gene expression data sets that are unlabeled. Cho et al. monitored the expression levels of 6218 *S. cerevisiae* gene transcripts at 10-min intervals over two cell cycles (160 min). Filtering of genes that do not change significantly across samples returns 826 genes using the same procedure reported by Tamayo et al. [13]. The data preprocessing steps of removing 90-min time-point and normalizing each expression to have mean zero and variance one within each of the two cell cycles were also performed. Among the 8613 gene expression profiles of human fibroblasts stimulated by addition of serum after deprivation, 517 genes whose expression levels changed substantially across samples were analyzed.

3.7. Evaluation measures for consistency and quality

Clustering consistency and quality measures were applied to compare MITree-*K* with the two popular clustering algorithms, *K*-means and SOM, for gene expression data analysis using the "CLUSTER" software downloadable at http://rana.stanford.edu/software/.

Averaged Rand index [23] was applied to measure clustering consistency. Let $C(P_i(M), P_j(M))$ be a clustering consistency between two sets of K clusters of M. All edges in M, $E(M)$, can be divided into two disjoint subsets, the concordant subset, $L_{\text{concordant}} = \{e|(e \in P_i(M) \text{ and } e \in P_j(M)) \text{ or } (e \in E(M) - P_i(M) \text{ and } e \in E(M) - P_j(M))\}$ and the discordant subset, $L_{\text{discordant}} = \{e|(e \in P_i(M) \text{ and } e \in E(M) - P_j(M)) \text{ or } (e \in E(M) - P_i(M) \text{ and } e \in P_j(M))\}$. Thus,

$$C(P_i, P_j) = \frac{|L_{\text{concordant}}|}{|L_{\text{concordant}}| + |L_{\text{discordant}}|}. \quad (5)$$

Adjustment of the index [24] was not needed because we are comparing clustering solutions at equal-number levels (see results). Clustering consistency of N trials is defined as the average of all pair-wise clustering consistency measures,

$$C_N = \frac{2}{N(N-1)} \sum_{i<j} C(P_i, P_j). \quad (6)$$

For cluster quality, the homogeneity and separation indices by Sharan and Shamir [7] were used. Homogeneity indices are the average and minimum correlation coefficients between an object and the fingerprint of its corresponding cluster (i.e., the mean vector of the fingerprints of the members of the cluster). If $F(x)$ and $F(M)$ are the fingerprints of an object x and its corresponding cluster M, respectively, and $\text{Correl}(x, y)$ is the correlation coefficient of fingerprints x and y, then

$$H_{\text{avg}} = \frac{1}{|O(M)|} \sum_{x \in O(M)} \text{Correl}(F(x), F(M)), \quad (7)$$

$$H_{\min} = \min_{x \in O(M)} \text{Correl}(F(x), F(M)). \quad (8)$$

Separation indices are the weighted average and the maximum correlation coefficient between cluster fingerprints:

$$T_{\text{avg}} = \frac{1}{\sum_{i<j} |O(M_i)||O(M_j)|} \times \sum_{i<j} |O(M_i)||O(M_j)| \text{Correl}(F(M_i), F(M_j)), \quad (9)$$

$$T_{\max} = \max_{i<j} \text{Correl}(F(M_i), F(M_j)). \quad (10)$$

Thus, increased H_{avg} and $H_{\min}$ and decreased T_{avg} and $T_{\max}$ suggest better clustering solutions.

Notice that the weighted mean of average similarity measures, S_{cluster}, itself is also a good index for measuring cluster homogeneity (instead of H_{avg}). Applying the same analogy to replace the other three indices is also possible. We used the above indices for the purpose of 'fair' comparison with other algorithms.

We created 2 and 3 clusters for Fisher's iris data set and 2, 3, and 4 clusters for Golub's leukemia data set (i.e., up to their known number of actual classes) with 30 repeated simulations to measure clustering accuracy, quality, and consistency. Statistical significances of the difference among the three algorithms for the five measures were tested by ANOVA (Analysis of Variance). A p-value less than 0.05 was considered as significant and the Duncan method was used for post hoc multiple comparisons.

We created 10, 20, 30, 40, and 50 clusters for the two untagged gene expression data sets with 30 repetitions by the three algorithms (2(data sets) $\times$ 5(experiments, $K = 10, 20, \ldots, 50$) $\times$ 30(repetitions) $\times$ 3(algorithms) $= 900$) and measured clustering quality and consistency. The differences of the clustering consistency and quality measures across different experiments were tested by graphical plots and repeated measures ANOVA in which the five variables were considered as the repeated measures. A p-value smaller than 0.5 was considered as significant. ANOVA with the post hoc multiple comparison was applied to compare the experiments.

4. Results

4.1. Fisher's iris and Golub's leukemia data sets

Binary partitioning ($K = 2$) of the r^2 (i.e., the square of Pearson's product moment correlation coefficient) similarity matrix derived from Fisher's iris data set perfectly separated all *setosa* from the other species (accuracy of 100%) by MITree-K. Tri-partitioning ($K = 3$) correctly clustered the three species with five errors (accuracy of 95% (145/150)). All members of *setosa* were correctly clustered together, two *virginicas* (cases 9, 40) were incorrectly clustered with 48 *versicolors*, and three *versicolors* (cases 66, 77, 81) were incorrectly clustered with 47 *virginicas.*

Interestingly enough, bi-partitioning ($K = 2$) of the 74 cell lines from Golub's leukemia data set perfectly discriminated the 38 training and the 34 test set cases instead of separating AML and ALL cases (Table 1). The distinction between the training and test sets may come from the potential differences between the independently collected data sets, as described by Golub et al. [21] In the tri-partitioning experiment ($K = 3$), the first cluster had 26 ALL's of the training set, the second cluster 24 AML's with one ALL, and the third cluster 20 ALL's with one AML (72/74, 97% of accuracy). Quadri-partitioning ($K = 4$) successfully recovered all four natural groups with only two errors (72/74, 97%).

One very important finding was that MITree-K could successfully recover the three natural groups of leukemia's (i.e., B-ALL, T-ALL, and AML) from the training set with only two errors, a classification performance that is superior to that of previous studies [21]. One B-ALL (case 12) was clustered with AML's and another B-ALL (case 17) with T-ALL's in the present study with MITree-K (36/38, 95%).

Table 2 demonstrates the clustering consistency and quality measures of the three clustering algorithms in 30 trials using Fisher's iris ($K = 3$) and Golub's leukemia ($K = 4$) data sets. In the analysis of Fisher's iris data set, MITree -K and SOM resulted in cluster consistency and homogeneity scores that were significantly higher than those resulting from K-means. Tests for clustering separation (T_{avg} and $T_{\max}$) showed no statistically significant difference. For Golub's leukemia data set, MITree-K resulted in cluster consistency, homogeneity (H_{avg} and $H_{\min}$), and average separation (T_{avg}) that were significantly higher than those resulting from the two other algorithms. $T_{\max}$ was best in MITree-K and K-means. T_{avg} (K-means) was greater than T_{avg}(SOM) and C_{30}(SOM) was higher than C_{30}(K-means). $H_{\min}$(SOM) was higher than $H_{\min}$(K-means).

Table 1
Clustering accuracy of MITree-K tested in leukemia gene expression data set [21]

K			Clinical class	Case	Data set
2	3	4			
M$_1$	**M$_1$**	**M$_1$**	**26 ALL:**	1–11, 13–27	Training set
	M$_2$	**M$_2$**	**11 AML:** 1 ALL:	28–38 12	
M$_2$		**M$_3$**	**13 AML:**	50–54, 57, 58, 60–65	Test set
	M$_3$	**M$_4$**	**20 ALL:** 1 AML:	39–49, 55, 56, 59, 67–72 66	

K, number of clusters; M_K, clusters (or sub-matrices); AML, acute myeloid leukemia; ALL, acute lymphoblastic leukemia.

Table 2
Comparison of cluster consistency and quality of three clustering algorithms after 30 trials of clustering applied to Fisher's iris [20] and Golub's leukemia [21] data sets

Index	MITree-*K*	*K*-means	SOM
Fisher's iris data set			
C_{30}‡	$1.000 \pm 0.00^*$	0.914 ± 0.14	$1.000 \pm 0.00^*$
H_{avg}‡	$0.997 \pm 0.00^*$	0.994 ± 0.00	$0.997 \pm 0.00^*$
H_{min}‡	$0.977 \pm 0.00^*$	0.728 ± 0.10	$0.977 \pm 0.00^*$
T_{avg}	0.786 ± 0.00	0.783 ± 0.03	0.785 ± 0.00
T_{max}	0.978 ± 0.001	0.979 ± 0.01	0.979 ± 0.00
Golub's leukemia data set			
C_{30}‡	$0.999 \pm 0.00^{*,\dagger}$	0.896 ± 0.69	$0.932 \pm 0.66^*$
H_{avg}‡	$0.778 \pm 0.00^{*,\dagger}$	0.742 ± 0.03	0.738 ± 0.04
H_{min}‡	$0.478 \pm 0.00^{*,\dagger}$	0.069 ± 0.10	$0.361 \pm 0.11^*$
T_{avg}‡	$0.108 \pm 0.00^{*,\dagger}$	$0.135 \pm 0.02^{\dagger}$	0.147 ± 0.03
T_{max}‡	$0.312 \pm 0.00^{\dagger}$	$0.384 \pm 0.12^{\dagger}$	0.534 ± 0.20

All reported values are means ± SD.

Increased homogeneity and decreased separation suggest better clustering solution.

C_N, consistency index with N trials; H_{avg}, average homogeneity index; H_{min}, minimum homogeneity index; T_{avg}, average separation index; T_{max}, maximum separation index.

‡ $p < 0.0005$, significantly different among the three methods by ANOVA.

* Significantly different by ANOVA and separated by the post hoc Duncan multiple comparison method from K-means ($p < 0.05$).

† Significantly different by ANOVA and separated by the post hoc Duncan multiple comparison method from SOM ($p < 0.05$).

4.2. Yeast cell-cycle and human fibroblast data sets

Fig. 5 shows a comparison of clustering consistency measures after 30 trials, C_{30} (mean ± SD), of the three clustering algorithms applied to the yeast cell-cycle and the human fibroblast data sets, with increasing number of clusters ($K = 10, 20, 30, 40, 50$). MITree-*K* algorithm exhibited higher cluster consistency than the other algorithms in most of the experiments cluster consistency was significantly different among the three algorithms in both data sets after adjustment for the increasing numbers of clusters by repeated measures ANOVA ($p < 0.0005$). All the 10 ($5 * 2$) experiments (Figs. 5a and b) exhibited significant difference by ANOVA ($p < 0.0005$). In Fig. 5, algorithm names (designated as M, K, and S for MITree-*K*, *K*-means, and SOM, respectively) were separated by "/" when they were grouped separately by the post hoc Duncan multiple comparison method. The order of the algorithm names represents the order of performances.

Fig. 6 demonstrates the cluster homogeneity and separation measures of the three clustering algorithms. H_{avg}, H_{min}, T_{avg}, and T_{min} were all significantly different for the three clustering algorithms in both data sets (by repeated measures ANOVA ($p < 0.0005$) after adjustment for the effect of the increasing numbers of clusters). MITree-*K* demonstrated significantly higher homogeneity measures (H_{avg} and H_{min}) in all of the 20 ($5 * 4$) experiments by ANOVA and the post hoc Duncan

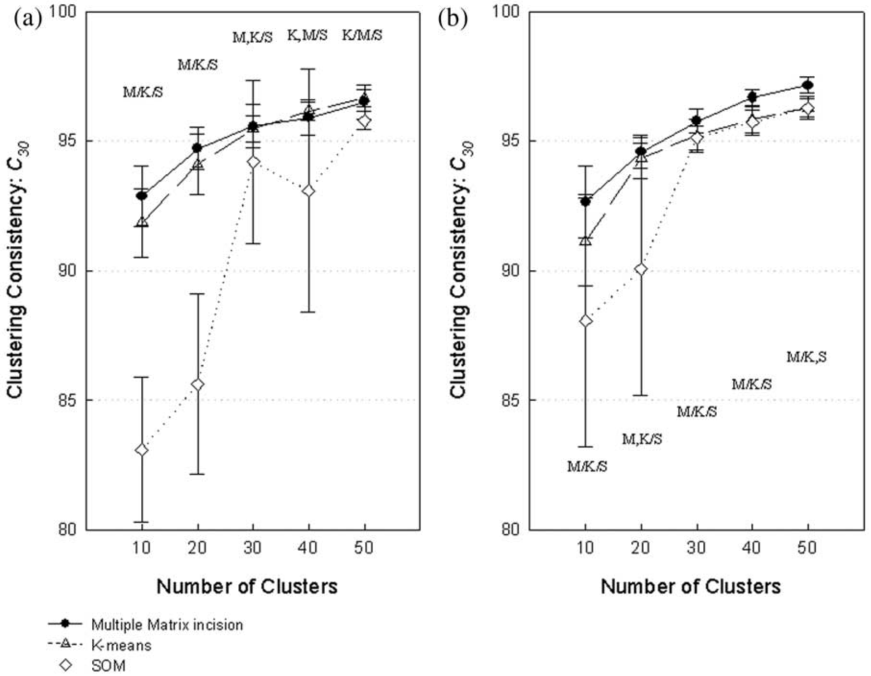


Fig. 5. Comparison of the clustering consistency among three clustering algorithms with increasing number of clusters ($K = 10, 20, 30, 40, 50$) for (a) yeast cell-cycle [22] and (b) human fibroblast [3] data sets. MITree-*K* showed better clustering consistency in general. Plots and error bars are mean ± SD. *M*: MITree-*K*, *K*: *K*-means algorithm, *S*: SOM. Algorithm names are separated by "/" when they are significantly different by ANOVA and separated by the posthoc Duncan multiple comparison method.

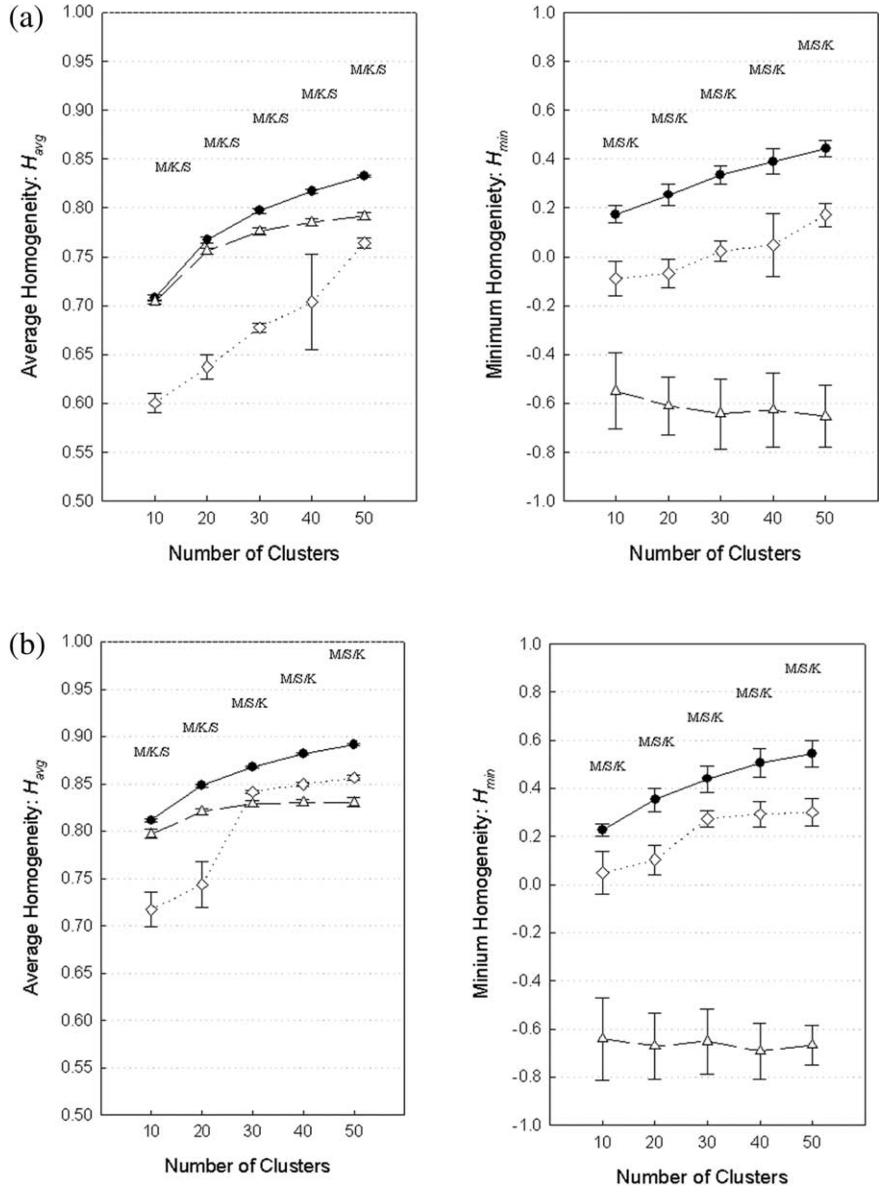


Fig. 6. Comparison of the clustering (a, b) homogeneity and (c, d) separation among three clustering algorithms with increasing number of clusters ($K = 10, 20, 30, 40, 50$) for (a, c) yeast cell cycle [22] and (b, d) human fibroblast [3] data sets. MITree-K showed the highest clustering homogeneity measures (H_{avg} and H_{min}) in both data sets (a, b). T_{avg} showed a tendency to be better in MITree-K and SOM than in K-means (c, d; left graphs). MITree-K showed lower (i.e., better) T_{max} than K-means or SOM in both data sets (c, d; right graphs). Plots and error bars are mean ± SD. M, MITree-K; K, K-means algorithm; S, SOM. Algorithm names are separated by "/" when they are significantly different by ANOVA and the post hoc Duncan multiple comparison method. Increased homogeneity and decreased separation suggest better clustering solution.

multiple comparison method ($p < 0.0005$). T_{avg} was higher in MITree-K and SOM than in K-means for both data sets (Figs. 6c and d, left graphs). MITree-K showed significantly higher T_{max} than K-means and SOM in both data sets (Figs. 6c and d, right graphs) by ANOVA and the Duncan method ($p < 0.0005$).

4.3. Quantitative visualization of clustering structures for further analysis

Fig. 7 demonstrates multilevel proximity-graph representations of clustering structures created by systematic matrix decomposition and reconstruction. For the purpose of illustration, we created 20 clusters from the yeast cell-cycle data, resulting three cluster proximity graphs (i.e., clusters of clusters, Figs. 7a–c) and three singleton clusters (Fig. 7d) at the threshold level of $S_{\mathrm{cluster}} = .25$, which can be determined using a permutation test [5]. We represent the size (i.e., the number of members) of cluster by the thickness of the borders and the similarity measures between clusters, S_{between} of the corresponding hyperplane, by the thickness of connecting edges.

In Fig. 7a, three (A, B, and C) among the seven clusters that show two cell-cycle patterns with different

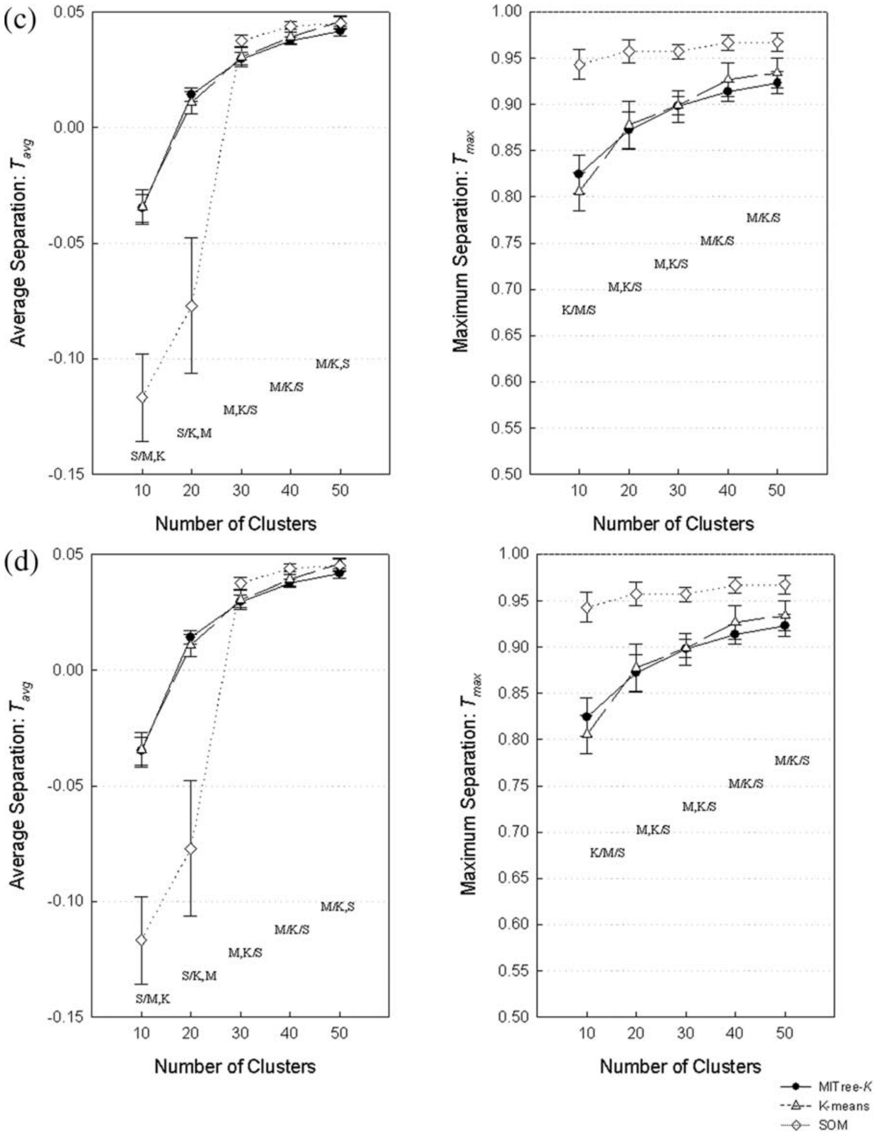


Fig. 6. (*continued*)

phases exhibit prominent cell-cycle patterns with two distinct peaks. The quantitative relationships among them are well represented by the connecting edges. Cluster A shows the earliest peak, followed by clusters B and then C. Clusters A and B ($S_{\text{between}} = .36$) are similar and clusters B and C are similar ($S_{\text{between}} = .27$) but clusters A and C are less similar ($S_{\text{between}} <$ threshold). The phase-lag pattern among A, B, and C are quantitatively represented such that they are linearly ordered to form "A–B–C–" pattern in the same cluster proximity graph (Fig. 7a).

Cluster proximity graphs can be nested to create multilevel representations capturing both hierarchical and partitional similarity structures (Figs. 7e and f). For example, cluster A in Fig. 7a, which shows the earliest peaks in the upper level, was sub-partitioned to reveal detailed sub-structures (Fig. 7e). For example, subcluster D in Fig. 7e shows a cluster of genes demonstrating even earlier peaks than A, B, or C as well as distinct two-cycle pattern, which is evidently inherited from the mother cluster A. Note that all the within- and between-cluster similarity measures are quantitatively visualized.

5. Conclusions

We have introduced a framework and formalizations for the consistent development of clustering algorithms that support both hierarchical and partitional structures as well as quantitative visualization. Complex data like gene expression profiles tend to have compound structures including both hierarchical and partitional substructures. The complexity can best be explored by a

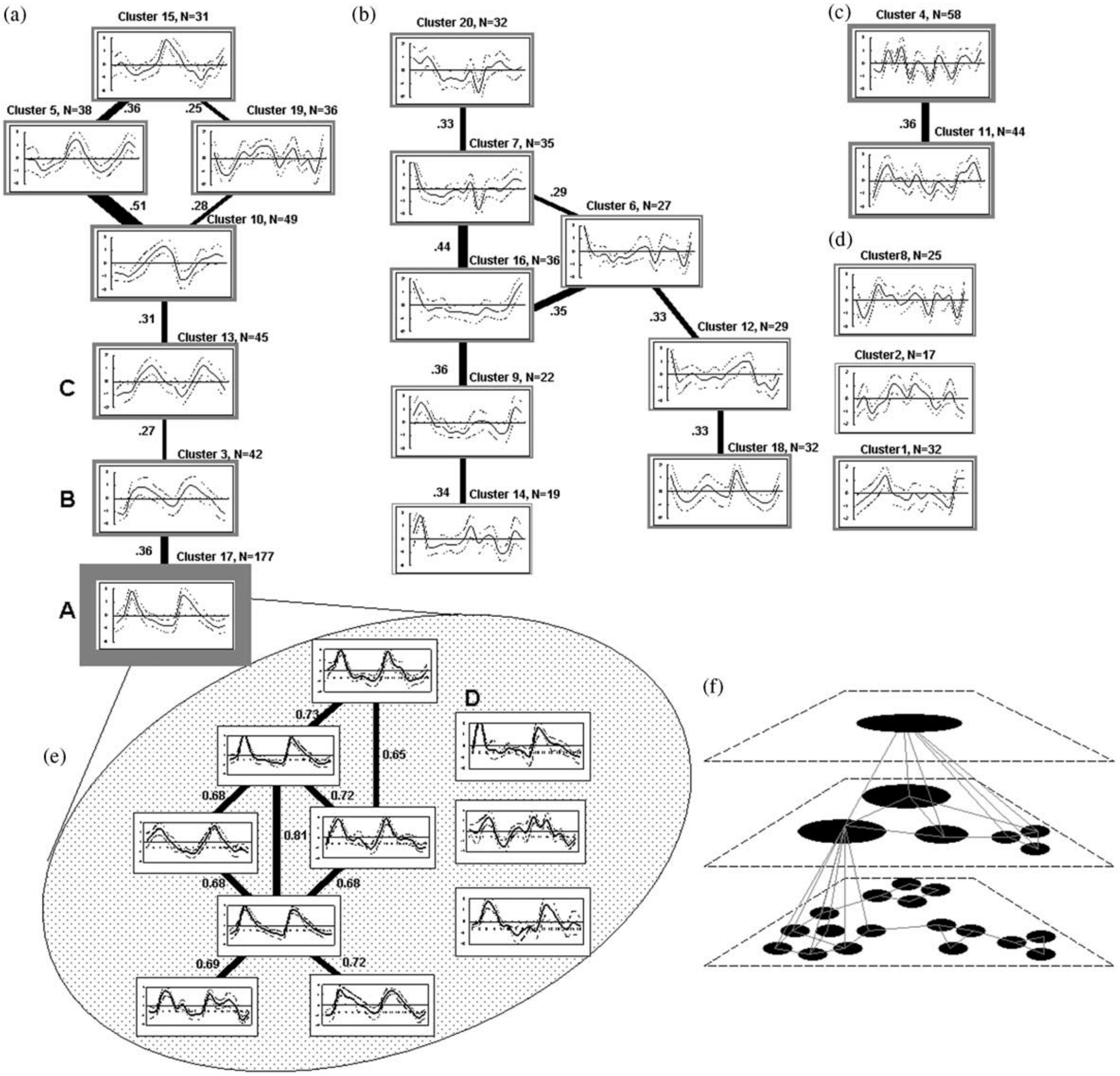


Fig. 7. Quantitative multilevel graphical representations of both hierarchical and partitional clustering structures of complex gene expression data (see results).

flexible and proper data analysis strategy. Hierarchical MITree [8] and the partitional MITree-*K* algorithm presented here are based on the same cluster-optimization principle, and may complement each other in the systematic exploration of similarity structures.

The matrix-incision framework makes no assumption on data distribution. It only uses very intuitive geometric properties of unique observations, such that all the within- and between-cluster similarity (and object similarity) measures can be uniformly developed. In contrast to the non-structure of *K*-means and the partial qualitative structure of SOM, the consistent development of similarity measurements of MITree's permits comprehensive multilevel quantitative visualization of similarity structures and substructures. The graphical representation may even be improved by other methods for similarity structure analysis. For example, multi-dimensional scaling can be applied to map the similarity structures of cluster proximity graphs, while maintaining the connecting edges and their weights between clusters.

In the comparison study with popular partitional clustering algorithms (*K*-means and SOM), MITree-*K* demonstrated higher accuracy, consistency, and quality. One widely known problem of center-based clustering algorithms such as (the non-soft implementations of) *K*-means and SOM is that they are sensitive to the initialization of the centers. Their winner-takes-all strategy makes the association between the data points and the local center so strong that the membership of a data point may be resistant to change. Although the basic MITree-*K* algorithm also relies on randomly initialized matrix centers, the matrix centers are designed to be non-fixed and free-floating, a feature that is achieved by using the incremental trimming-and-reassignment strategy that works also as a scheduling function for the optimization. The matrix centers are locally optimized in each trimming-and-reassignment step. We have also observed the matrix centers to converge rapidly (within 3–7 iterations), which may explain part of the lower sensitivity of MITree-*K* to outliers. The fundamental difference is that we use all pair-wise measures to determine cluster "centers," instead of using simple mean vectors (or centroids). Dynamic matrix centers with local optimization strategy may have been responsible for the improved clustering accuracy, consistency, and quality of MITree-*K*. We are currently investigating this issue.

The matrix-incision principle provides a consistent framework for clustering, by effectively separating the three basic layers of cluster analysis, i.e., (1) the choice of similarity measure, (2) the definition of cluster optimality, and (3) the implementation of the actual algorithm.

The significance of this work in terms of clinical or biological importance can be summarized as follows. Clustering algorithms have been extensively used in the clinical and genomics literature to facilitate knowledge extraction from large databases. The algorithms have been applied in an ad-hoc manner, and few authors justify their choices of similarity measures, cluster-optimization function, and algorithms. We describe a framework for cluster analysis and propose a consistent-optimization function, and a compatible algorithm. We show that our framework is sensible and that the implemented algorithm compares favorably to two popular clustering algorithms in terms of cluster accuracy, consistency, and quality. MITree-*K* also provides the extra benefit of allowing quantitative high-dimensional visualization of the resulting clusters. Our implementation can be found at http://www.snubi.org/MITree.

Acknowledgments

This study was supported by a grant of the Korea Health 21 R&D Project, Ministry of Health and Welfare, Republic of Korea (01-PJ10-PG6-01GM01-0004).

References

[1] Aldenderfer MS, Blashfield RK. Cluster analysis. Newbury Park, London: Sage Publications; 1984.

[2] Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci USA 1998;95(25):14863–8.

[3] Iyer VR, Eisen MB, Ross DT, Schuler G, Moore T, Lee JC, Trent JM, Staudt LM, Hudson Jr. J, Boguski MS, Lashkari D, Shalon D, Botstein D, Brown PO. The transcriptional program in the response of human fibroblasts to serum. Science 1999;283(5398):83–7.

[4] Duda RO, Hart PE, Stork DG. Pattern classification. 2nd ed. New York: Wiley-Interscience, 2000.

[5] Butte AJ, Kohane IS. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. Pac Symp Biocomput 2000:418–29.

[6] Butte AJ, Tamayo P, Slonim D, Golub TR, Kohane IS. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. Proc Natl Acad Sci USA 2000;97(22):12182–6.

[7] Sharan R, Shamir R. CLICK: a clustering algorithm with applications to gene expression analysis. Proc Int Conf Intell Syst Mol Biol 2000:307–16.

[8] Kim JH, Ohno-Machado L, Kohane IS. Unsupervised learning from complex data: the matrix incision tree algorithm. Pac Symp Biocomput 2001:30–41.

[9] Jain AK, Dubes RC. Partitional clustering. In: Jain AK, Dubes RC, editors. Algorithms for clustering data. Englewood Cliffs, NJ: Prentice-Hall; 1988. p. 89–133.

[10] MacQueen JB. Some methods for classification and analysis of multivariate observations. In: Le Cam LM, Neyman J, editors. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol 1. Berkeley: University of California Press; 1967. p. 281–97.

[11] Tavazoie S, Hughes JD, Campbell MJ, Cho RJ, Church GM. Systematic determination of genetic network architecture. Nature Genetics 1999;22:281–5.

[12] Kohonen T. Self-organized formation of topologically correct feature maps. Biol Cybern 1982;43:59–69.

[13] Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Golub TR. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proc Natl Acad Sci USA 1999;96(6):2907–12.

[14] Ben-Dor A, Yakhini Z. Clustering gene expression patterns. In: RECOMB99: The 3rd Annual Conference on Research in Computational Molecular Biology, Lyon, France. 1999. p 33–42.

[15] Yeung KY, Fraley C, Murua A, Raftery AE, Ruzzo WL. Model-based clustering and data transformations for gene expression data. Bioinformatics 2001;17:977–87.

[16] Lee K, Kim JH, Chung TS, Moon BS, Lee H, Kohane IS. Evolution strategy applied to global optimization of clusters in gene expression data of DNA microarrays. In: Proceedings of the 2001 IEEE Congress in Evolutionary Computation. 2001. p 845–50.

[17] Vapnik VN. Methods of pattern recognition. In: Vapnik VN, editor. The nature of statistical learning theory. New York: Springer; 1999. p. 123–69.

[18] Leighton FT, Maggs BM, Rao SB. Universal packet routing algorithms. In: Proceedings of the 29th Annual Symposium on Foundations of Computer Science. 1988. p 256–71.

[19] Aumann Y, Rabani Y. An O($\log k$) approximate min-cut max-flow theorem and approximation algorithm. SIAM J Comput 1998;27(1):291–301.

[20] Fisher RA. The use of multiple measurements in taxonomic problems. Annals of Eugenics 1936;17(Part II):179–88.

[21] Golub TR, Slonim DK, Tamayo P, Huard C, Caasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 1999;286:531–7.

[22] Cho RJ, Campbell JJ, Winzeler EA, Steinmetz J, Conway A, Wodicka L, Wolfsberg TG, Gabrielian AE, Landsman D, Lockart DJ, Davis RW. A genome-wide transcriptional analysis of the mitotic cell cycle. Mol Cell 1998;2:65–73.

[23] Rand WM. Objective criteria for the evaluation of clustering methods. J Am Stat Assoc 1971;66:846–50.

[24] Hubert L, Arabie P. Comparing partitions. J Classification 1985;1:193–218.