JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 11, Number 1, 2004

Pp. 147–161

Gene Clustering Based on Clusterwide Mutual Information

XIAOBO ZHOU,[1] XIAODONG WANG,[2] EDWARD R. DOUGHERTY,[1,3]
DANIEL RUSS,[4] and EDWARD SUH[4]

ABSTRACT

Cluster analysis of gene-wide expression data from DNA microarray hybridization studies has proved to be a useful tool for identifying biologically relevant groupings of genes and constructing gene regulatory networks. The motivation for considering mutual information is its capacity to measure a general dependence among gene random variables. We propose a novel clustering strategy based on minimizing mutual information among gene clusters. Simulated annealing is employed to solve the optimization problem. Bootstrap techniques are employed to get more accurate estimates of mutual information when the data sample size is small. Moreover, we propose to combine the mutual information criterion and traditional distance criteria such as the Euclidean distance and the fuzzy membership metric in designing the clustering algorithm. The performances of the new clustering methods are compared with those of some existing methods, using both synthesized data and experimental data. It is seen that the clustering algorithm based on a combined metric of mutual information and fuzzy membership achieves the best performance. The supplemental material is available at *www.gspsnap.tamu.edu/gspweb/zxb/glioma_zxb*.

Key words: gene microarray, clustering, mutual information, simulated annealing, bootstrap technique, K-means, fuzzy C-means.

1. INTRODUCTION

To understand the nature of cellular functions, it is necessary to study the behavior of genes in a holistic (Akutsu *et al.*, 2000; Debouck and Goodfellow, 1999; Huang, 1999; Kauffman, 1993; Shmulevich *et al.*, 2002) rather than in an individual manner because the expressions and activities of genes are not isolated or independent of each other. Due to the large number of genes and the high complexity of biological networks, clustering is a useful exploratory technique for the analysis of gene expression data. Clustering has been used in a number of studies to obtain a global, unsupervised perspective on the similarity of expression profiles (Ben-Dor *et al.*, 1999; Bittner *et al.*, 2000; Claverie, 1998; Dougherty

[1]Department of Electrical Engineering, Texas A&M University, College Station, TX 77843.
[2]Department of Electrical Engineering, Columbia University, New York, NY 10027.
[3]Department of Pathology, University of Texas M.D. Anderson Cancer Center, Houston, TX 77030.
[4]High Performance Computing and Informatics Office NIH—Center for Information Technology, Building 12A, Room 2011, Bethesda, MD 20892.

et al., 2002; Eisen *et al.*, 1998; Iyer, 1999). A wide range of clustering algorithms has been proposed to analyze gene expression data, including hierarchical clustering (Eisen *et al.*, 1998), self-organizing maps (Tamayo *et al.*, 1999), K-means (Tavazoie *et al.*, 1999), graph-theoretic approaches (Ben-Dor *et al.*, 1999), support vector machines (Brown *et al.*, 2000) and fuzzy C-means (Dougherty *et al.*, 2002). Successes in application have been reported for many clustering approaches (Ghosh and Chinnaiyan, 2002; Horimoto and Toh, 2001; Lukashin and Fuchs, 2001; Strehl and Ghosh, 2002; Xing and Karp, 2001; Yeung *et al.*, 2001) but so far no single method has emerged as the method of choice in the gene expression analysis community.

In this paper, we develop a new gene clustering strategy based on minimizing the mutual information among clusters. Pertinent to our approach is a study based on computing the mutual information for all pairs of genes and then choosing a threshold of the mutual information to create clusters of genes encompassing those with mutual information higher than the threshold (Butte and Hohane, 2000). These have been similar treatments (Chen and Church, 2000; Friedman *et al.*, 1999, 2000; D'Haeseleer *et al.*, 1999, 2000; Michaels *et al.*, 1998). These works are based on pairwise mutual information (PMI) and thus essentially only explore the marginal distributions of the multi-dimensional data. Our clustering strategy is based on minimizing the mutual information of the variables among clusters, and hence it fully explores the underlying *joint* probability distribution of the data. Bootstrap techniques (Zoubir and Boashash, 1998) are employed to obtain more accurate estimates of the mutual information for the typically small sets of data samples. Data are assumed to be truncated (Chen *et al.*, 1997).

Mutual-information-based clustering minimizes the statistical correlation among clusters, whereas the traditional K-means and fuzzy C-means algorithms minimize the total variance within different clusters. It is natural to consider combining these two paradigms to obtain more effective clustering techniques. To this end, we propose two clustering algorithms based on a combined mutual information and Euclidean distance criterion and a combined mutual information and fuzzy membership criterion. The performances of the new clustering methods are compared with that of some existing methods, using both synthesized data and experimental data.

The rest of this paper is organized as follows. In Section 2, we formulate the new clustering strategy based on mutual information minimization. We also discuss the bootstrap procedure for estimating the mutual information from a small dataset, as well as the simulated annealing procedure for solving the corresponding optimization problem. In Section 3, we propose two additional clustering methods based on combining the mutual information metric and the conventional Euclidean distance or the fuzzy membership metric. In Section 4, we provide performance comparisons of the above clustering methods using synthesized data. In Section 5, we present clustering results on gene microarray measurement data. Section 6 contains the conclusions.

2. GENE CLUSTERING VIA MUTUAL INFORMATION MINIMIZATION

In this section, we propose a new gene clustering method based on mutual information minimization. We introduce the concepts of mutual information and normalized mutual information, discuss bootstrap procedures for estimating the mutual information from a small dataset, introduce clustering based on pairwise mutual information, formulate the new clustering strategy based on mutual information minimization, and present a simulated annealing algorithm for solving the corresponding optimization.

2.1. Mutual information

The motivation for considering mutual information is its capability to measure a general dependence among random variables. Shannon's information theory provides a suitable formalism for quantifying such a concept. The *entropy* of a gene expression pattern is a measure of the uncertainty information content in that pattern. Given a random vector $\boldsymbol{X}$ and its probability distribution $P(\boldsymbol{X} = \boldsymbol{x}_i),\ i = 1, \cdots, N_x$, where N_x is the number of possible values $\boldsymbol{X}$ can take, the *entropy* is defined as

$$H(\boldsymbol{X}) \triangleq -\sum_{i=1}^{N_x} P(\boldsymbol{X} = \boldsymbol{x}_i) \log P(\boldsymbol{X} = \boldsymbol{x}_i). \tag{1}$$

Higher entropy for gene variables means that their expression levels are more randomly distributed. The *joint entropy* of $\boldsymbol{X}$ and $\boldsymbol{Y}$ is a measure of the uncertainty information between $\boldsymbol{X}$ and $\boldsymbol{Y}$, and is defined by

$$H(\boldsymbol{X}, \boldsymbol{Y}) \triangleq -\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} P(\boldsymbol{X} = \boldsymbol{x}_i, \boldsymbol{Y} = \boldsymbol{y}_j) \log P(\boldsymbol{X} = \boldsymbol{x}_i, \boldsymbol{Y} = \boldsymbol{y}_j), \tag{2}$$

where N_y is the number of possible values $\boldsymbol{Y}$ can take. When certain variables are known and others are not, the remaining uncertainty is measured by the *conditional entropy*

$$\begin{aligned} H(\boldsymbol{Y}|\boldsymbol{X}) &\triangleq \sum_{i=1}^{N_x} P(\boldsymbol{X} = \boldsymbol{x}_i) H(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}_i) \\ &= -\sum_{i=1}^{N_x} P(\boldsymbol{X} = \boldsymbol{x}_i) \sum_{j=1}^{N_y} P(\boldsymbol{Y} = \boldsymbol{y}_j|\boldsymbol{X} = \boldsymbol{x}_i) \log P(\boldsymbol{Y} = \boldsymbol{y}_j|\boldsymbol{X} = \boldsymbol{x}_i) \\ &= -\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} P(\boldsymbol{X} = \boldsymbol{x}_i, \boldsymbol{Y} = \boldsymbol{y}_j) \log P(\boldsymbol{Y} = \boldsymbol{y}_j|\boldsymbol{X} = \boldsymbol{x}_i). \end{aligned} \tag{3}$$

The *mutual information* between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is a measure of information about $\boldsymbol{X}$ (or $\boldsymbol{Y}$) contained in $\boldsymbol{Y}$ (or $\boldsymbol{X}$) and is given by

$$I(\boldsymbol{X}; \boldsymbol{Y}) \triangleq H(\boldsymbol{Y}) - H(\boldsymbol{X}|\boldsymbol{Y}) = H(\boldsymbol{X}) - H(\boldsymbol{Y}|\boldsymbol{X}) \tag{4}$$

$$= H(\boldsymbol{X}) + H(\boldsymbol{Y}) - H(\boldsymbol{X}, \boldsymbol{Y}) \tag{5}$$

$$= \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} P(\boldsymbol{X} = \boldsymbol{x}_i, \boldsymbol{Y} = \boldsymbol{y}_j) \log \frac{P(\boldsymbol{X} = \boldsymbol{x}_i, \boldsymbol{Y} = \boldsymbol{y}_j)}{P(\boldsymbol{X} = \boldsymbol{x}_i) P(\boldsymbol{Y} = \boldsymbol{y}_j)}. \tag{6}$$

It is known that mutual information is always nonnegative, i.e., $I(\boldsymbol{X}; \boldsymbol{Y}) \geq 0$ (Cover and Thomas, 1991).

Note that the mutual information defined in (4) is not normalized; and $I(\boldsymbol{X}; \boldsymbol{Y})$ can be quite small even if $\boldsymbol{X}$ and $\boldsymbol{Y}$ are highly correlated since $H(\boldsymbol{X})$ and $H(\boldsymbol{Y})$ may be small. Therefore, we normalize $I(\boldsymbol{X}; \boldsymbol{Y})$ by the maximal entropy of each of the contributing sequences, giving a high value for highly correlated sequences, independent of the individual entropy (Michaels *et al.*, 1998):

$$\bar{I}(\boldsymbol{X}; \boldsymbol{Y}) = \frac{I(\boldsymbol{X}; \boldsymbol{Y})}{\max\{H(\boldsymbol{X}), H(\boldsymbol{Y})\}}. \tag{7}$$

Unlike the Euclidean distance, this measure also recognizes negatively and nonlinearly correlated data sets as proximal (Michaels *et al.*, 1998).

The probabilities in (6) can be estimated by the corresponding histograms, i.e.,

$$P(\boldsymbol{X} = \boldsymbol{x}_i, \boldsymbol{Y} = \boldsymbol{y}_j) \cong \frac{\#(\boldsymbol{x}_i, \boldsymbol{y}_j)}{M}, \tag{8}$$

$$P(\boldsymbol{X} = \boldsymbol{x}_i) \cong \frac{\#(\boldsymbol{x}_i)}{M}, \tag{9}$$

$$P(\boldsymbol{Y} = \boldsymbol{y}_j) \cong \frac{\#(\boldsymbol{y}_j)}{M}, \tag{10}$$

where M is the total number of samples, and $\#(\boldsymbol{x}_i)$ denotes the number of occurrences of $\boldsymbol{x}_i$.

2.2. Mutual information estimation based on bootstrap

In practice, the sample size M is typically small, compared with the total number of possible values N_x and N_y. In order to get a more accurate estimate of the mutual information, we resort to the bootstrap technique (Zoubir and Boashash, 1998).

Let $z = [z_1, z_2, \cdots, z_N]$ denote the vector of N gene variables, where $z_i \in \{-1, 0, 1\}$. Denote $\boldsymbol{Z} = [z(1), z(2), \cdots, z(M)]$ as M realizations (i.e., samples) of z. At each iteration of the bootstrap procedure, M random draws are performed on $\boldsymbol{Z}$, to form a "resample" $\boldsymbol{Z}^* = \left[z^*(1), z^*(2), \cdots, z^*(M)\right]$, and the mutual information is computed based on the resample. The basic bootstrap method for estimating the mutual information is summarized as follows.

Algorithm 1 (Basic bootstrap procedure for mutual information estimation).

- *For* $n = 1, 2, \cdots, P$
  —*Resample: Draw a random sample* $\boldsymbol{Z}_n^*$ *of* M *values from* $\boldsymbol{Z}$*;*
  —*Calculate the estimated mutual information* $\bar{I}_n$ *based on the resample* $\boldsymbol{Z}_n^*$*;*
- *Sort the bootstrap estimates* $\bar{I}_n$, $n = 1, \cdots, P$*, according to increasing order to obtain* $\bar{I}_{k_1}, \bar{I}_{k_2}, \cdots, \bar{I}_{k_P}$*;*
- *The desired* $(1 - \alpha)$ *100% bootstrap confidence interval is* $\left(\bar{I}_{k_p}, \bar{I}_{k_q}\right)$*, where* $p = \lfloor P\alpha/2 \rfloor$ *and* $q = P - p + 1$*;*
- *The final estimated mutual information* $\hat{\bar{I}}$ *is the mean of the mutual information values in the interval* $\left(\bar{I}_{k_p}, \bar{I}_{k_q}\right)$.

We set $\alpha = 0.05$ and $P = 1{,}000$ in our simulations. According to Zoubir and Boashash (1998), Algorithm 1 can be substantially improved because the interval calculated is an interval with coverage less than the nominal values (Robinson, 1988). Next we give a more sophisticated bootstrap algorithm that would lead to a more accurate estimate of mutual information (Zoubir and Boashash, 1998).

Algorithm 2 (Percentile-*t* bootstrap technique for mutual information estimation).

- *Calculate the mutual information* $\bar{I}$ *based on initial sample* $\boldsymbol{Z}$*;*
- *For* $n = 1, 2, \cdots, P$
  —*Resample: Draw a random sample* $\boldsymbol{Z}_n^*$ *of* M *values from* $\boldsymbol{Z}$*;*
  —*Calculate the estimated mutual information* $\bar{I}_n$ *based on the resample* $\boldsymbol{Z}_n^*$*; use nested bootstrap to estimate the standard deviation* $\hat{\sigma}_n$ *of* $\bar{I}_n$ *(i.e., estimate* $\hat{\sigma}_n$ *using bootstrap technique again). Then, form*

$$\Upsilon_n = \frac{\bar{I}_n - \bar{I}}{\hat{\sigma}_n}. \tag{11}$$

- *Sort the bootstrap estimates* Υ_n, $n = 1, \cdots, P$*, according to increasing order to obtain* $\Upsilon_{k_1}, \Upsilon_{k_2}, \cdots, \Upsilon_{k_P}$*; and estimate the standard deviation* $\hat{\sigma}$ *from* $\bar{I}_1, \cdots, \bar{I}_P$.
- *The desired* $(1 - \alpha)$ *100% bootstrap confidence interval is* $\left(\bar{I} - \hat{\sigma}\,\Upsilon_{k_p}, \bar{I} - \hat{\sigma}\,\Upsilon_{k_q}\right)$*, where* $p = \lfloor P\alpha/2 \rfloor$ *and* $q = P - p + 1$.
- *The final estimated mutual information* $\hat{\bar{I}}$ *is the mean of the mutual information values in the interval* $\left(\bar{I} - \hat{\sigma}\,\Upsilon_{k_p}, \bar{I} - \hat{\sigma}\,\Upsilon_{k_q}\right)$.

We set $\alpha = 0.05$ and $P = 500$ in our simulations for this algorithm. We perform 50 resamples for each $\boldsymbol{Z}_n^*$ to estimate the standard deviation $\hat{\sigma}_n$ in the nested bootstrap step. Obviously, Algorithm 2 has a much higher computational complexity than Algorithm 1.

Next, we introduce the clustering technique based on pairwise mutual information. Then we describe our proposed method.

2.3. Clustering based on pairwise mutual information

Here we describe a threshold clustering algorithm based on pairwise mutual information (PMI). In Heyer *et al.* (1999), a threshold clustering algorithm is developed using jackknife correlation. We apply this idea

to the pairwise mutual information-based cluster analysis. The PMI works as follows: a candidate cluster is formed by starting with the first gene and grouping the gene that has smallest mutual-information-based distance with the target gene. The distance is defined as

$$d(\boldsymbol{X};\boldsymbol{Y}) = 1 - \bar{I}(\boldsymbol{X};\boldsymbol{Y}) = 1 - \frac{I(\boldsymbol{X};\boldsymbol{Y})}{\max\{H(\boldsymbol{X}), H(\boldsymbol{Y})\}}. \tag{12}$$

Each iteration adds the gene that has a minimal distance to the target gene to the cluster. The process continues until no gene can be added without surpassing the distance threshold. A second candidate cluster is formed by starting with the second gene and repeating the same procedure. Note that all genes are made available to the second gene, that is, the genes from the first candidate cluster are not removed from consideration. The process continues for all genes. The largest candidate cluster is selected and retained. The genes in the largest candidate cluster are removed from the whole gene set, and the entire procedure is repeated on the smaller gene set. For the predefined cluster number case, K in this study, when the number of clusters reaches K, add all the remaining genes to the last cluster. In this algorithm, the threshold is chosen as the mean of the distances of all gene pairs, or chosen empirically (Butte and Hohane, 2000).

2.4. Problem formulation

In this paper, we fix the number of clusters. Suppose we are to partition the set of gene variables into K disjoint subsets as $\boldsymbol{X}_1 \bigcup \boldsymbol{X}_2 \cdots \bigcup \boldsymbol{X}_K$. The cost function is defined as the sum of pairwise mutual information between any two subsets,

$$f(s) = \sum_{i \neq j} \bar{I}(\boldsymbol{X}_i; \boldsymbol{X}_j), \tag{13}$$

where s denotes a particular partition scheme. The simulated annealing algorithm is employed to find an optimal partition scheme such that the cost function attains the minimum, i.e.,

$$s^* = \arg\min_{s \in S} f(s), \tag{14}$$

where S denotes the set of all possible K-partition schemes.

2.5. Optimization algorithm

We employ the simulated annealing algorithm (Aarts and Emile, 1989) to minimize the cost function of (Eisen *et al.*, 1998). The basic procedure involves a cooling procedure, in which a temperature parameter starts out high and is gradually lowered until the system is frozen. At each temperature, the state is perturbed many times, which avoids the limitation of being initialization dependent. The algorithm moves to the next temperature in the schedule until the system reaches the thermal equilibrium at that temperature on the basis of a decreasing energy cost function.

Let s_0 and s_1 denote two different K-partition schemes with cost values $f(s_0)$ and $f(s_1)$, respectively. Then s_1 is accepted from s_0 according to the acceptance probability:

$$P\{\text{accept } s_1\} = \begin{cases} 1, & \text{if } f(s_1) \leq f(s_0), \\ \exp\left(\dfrac{-[f(s_1) - f(s_0)]}{T}\right), & \text{if } f(s_1) > f(s_0), \end{cases} \tag{15}$$

where $T \in \mathbb{R}^+$ denotes the temperature parameter. Since the basic simulated annealing procedure suffers from very slow convergence, we resort to a parallel annealing procedure (Ingber and Rosen, 1992). The basic idea is to run a set of K partitions in parallel. The initial temperature parameter is set as 1, and this parameter T is updated according to $T \leftarrow 0.85T$.

To generate a new partition $s_i, i = 1, 2, \cdots, K$ from the initial partition s_0, we randomly select two clusters from s_0 and then randomly pick a gene variable from one cluster and put it in the other cluster.

3. COMBINED MUTUAL-INFORMATION AND DISTANCE-BASED CLUSTERING ALGORITHMS

The Euclidean distance measure can capture only positive correlations between temporal gene expression patterns, whereas mutual information can capture any correlative behavior (positive, negative, and nonlinear) between expression time series (Michaels *et al.*, 1998). When the sample size is large, the mutual information can be estimated accurately, and then the mutual-information minimization clustering exhibits optimality. However, in practice, the data sample size is small and mutual information estimation is problematic. In order to enhance the clustering performance under such a condition, we propose to combine the mutual information criterion and the traditional distance criterion in designing the clustering algorithm.

3.1. Clustering based on combined mutual information and Euclidean distance metrics

Recall that in the K-means algorithm for clustering, associated with each gene i, we have an observation vector $\boldsymbol{y}_i = [y_i(1), y_i(2), \cdots, y_i(M)]$, $i = 1, 2, \cdots, N$. Suppose that the genes are partitioned into K clusters, with centroids $\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_K$. Denote $1_{i,k}$ as an indicator such that $1_{i,k} = 1$ if gene i belongs to cluster k, and $1_{i,k} = 0$ otherwise. Then the objective function associated with a particular partition s is

$$g(s) = \sum_{k=1}^{K} \sum_{i=1}^{N} \left\| \boldsymbol{y}_i 1_{i,k} - \boldsymbol{c}_k \right\|^2. \tag{16}$$

The K-means algorithm for clustering is as follows. Given a partition s (i.e., given the values $\{1_{i,k}\}$), we calculate the centroid of each cluster $\boldsymbol{c}_k$. We then reassign each $\boldsymbol{y}_i$ to its nearest centroid to get a new partition s^*. This procedure is repeated until there is no more change in the partition.

The mutual-information-based clustering technique minimizes the statistical correlation between different clusters while the traditional K-means algorithm minimizes the total variance within different clusters. Here we propose to combine the two different objectives. The new objective function is given by

$$h(s) = (1 - \lambda)\frac{1}{M} \sum_{k=1}^{K} \sum_{i=1}^{N} \left\| \boldsymbol{y}_i 1_{i,k} - \boldsymbol{c}_k \right\|^2 + \lambda \frac{2}{K^2 - K} f(s), \tag{17}$$

where $0 \leq \lambda \leq 1$ is a weight factor to adjust the relative importance of the two criteria; $f(s)$ is defined in (13); and $1/M$ and $2/(K^2 - K)$ are normalization constants. The procedure for solving this optimization problem is summarized as follows.

Algorithm 3 (Clustering based on combined mutual information and Euclidean distance). *The algorithm is the same as the simulated annealing algorithm introduced in Section 2 with the objective function replaced by (17) and* $\boldsymbol{c}_k$ *being the mean of the samples in the k-th cluster at each iteration.*

3.2. Clustering based on combined mutual information and fuzzy membership metrics

The fuzzy C-means method is a variation of the K-means method in which each gene $\boldsymbol{y}_i$, $i = 1, 2, \cdots, N$ has a degree of membership $u_{i,k}$ $(0 \leq u_{i,k} \leq 1)$ of belonging to each cluster k such that $\sum_{i=1}^{N} u_{i,k} = 1$, $1 \leq k \leq K$. Randomly set the initial membership matrix $\boldsymbol{U} = (u_{i,k})_{N \times K}$, $(u_{i,k} \in [0, 1])$, and calculate the centroid of each cluster $\boldsymbol{c}_k$ as

$$\boldsymbol{c}_k = \frac{\sum_{i=1}^{N} u_{i,k}^b \boldsymbol{y}_i}{\sum_{i=1}^{N} u_{i,k}^b}, \quad k = 1, \cdots, K, \tag{18}$$

where $b > 1$ ($b = 2$ in our simulations). The membership is calculated from the dataset by

$$u_{i,k} = \frac{\left\| \boldsymbol{y}_i - \boldsymbol{c}_k \right\|^{\frac{1}{1-b}}}{\sum_{j=1}^{K} \left\| \boldsymbol{y}_i - \boldsymbol{c}_j \right\|^{\frac{1}{1-b}}}, \quad i = 1, \cdots, N, \ k = 1, \cdots, K. \tag{19}$$

Then the objective function associated with a particular partition s is

$$g(s) = \sum_{i=1}^{N} \sum_{k=1}^{K} u_{i,k} \left\| \boldsymbol{y}_i - \boldsymbol{c}_k \right\|^2. \tag{20}$$

The fuzzy C-means algorithm for clustering is as follows. After determining the initial $u_{i,k}$, $\boldsymbol{c}_k$ and $g(s)$, repeat (19), (18), and (20) until there is no more change in $u_{i,k}$ or $g(s)$. Denote $\hat{k}_i = \arg\max_{1 \le k \le K} u_{i,k}$. We finally assign $\boldsymbol{y}_i$ to the $\hat{k}_i$-th cluster.

Here we propose to combine the two metrics of mutual information and fuzzy membership. The new objective function is given by

$$h(s) = (1 - \lambda) \frac{1}{M} \sum_{i=1}^{N} \sum_{k=1}^{K} u_{i,k} \left\| \boldsymbol{y}_i - \boldsymbol{c}_k \right\|^2 + \lambda \frac{2}{K^2 - K} f(s), \tag{21}$$

where $0 \le \lambda \le 1$ is a weight factor to adjust the relative importance of the two criteria. There are two kinds of parameters affecting the objective function value, the fuzzy membership values $u_{i,k}$, $i = 1, \cdots, N$, $k = 1, \cdots, K$, and the partition s. Note that here we keep N instead of K parallel partitions.

Algorithm 4 (Clustering based on combined mutual information and fuzzy membership).

- *Initialization: Set initial temperature* $T = 1$. *Randomly select the initial partition* s_0, *and compute the cost* $h(s_0)$. *Randomly set initial membership* $u_{i,k}$, $1 \le i \le N$, $1 \le k \le K$ *such that* $\sum_{i=1}^{N} u_{i,k} = 1$.
- *Repeat*
  - *for* $l = 1, \cdots, 100$
    - *for* $1 \le i \le N$
      - *generate* $s_i^{(1)}$ *from* $s^{(0)}$; *//Assume that gene* i *was originally* $(\star)$
      - *// assigned to cluster* r. *We then random pick a cluster label* j ($j \ne i$),
      - *// and assign gene* i *to cluster* j, *and exchange the values of* $u_{i,r}$ *and* $u_{i,j}$.
      - $\boldsymbol{c}_k \leftarrow \sum_{i=1}^{N} (u_{i,k})^b \boldsymbol{y}_i / \sum_{i=1}^{N} (u_{i,k})^b$ *(for all* $k = 1, \cdots, K$*);*
      - *compute* $h(s_i^{(1)})$;
      - $u_{i,k} \leftarrow \left\| \boldsymbol{y}_i - \boldsymbol{c}_k \right\|^{\frac{1}{1-b}} / \sum_{j=1}^{K} \left\| \boldsymbol{y}_i - \boldsymbol{c}_j \right\|^{\frac{1}{1-b}}$;
      - *if* $h(s_i^{(1)}) \le h(s^{(0)})$ *then accept* $s_i^{(1)}$;
      - *elseif* $\exp\left(\frac{-\left[h(s_i^{(1)}) - h(s^{(0)}) \right]}{T} \right) > u \sim \mathcal{U}_{[0,1)}$ *then accept* $s_i^{(1)}$;
      - *else accept* $s^{(0)}$; *// The accepted partition denoted by* $s_i^{(1)}$.
    - *endfor*
    - $s^{(0)} = \arg\min_{1 \le i \le N} h(s_i^{(1)})$.
  - *endfor*
  - $T \leftarrow 0.85T$;
- *until* $T \le 0.001$.

Note that in step $(\star)$, we generate $s_i^{(1)}$ from $s^{(0)}$.

4. CLUSTERING PERFORMANCE ON SIMULATED DATA

In this section, we test the performance of several clustering algorithms using simulated data. The algorithms under consideration include

- K-means algorithm;
- Fuzzy C-means algorithm;
- Clustering algorithm based on mutual information (MI) minimization discussed in Section 2;
- Clustering algorithm based on the combined metric of mutual information and Euclidean distance (MIK) discussed in Section 3.1;
- Clustering algorithm based on the combined metric of mutual information and fuzzy membership distance (MIF) discussed in Section 3.2;
- Hierarchical clustering algorithm: single linkage clustering algorithm;
- Biclustering algorithm (a node-deletion algorithm proposed by Chen and Church [2000]).
- Threshold clustering algorithm based on pairwise mutual information (PMI) explained in Section 2.

Numerous cluster measures based on the sample points have been proposed (Halkidi *et al.*, 2001; Jain *et al.*, 1999). Many of these are based on spatial separation, an exception being the figure of merit (FOM), which is based on the consistency of clusters when leaving a point out (Yeung *et al.*, 2001). Since our interest is solely with algorithm accuracy, in this paper we measure performance by the percentage of points placed into correct clusters (Dougherty *et al.*, 2002). Performance analysis on synthetic data is critical because only in this way do we have ground truth (true clusters) from which to measure performance deviation.

Example 1

In this example, we assume there are four binary random variables x_1, x_2, x_3, x_4 such that their joint distribution satisfies

$$p(x_1, x_2, x_3, x_4) = p(x_1, x_2)\, p(x_3, x_4), \tag{22}$$

where $p(\cdot)$ follows a Bernoulli distribution. Hence, the two clusters are (x_1, x_2) and (x_3, x_4). The probabilities used in the Bernoulli distribution are $1/2^n$ for each state, where n is the number of variables. In the first simulation, we vary the sample size M and perform the cluster analysis using the above eight algorithms. The results are the average of 100 simulations. The value of λ is set as 0.5 empirically in this paper. Since the mutual information estimation will become imprecise with an increasing number of gene variables, λ should become small to decrease the effect from the imprecise mutual information. Table 1 and Fig. 1 show the clustering results using different sample sizes. It is seen that the MI method always outperforms the fuzzy C-means, the K-means, the linkage, the biclustering, and the pairwise MI methods. The combined mutual-information and fuzzy membership-based clustering algorithm has the best clustering accuracy. The MIK method has similar performance as the MI method. The MI-based clustering methods become more accurate as the sample size increases, whereas the fuzzy C-means, the K-means, the linkage,

TABLE 1. CLUSTERING RESULTS FOR EXAMPLE 1 FOR DIFFERENT SAMPLE SIZES

Sample size (M)	*Clustering algorithms*							
	Fuzzy	*MI*	*MIK*	*MIF*	*K-means*	*PMI*	*Linkage*	*Biclustering*
10	0.7225	0.7600	0.7500	0.7825	0.7175	0.7175	0.7150	0.7000
30	0.7425	0.7850	0.7950	0.8225	0.7025	0.7750	0.7325	0.7400
50	0.7525	0.8375	0.8275	0.8550	0.6925	0.8200	0.7550	0.7335
80	0.7570	0.8400	0.8400	0.8650	0.6975	0.8300	0.7625	0.7475
100	0.7600	0.8625	0.8325	0.8725	0.7175	0.8500	0.7675	0.7550

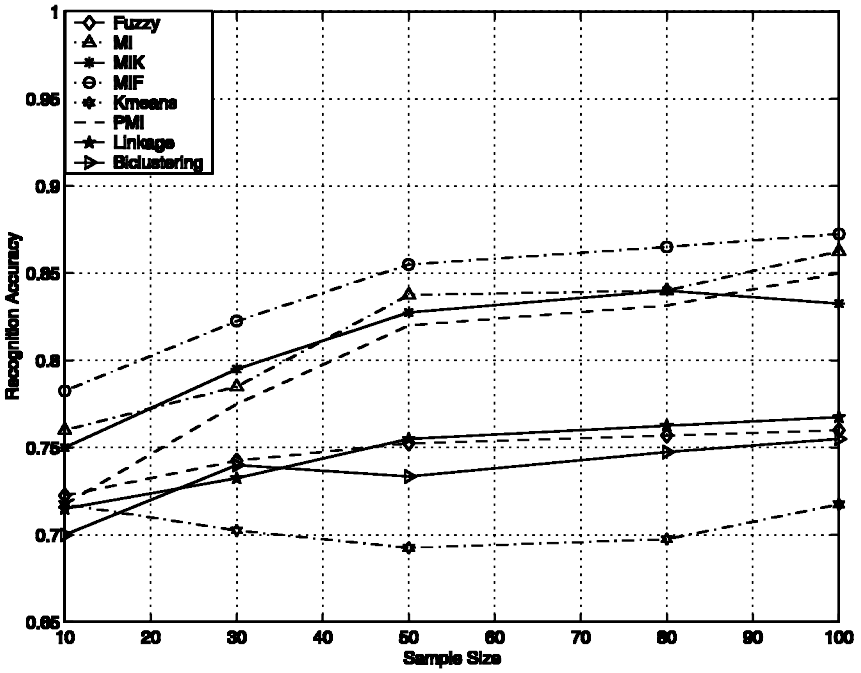

FIG. 1. The recognition accuracy comparisons of eight clustering algorithms for example 1. The x-axis denotes the sample size, and the y-axis denotes the percentage of correct clustering.

and the biclustering algorithm are insensitive to the sample size. Note that the biclustering algorithm is the algorithm 1 (single node deletion) proposed by Chen and Church (2000), where the parameter δ (maximum acceptable mean squared residue score) is set as 0.1.

Next, we fix the sample size to $M = 30$ and compare the seven clustering algorithms (not including the biclustering algorithm since it is for two-cluster clustering) under different numbers of variables and clusters as in Table 2. The data are generated according to

$$p(\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_K) = p(\boldsymbol{X}_1)p(\boldsymbol{X}_2)\cdots p(\boldsymbol{X}_K), \tag{23}$$

where $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_K$ are the K clusters. The MI method outperforms the fuzzy C-means, the K-means, the linkage, and the pairwise MI methods. The MIK method has a similar performance as the MI method. Again, the combined MIF clustering algorithm has the best performance.

Note that algorithm performances degrade substantially as the number of genes increases. This is to be expected and can be significantly rectified by replicating experiments (Dougherty *et al.*, 2002). The degree of degradation depends on the distributions governing the data according to (23). Performance holds up better for fuzzy C-means if the distributions are separated and their variances small and holds up better for mutual-information clustering if the mutual information within the individual distributions is high in comparison to the mutual information between individual distributions.

TABLE 2. CLUSTERING RESULTS FOR EXAMPLE 1 ($M = 30$)

(N)	(K)	*Clustering algorithms*						
No. genes	*No. clusters*	*Fuzzy*	*MI*	*MIK*	*MIF*	*K-means*	*PMI*	*Linkage*
4	2	0.7425	0.7850	0.7950	0.8225	0.7025	0.7750	0.7325
10	5	0.5280	0.6520	0.5770	0.6950	0.5190	0.4840	0.3180
50	5	0.3440	0.3540	0.4500	0.4700	0.3300	0.3020	0.2260

Example 2

In this example, associated with each gene x_i, we have M observations $\boldsymbol{x}_i \triangleq [x_i(1), x_i(2), \cdots, x_i(M)]^T$, which is the quantized version of a continuous random vector $\boldsymbol{z}_i \triangleq [z_i(1), z_i(2), \cdots, z_i(M)]^T$. The vector $\boldsymbol{z}_i$ is generated in the following way: if $\boldsymbol{x}_i$ belongs to the k-th cluster, then

$$\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{24}$$

where $\boldsymbol{\mu}_i \triangleq [\mu_{i1}, \mu_{i2}, \cdots, \mu_{iM}]^T$ is called a *template* (Dougherty *et al.*, 2002) and

$$\boldsymbol{\Sigma}_k \triangleq \begin{bmatrix} 1 & \sigma_k^2 & \cdots & \sigma_k^2 & \sigma_k^2 \\ \sigma_k^2 & 1 & \cdots & \sigma_k^2 & \sigma_k^2 \\ \vdots & \vdots & \ddots & \vdots & \\ \sigma_k^2 & \sigma_k^2 & \cdots & 1 & \sigma_k^2 \\ \sigma_k^2 & \sigma_k^2 & \cdots & \sigma_k^2 & 1 \end{bmatrix}_{M \times M}, \tag{25}$$

where $\sigma_k^2 < 1$. In the simulations, we set

$$\begin{bmatrix} \boldsymbol{u}_1^T \\ \boldsymbol{\mu}_2^T \\ \boldsymbol{\mu}_3^T \\ \boldsymbol{\mu}_4^T \\ \boldsymbol{\mu}_5^T \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 1 & 1 \\ 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 1 & 1 \\ 0 & 1 & \cdots & 0 & 1 \end{bmatrix}}_{\frac{M}{2}} \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 1 & 1 \\ 1 & 1 & \cdots & 1 & 1 \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 1 \end{bmatrix}}_{\frac{M}{2}} \tag{26}$$

and $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_K^2 = \sigma^2$.

First, the clustering performance under different numbers of genes and clusters is given in Table 3, for $\sigma^2 = 0.9$ and $M = 30$. In the first case, where there are $N = 4$ genes and $K = 2$ clusters, the performance of the MI method is much better than the fuzzy C-means, the linkage, and the K-means methods, and the MIF method has the best performance. With an increased number of variables and clusters, the MI method is slightly better than the fuzzy C-means, but the MIF method still has the best performance. The K-means, the linkage, and the PMI methods have the worst performances.

In Table 4 and Fig. 2, the clustering performance under different values of σ^2 is shown with $N = 4$, $K = 2$, and $M = 30$. When $\sigma^2 = 0$, the fuzzy C-means method has the best performance and the MI method is worse than the fuzzy C-means, the linkage, the biclustering, the MIK, and the MIF methods. With an increased σ^2, the performance of the MI method becomes better, and the fuzzy C-means method becomes worse. When $\sigma^2 \geq 0.3$, the MI method outperforms the fuzzy C-means. It is interesting to note that the combined MIF clustering algorithm has the best performance except for case $\sigma^2 = 0$. When $\sigma^2 = 0$, the data are completely uncorrelated, and therefore the mutual information criterion is not effective in clustering them.

Next, the clustering performance under different numbers of genes and clusters is given in Table 5, for $\sigma^2 = 0.1$ and $M = 30$. In the first case, where there are $N = 4$ genes and $K = 2$ clusters, the performance

TABLE 3. CLUSTERING RESULTS FOR EXAMPLE 2 ($\sigma^2 = 0.9$, $M = 30$)

N	K	*Clustering algorithms*						
No. genes	*No. clusters*	*Fuzzy*	*MI*	*MIK*	*MIF*	*K-means*	*PMI*	*Linkage*
4	2	0.7525	0.9150	0.7900	0.9950	0.7650	0.7075	0.7535
10	5	0.6000	0.6100	0.5500	0.6400	0.5800	0.5200	0.3600
50	5	0.3460	0.4631	0.4200	0.4800	0.3320	0.2920	0.2494

TABLE 4. CLUSTERING RESULTS FOR EXAMPLE 2 ($K = 2, M = 30, N = 4$)

	Clustering algorithms							
σ^2	*Fuzzy*	*MI*	*MIK*	*MIF*	*K-means*	*PMI*	*Linkage*	*Biclustering*
0.0	0.9175	0.7500	0.8375	0.8425	0.6925	0.6850	0.8850	0.7600
0.1	0.8350	0.7425	0.7975	0.8775	0.7475	0.6800	0.8325	0.7050
0.2	0.8125	0.7750	0.7975	0.8925	0.7450	0.6875	0.8125	0.6925
0.3	0.8075	0.8100	0.7900	0.8975	0.7675	0.6825	0.8100	0.6850
0.4	0.8050	0.8225	0.7875	0.9050	0.7775	0.6850	0.7850	0.6850
0.5	0.7800	0.8250	0.7825	0.9075	0.7525	0.6825	0.7825	0.6700
0.6	0.7800	0.8450	0.7825	0.9075	0.7500	0.6675	0.7750	0.6675
0.7	0.7725	0.8525	0.7850	0.9275	0.7525	0.6775	0.7675	0.6675
0.8	0.7700	0.9000	0.7950	0.9375	0.7575	0.7025	0.7625	0.6550
0.9	0.7525	0.9150	0.7900	0.9450	0.7650	0.7075	0.7535	0.6475
0.95	0.7125	0.9200	0.8050	0.9500	0.7350	0.7175	0.7500	0.6450

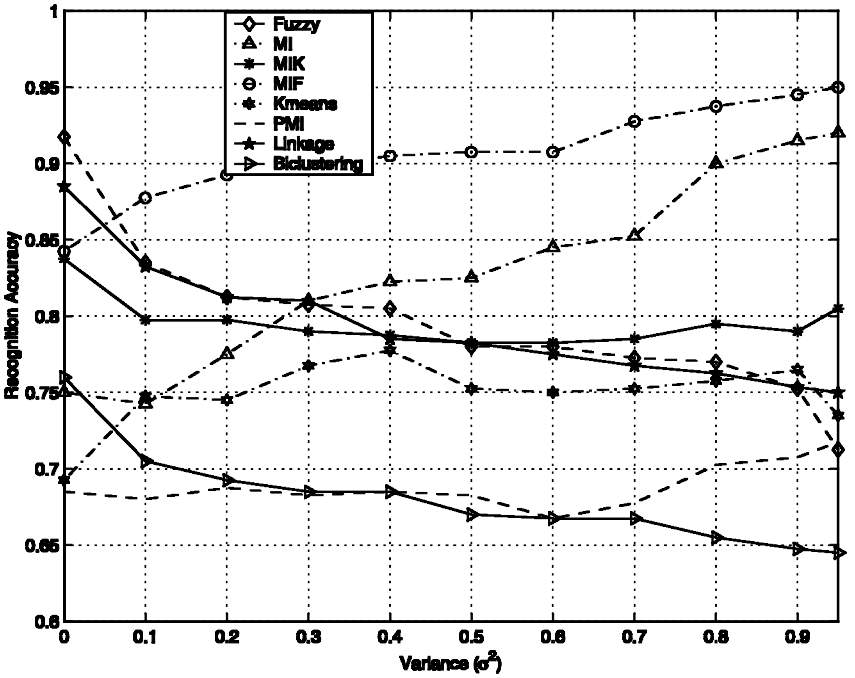

FIG. 2. The recognition accuracy comparisons of eight clustering algorithms for example 2 with sample size being 30. The x-axis denotes datasets with different variances defined in (25), and the y-axis denotes the percentage of correct clustering.

TABLE 5. CLUSTERING RESULTS FOR EXAMPLE 2 ($\sigma^2 = 0.1, M = 30$)

N	*K*	*Clustering algorithms*						
No. genes	*No. clusters*	*Fuzzy*	*MI*	*MIK*	*MIF*	*K-means*	*PMI*	*Linkage*
4	2	0.8350	0.7425	0.7975	0.8775	0.7475	0.6800	0.8325
10	5	0.6600	0.5700	0.5800	0.6300	0.5100	0.4000	0.3430
50	5	0.4000	0.3380	0.3780	0.3800	0.4440	0.3580	0.2208

of the MI method is much better than the fuzzy C-means and K-means methods, and the MIF method has the best performance. With increased numbers of variables and clusters, the MI method is slightly better than the fuzzy C-means, but the MIF method still has the best performance. The K-means and the PMI methods have the worst performance.

5. EXPERIMENTAL ANALYSIS

We have applied the clustering algorithms to binarized expression data for 597 genes derived from 26 human glioma surgical tissue samples (Fuller *et al.*, 1999). The original expression data is adjusted by combining genes possessing the same binarized expression profiles (Shmulevich and Zhang, 2002). The adjusted set has 526 genes. Both the original and reduced sets are available at *gspsnap.tamu.edu/gspweb/zxb/glioma_zxb/glioma_web.htm*, along with the clusters determined by the algorithms. Owing to the size of the gene set and the computational requirements of the algorithms (in particular, simulated annealing), parallel implementations have been developed and run on the NIH Beowulf Cluster.

The effect of combining the fuzzy and MI clustering criteria can be seen in Fig. 3, which shows (a) the binary profiles for the adjusted gene set (red $= +1$, green $= 0$), (b) the fuzzy C-means clusters, and (c) the MIF clusters. Essentially, two small clusters were broken out from fuzzy C-means clusters to become new clusters in the MIF clustering. The twelve-gene and five-gene clusters are listed in Tables 6 and 7, respectively. While these new clusters were not separated by the fuzzy C-means criterion, their internal mutual information was sufficiently high relative to their mutual information with the original clusters that the combined algorithm separated them out. While the number of genes changed between the fuzzy C-means

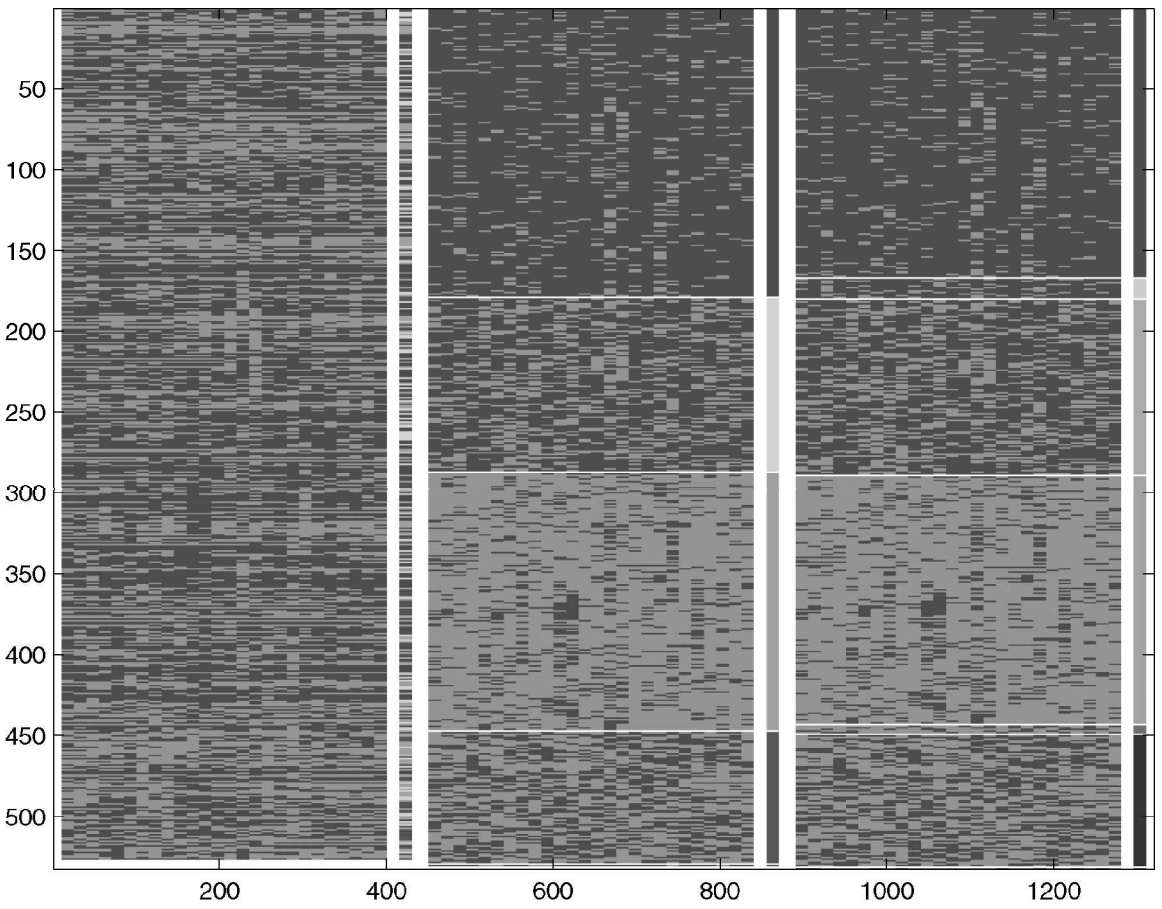

FIG. 3. Dendrogram for fuzzy C-means clustering and MIF clustering: the left one-(a) the binary profiles for the adjusted gene set (red $= +1$, green $= 0$), the middle one-(b) the fuzzy C-means clusters, and the right one-(c) the MIF clusters.

TABLE 6. THE CLUSTER WITH TWELVE GENES

Index no. of genes	*Gene description (name)*
42	Tight junction protein 1 (TJP1); zonula occludens (ZO1)
76	Retinoblastoma-associated protein 1 (RB1)
159	Gene 159 interferon regulatory factor 1 (IRF1)
189	Tumor necrosis factor receptor 1-associated death domain protein
224	Caspase 4 (CASP4); CASP5; ICH2 cysteine protease
235	DNA polymerase alpha catalytic subunit (POLA)
270	Basic transcription factor 2 44-kDa subunit (BTF2p44)
286	Transcriptional repressor NF-X1
290	Transcription factor relB; I-rel
299	45-kDa nuclear factor (NF45)
343	Interferon alpha/beta/omega receptor subunit 1
406	Corticotropin-releasing factor receptor 1 (CRFR; CRF1) and (CRHR1)

TABLE 7. THE CLUSTER WITH FIVE GENES

Index no. of genes	*Gene description (name)*
109	Activating transcription factor 2 (ATF2); CREBP1; HB16
248	DNA topoisomerase I (TOP1)
305	Homeobox protein D3 (HOXD3); HOX4A
317	90-kDa TATA (TAF3C); transcription factor TFIIIB 90-kDa subunit (TFIIIB90)
325	Transcription factor HTF4; TCF12; E-box-binding protein HEB

and MIF algorithms is small, the error decrease is not insignificant. The MIF error from the objective function of Equation (21) goes from 2.013 for the fuzzy C-means clustering to 1.084 for the MIF clustering. The clustering results of the other methods are available at *gspsnap.tamu.edu/gspweb/zxb/glioma_zxb* (user: gspweb; passwd: gsplab). The clustering results using MIK are similar to the results using the fuzzy C-means clustering method. Some genes in different clusters are listed in the above web site. The MIK error from the objective function of Equation (17) goes from 2.013 for the fuzzy C-means clustering to 0.908 for the MIK clustering. Compared with the MIF and MIK algorithms, the MI and PMI methods give quite different clustering results. This is to be expected since they depend only on mutual information, not a weighted combination of mutual information and Euclidean distance factors.

6. CONCLUSIONS

In this study, we have proposed a novel clustering strategy based on minimizing mutual information among gene clusters. Simulated annealing was employed to solve the optimization problem. Bootstrap techniques were employed to get more accurate estimation of mutual information when the data sample size is small. Moreover, we proposed to combine the mutual information criterion and traditional distance criteria, such as the Euclidean distance and the fuzzy membership metric, in designing the clustering algorithm. The performance of the new clustering methods has been compared with that of some existing methods, using both synthesized data and experimental data. The clustering algorithm based on a combined metric of mutual information and fuzzy membership has achieved the best performance.

Note that the clustering algorithm (named "cluster ensembles") of combining multiple partitions (Strehl and Ghosh, 2002) is quite different from our methods. There, given a dataset, assume there are different partitions (say, obtained by some clustering algorithms); then the cluster ensembles method is to find a partition of the dataset that is an optimal combination of the different partitions. The mutual information defined in cluster ensembles is based on different partitions, and it just depends on the numbers of elements in the clusters of the partitions.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their careful reading of our manuscript and their constructive comments. This research was supported by the National Human Genome Research Institute. X. Wang was supported in part by the U.S. National Science Foundation under grant DMS-0225692.

REFERENCES

Aarts, E.H.L., and Emile, H.L. 1989. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*, Wiley, New York.

Akutsu, T., Miyano, S., and Kuhara, S. 2000. Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics* 16, 727–743.

Ben-Dor, A., Shamir, R., and Yakhini, Z. 1999. Clustering gene expression patterns. *J. Comp. Biol.* 6(3–4), 281–297.

Bittner, M., Meltzer, P., Khan, J., Chen, Y., Jiang, Y., Seftor, E., Hendrix, M., Radmacher, M, Simon, R., Yakhini, Z., Ben-Dor, A., Dougherty, E., Wang, E., Marincola, F., Gooden, C., Lueders, J., Glatfelter, A., Pollock, P., Gillanders, E., Leja, A., Dietrich, K., Beaudry, C., Berrens, M., Alberts, D., Sondak, V., Hayward, N., and Trent, J.M. 2000. Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature* 406, 536–540.

Brown, M.P.S., Grundy, W.N., Cristianini, N., Sugnet, C., Furey, T.S., Ares, M., and Haussler, D. 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA* 97, 262–267.

Butte, A.J., and Hohane, I.S. 2000. Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurements. *Pac. Symp. Biocomputing* 4.

Chen, Y., and Church, G.M. 2000. Biclustering of expression data. *Proc. 8th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB).*

Chen, Y., Dougherty, E.R., and Bitter, M.L. 1997. Ratio-based decisions and the quantitive analysis of cDNA microarray images. *J. Biomedical Optics* 2(4), 364–374.

Claverie, J.M. 1998. Computational methods for the identification of differential and coordinated gene expression. *Human Mol. Genet.* 8(10), 1821–1832.

Cover, T.M., and Thomas, J.A. 1991. *Elements of Information Theory*, Wiley, New York.

Debouck, C., and Goodfellow, P.N. 1999. DNA microarrays in drug discovery and development. *Nature Genet.* 21(1), 48–50.

Dougherty, E.R., Barrera, J., Brun, M., Kim, S., Cesar, R.M., Chen, Y., Bittner, M., and Trent, J.M. 2002. Inference from clustering: Application to gene-expression time series. *J. Comp. Biol.* 9(1).

Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. 1998. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 14863–14868.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. 2000. Using Baysian network to analyze expression data. *J. Comp. Biol.* 7, 601–620.

Friedman, N., Nachman, I., and Pe'er, D. 1999. Learning Bayesian network structure from massive data sets: The "Sparse Candidate" algorithm. *Proc. 15th Conf. on Uncertainty in Artificial Intelligence (UAI).*

Fuller, G.N., Rhee, C.H., Hess, K.R., Caskey, L.S., Wang, R., Bruner, J.M., Yung, W.K.A., and Zhang, W. 1999. Reactivation of insulin-like growth factor binding protein 2 expression in glioblastoma multiforme: A revelation by parallel gene expression profiling. *Cancer Res.* 59, 4228–4232.

Ghosh, D., and Chinnaiyan, A.M. 2002. Mixture modelling of gene expression data from microarray experiments. *Bioinformatics* 18, 275–286.

D'Haeseleer, P., Liang, P., Fuhrman, S., and Somogyi, R. 2000. Genetic network inference: From co-expression clustering to reverse engineering. *Bioinformatics* 16(8), 707–726.

D'Haeseleer, P., Wen, X., Fuhrman, S., and Somogyi, R. 1999. Linear modeling of mRNA expression levels during CNS development and injury. *Pac. Symp. Biocomputing* 4, 41–52.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. 2001. On clustering validation techniques. *Intelligent Information Systems* 17, 107–145.

Heyer, L.J., Kruglyak, S., and Yooseph, S. 1999. Exploying expression data: Identification and analysis of coexpressed genes. *Genome Res.* 9, 1106–1115.

Horimoto, K., and Toh, H. 2001. Statistical estimation of cluster boundaries in gene expression profile data. *Bioinformatics* 17, 1143–1151.

Huang, S. 1999. Gene expression profiling, genetic networks, and cellular states: An integrating concept for tumorigenesis drug biology. *J. Mol. Med.* 77, 469–480.

Ingber, L., and Rosen, B. 1992. Genetic algorithms and very fast simulated reannelaing: A comparison. *Math. Comp. Modelling* 16(16), 87–100.

Iyer, V.R., Eisen, M.B., Ross, D.T., Schuler, G., Moore, T., Lee, J.C.F., Trent, J.M., Staudt, L.M., Hudson, J., Boguski, M.S., Lahkari, D., Shalon, D., Botstein, D., and Brown, P.O. 1999. The transcriptional program in the response of human firbroblasts to serum. *Science* 283(5398), 83–87.

Jain, A.K., Murty, M.N., and Flynn, P.J. 1999. Data clustering: A review. *ACM Computing Surveys* 3(31), 264–323.

Kauffman, S.A. 1993. *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York.

Lukashin, A.V., and Fuchs, R. 2001. Analysis of temporal gene expression profiles: Clustering by simulated annealing and determining the optimal number of clusters. *Bioinformatics* 17(17), 405–414.

Michaels, G., Carr, D.B., Askenazi, M., Fuhrman, S., Wen, X., and Somogyi, R. 1998. Cluster analysis and data visulaization of large-scale gene expression data. *Pac. Symp. Biocomputing* 3, 42–53.

Robinson, J. 1988. Theoretical comparison of bootstrap confidence intervals. *Ann. Statistics* 16, 976–977.

Shmulevich, I., Dougherty, E.R., Kim, S., and Zhang, W. 2002. Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics* 18(1).

Shmulevich, I., and Zhang, W. 2002. Binary analysis and optimization-based normalization of gene expression data. *Bioinformatics* 18, 555–565.

Strehl, A., and Ghosh, J. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. Machine Learning Res.* 3, 583–617.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E.S., and Golub, T.R. 1999. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA* 96, 2907–2912.

Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., and Church, G.M. 1999. Systematic determination of genetic network architecture. *Nature Genet.* 22, 281–285.

Xing, E.P., and Karp, R.M. 2001. CLIFF: Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics* 17, s306–s315.

Yeung, K.Y., Fraley, C., Murua, A., Raftery, A.E., and Ruzzo, W.L. 2001. Model-based clustering and data transformations for gene expression data. *Bioinformatics* 17, 977–987.

Yeung, K.Y., Haynor, D.R., and Ruzzo, W.L. 2001. Validating clustering for gene expression data bioinformatics. *Bioinformatics* 17, 309–318.

Zoubir, A.M., and Boashash, B. 1998. The bootstrap and its application in signal processing. *IEEE Signal Processing Magazine* 15, 56–76.

Address correspondence to:
Edward R. Dougherty
Department of Electrical Engineering
214 Zachry Engineering Center
Texas A&M University
College Station, TX 77843

E-mail: edward@ee.tamu.edu