JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 9, Number 1, 2002
Mary Ann Liebert, Inc.
Pp. 105–126

# Inference from Clustering with Application to Gene-Expression Microarrays

EDWARD R. DOUGHERTY,[1] JUNIOR BARRERA,[2] MARCEL BRUN,[2]
SEUNGCHAN KIM,[1] ROBERTO M. CESAR,[2] YIDONG CHEN,[3]
MICHAEL BITTNER,[3] and JEFFREY M. TRENT[3]

## ABSTRACT

**There are many algorithms to cluster sample data points based on nearness or a similarity measure. Often the implication is that points in different clusters come from different underlying classes, whereas those in the same cluster come from the same class. Stochastically, the underlying classes represent different random processes. The inference is that clusters represent a partition of the sample points according to which process they belong. This paper discusses a model-based clustering toolbox that evaluates cluster accuracy. Each random process is modeled as its mean plus independent noise, sample points are generated, the points are clustered, and the clustering error is the number of points clustered incorrectly according to the generating random processes. Various clustering algorithms are evaluated based on process variance and the key issue of the rate at which algorithmic performance improves with increasing numbers of experimental replications. The model means can be selected by hand to test the separability of expected types of biological expression patterns. Alternatively, the model can be seeded by real data to test the expected precision of that output or the extent of improvement in precision that replication could provide. In the latter case, a clustering algorithm is used to form clusters, and the model is seeded with the means and variances of these clusters. Other algorithms are then tested relative to the seeding algorithm. Results are averaged over various seeds. Output includes error tables and graphs, confusion matrices, principal-component plots, and validation measures. Five algorithms are studied in detail: K-means, fuzzy C-means, self-organizing maps, hierarchical Euclidean-distance-based and correlation-based clustering. The toolbox is applied to gene-expression clustering based on cDNA microarrays using real data. Expression profile graphics are generated and error analysis is displayed within the context of these profile graphics. A large amount of generated output is available over the web.**

**Key words:** clustering, gene expression, microarray.

[1]Department of Electrical Engineering, Texas A&M University, College Station, TX 77840.
[2]Departamento de Ciencia de Computacao, Universidade de Sao Paulo, Sao Paulo, Brazil.
[3]National Human Genome Research Institute, National Institutes of Health, Bethesda, MD 20892-4470.

## INTRODUCTION

Clustering is a popular way of analyzing data. There are many algorithms and, given a data set, these algorithms may or may not be in close agreement (Jain *et al*., 1999). Since a clustering algorithm forms clusters, there are various validity measures for the strengths of the clusters formed from a given sample. These measures provide statistical support for the existence of clusters within the sample (Theodoridas and Koutroumbas, 1999; Duda *et al*., 2000). It is not unusual for a researcher to look at some visualization of the data, decide that there are valid clusters, apply a clustering algorithm, and then infer some similarity among the elements within the clusters. This inference concerns us here. The data comprise a sample from a set of random vectors. An inference can only be meaningful if it pertains to the random vectors. Specifically, if clustering is being used to group observations, then it is implicit in the formation of the clusters that they represent an estimate of some set of classes that partition the set of random vectors from which the sample has been drawn. Our purpose is to examine the precision of sample-based clustering relative to population inference. To accomplish this end, we will postulate a model in which it is assumed that the full set of random vectors is partitioned into *congruency classes* and a clustering algorithm estimates these congruency classes by forming clusters from the sample data. The key issue is the degree to which the sample clusters estimate the underlying congruency classes. This is measured by the expected number of misclassifications.

The precision of estimation will depend on several factors: the separation between congruency classes, experimental variability, and the number of sample replications. Given the congruency classes, precision declines with increasing variance and improves with an increasing number of replications. Here we provide a model-based simulation approach to examine the quality of clustering algorithms relative to inference. Beginning with a set of congruency classes, we examine a number of clustering algorithms across a range of noise variances and sample replications. More generally, the method can be used to estimate beforehand the number of replications necessary to achieve a desired degree of inference precision. It also can be used to check the accuracy of a given clustering by estimating the model parameters from the observed data and then using the model to measure clustering precision under those parameters.

The general analysis will be applied to clustering based on cDNA microarrays (Ben-Dor *et al*., 1999; Bittner *et al*., 2000; Eisen *et al*.; 1998; Spellman *et al*., 1998; Tamayo *et al*., 1999). Each microarray provides expression measurements for up to several thousand genes. One way of looking at expression data from microarrays is to track expression levels of each gene across discrete time points 1, 2, . . . , $n$, so that there are $n$ measurements corresponding to each gene. Time-series clustering groups together genes whose expression levels exhibit similar behavior through time. Similarity indicates possible co-regulation. Another way to use the expression data is to take expression profiles over various tissue samples and then cluster these samples based on the expression levels for each sample. This approach offers the potential to discriminate pathologies based on their differential patterns of gene expression. In either application, because expression measurements exhibit random behavior across various samples, a gene's expression levels are modeled stochastically. The application on which we will focus is time-series clustering. An expression time series will be modeled as a time-expression template defining the congruency class to which a gene belongs plus random noise. We reiterate that, although we focus on a time-series application, the evaluation analysis of clustering algorithms discussed in the paper is applicable to clustering in general.

This paper proposes a model for measuring clustering precision, applies the model to evaluate various clustering algorithms, and describes the fundamentals of a cluster-analysis toolbox that has been developed. Algorithm performance will be analyzed relative to increasing numbers of replications. Since replications for microarray applications are severely limited relative to the enormous number of variables being examined, a critical issue is quickly improving performance for small numbers of replications. The paper closes with some theoretical considerations relating the overall approach to pattern recognition.

The output of the inference analysis discussed in this paper is extensive, both numerically and graphically. A website has been set up to view the output of the analysis for several examples and provide supplementary information, including documentation for various statistical techniques employed in the analysis (see Appendix for website access and description).

## CONGRUENCY CLASS MODEL

Before giving a mathematical description of the congruency class model, we motivate it in terms of identifying gene expression patterns that indicate possible co-regulation. Ignoring random experimental error, the temporal expression profile of a gene can be viewed as a stochastic process. The specific measurements for each observation of the profile will differ depending on both internal and external factors regulating the expression level, including the conditions to which the cell lines are being subjected. Formally, we can write the observed expression as

$$X(t) = \mu_X(t) + X_r(t) \tag{1}$$

where $X_r(t)$ is the stochastic displacement of $X(t)$ from its deterministic mean $\mu_X(t)$. Since the mean of the stochastic process is its expectation, the expectation of the displacement is 0.

We must now define the meaning of a necessary condition for two genes that are co-regulated. If $X(t)$ and $Y(t)$ are the expression profiles of genes $g_x$ and $g_y$, respectively, one way to proceed is to define $g_x$ and $g_y$ to be co-regulated, denoted $g_x \sim g_y$, if $\|X(t) - Y(t)\|$ is sufficiently small, where the double bars indicate some norm between two stochastic functions. The matter can be greatly simplified if we do not consider the random displacement when defining co-regulation, but only the mean. In this case, we could define $g_x \sim g_y$ if $\|\mu_X(t) - \mu_Y(t)\|$ is sufficiently small. This means that the genes are considered to be candidates for co-regulation if their means are sufficiently close relative to the norm. An added benefit of this approach is that, if we include experimental noise $N(t)$, which we assume to have zero mean and be independent of the expression profile, then Equation 1 becomes

$$X(t) = \mu_X(t) + X_r(t) + N(t) \tag{2}$$

and the addition of the additive noise does not effect our definition of co-regulation candidacy because the mean of the process remains the same. An obvious possible choice for the norm is to define $\|\mu_X(t) - \mu_X(t)\|$ to be the maximum difference between $\mu_x(t)$ and $\mu_Y(t)$. In practice, if we model the problem in such a way that the number of mean functions is much less than the total number of genes, meaning there are many genes per mean, then it is reasonable to define $\|\mu_X(t) - \mu_Y(t)\|$ to be "sufficiently small" if it is 0. This means that $g_x \sim g_y$ if the means are identical. In sum, a congruency class is composed of a set of genes all possessing the same mean.

From a modeling perspective, stochastic gene profiles defined according to Equation 2 depend on selections of potential mean functions and a model for the random part $X_r(t) + N(t)$. We will lump both stochastic terms together into a single noise term to arrive at the linear model

$$X(t) = \mu_X(t) + N(t). \tag{3}$$

This simplification has two justifications. First, at present there is insufficient appreciation of the displacement characteristics for microarray data to accurately separate the noise components. Second, it simplifies the analysis and simulation machinery. Even were there a satisfactory characterization of the stochastic nature of expression profiles themselves, it would still be useful to use the simplified model of Equation 3 to estimate the number of misclassifications in a clustering algorithm. As will be seen, the model provides simple spherical geometry for the clusters, affords easy visualization, is in accord with distance-based clustering algorithms, and is describable with a small number of parameters that can be easily estimated from real data.

Equation 3 is quantized to a finite number of time points to form the congruency class model. We assume there are $m$ deterministic functions of discrete time $1, 2, \ldots, n$. Each function, called a *template*, corresponds to a mean function. A template is defined by an $n$-dimensional vector. There are $m$ template vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m$, each of the form $\mathbf{u}_k = (u_{k1}, u_{k2}, \ldots, u_{kn})$, where $u_{kj}$ is the value of the $k$th template at time point $j$. Congruency classes are defined by randomizations of $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m$. For $k = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, let $N_{kj}$ be a random variable possessing a Gaussian distribution with mean 0 and

variance $\sigma_k^2$. Assume that the collection $\{N_{kj}\}$ is probabilistically independent. The $k$th *congruency class*, $U_k$, is defined by the random vector

$$\mathbf{X}_k = \begin{pmatrix} X_{k1} \\ X_{k2} \\ \vdots \\ X_{kn} \end{pmatrix} = \begin{pmatrix} u_{k1} + N_{k1} \\ u_{k2} + N_{k2} \\ \vdots \\ u_{kn} + N_{kn} \end{pmatrix} = \mathbf{u}_k + \mathbf{N}_k \tag{4}$$

where $\mathbf{N}_k = (N_{k1}, N_{k2}, \ldots, N_{kn})$, $\mathbf{X}_k$ is Gaussian with mean vector $\mathbf{u}_k$, and variance vector $\sigma_k^2 = (\sigma_k^2, \sigma_k^2, \ldots, \sigma_k^2)$. According to the model, $\mathbf{X}_k$ and $\mathbf{X}_j$ are uncorrelated if $k \neq j$.

A random vector belongs to congruency class $U_k$ if it is identically distributed to $\mathbf{X}_k$. In our particular application, each such vector corresponds to an expression time series. Hence, we will speak of genes being in congruency classes. This means that a gene is in congruency class $U_k$ if its expression profile is modeled by $\mathbf{X}_k$. According to our preceding remarks, genes $g_x$ and $g_y$ are co-regulated if they belong to the same congruency class. If there are $T$ genes altogether, then there are $m$ congruency classes $U_1, U_2, \ldots, U_m$ with $r_k$ genes in class $U_k$, $T = r_1 + r_2 + \cdots + r_m$, and $g_x \sim g_y$ if and only if they possess the same mean among $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m$.

A single experiment produces a random sample of size $r_k$ for each the congruency class $U_k$. Each is a sample of $\mathbf{X}_k$. Because there are $n$ time points, these correspond to $r_k$ points $\mathbf{x}_k^1, \mathbf{x}_k^2, \ldots, \mathbf{x}_k^{r_k}$ in $n$-dimensional space. Each sample produces $T$ points. The statistical model for the sampling is that there are $r_k$ random vectors $\mathbf{X}_k^1, \mathbf{X}_k^2, \ldots, \mathbf{X}_k^{r_k}$ identically distributed to $\mathbf{X}_k$. A single sample produces the deterministic points $\mathbf{x}_k^1, \mathbf{x}_k^2, \ldots, \mathbf{x}_k^{r_k}$. A clustering algorithm is run on the $T$ points $\mathbf{x}_1^1, \mathbf{x}_1^2, \ldots, \mathbf{x}_1^{r_1}, \mathbf{x}_2^1, \mathbf{x}_2^2, \ldots, \mathbf{x}_2^{r_2}$, $\mathbf{x}_3^1, \mathbf{x}_3^2, \ldots, \mathbf{x}_m^{r_m}$. We assume that the number of clusters is known beforehand and therefore the algorithm is preset to have $m$ clusters $C_1, C_2, \ldots, C_m$. To analyze clustering precision, we assign clusters to congruency classes. Cluster $C_j$ is assigned to a congruency class by voting: $C_k$ is assigned to congruency class $U_k$ if the number of genes in $C_j$ from $U_k$ exceeds the number from any other congruency class. In the case of ties, the congruency class is chosen randomly. If there are many misclassifications, then it is possible that different clusters may be assigned to the same congruency class. This means for error calculation that the clusters have been joined to form a single cluster. The number of misclassifications is the number of sample points assigned to the wrong congruency class. This number is a random variable dependent on the sample. The misclassification error, $\rho_n$, is defined as the number of misclassifications divided by the number of sample points. We are mainly interested in the expected misclassification error, $E[\rho_n]$, as a measure of clustering precision.

If an experiment is replicated $N$ times, then there will be $N$ measurements for each gene. We can average these and then cluster. Since each gene in congruency class $U_k$ has an expression profile probabilistically identical to $\mathbf{X}_k$, its average profile for the $N$ experiments is modeled by the sample mean $\underline{\mathbf{X}}_k$ of $\mathbf{X}_k$. Whereas the variance of each component of $\mathbf{X}_k$ is $\sigma_k^2$, the variance of each component of $\underline{\mathbf{X}}_k$ is $\sigma_k^2/N$. This means the clusters get tighter as the number of replications increases. Increasing the replications should reduce the expected number of misclassifications.

## INFERENCE ANALYSIS

The inference analysis we have developed uses the foregoing model to analyze the inferential precision of a clustering procedure in terms of a collection of congruency classes. Currently, the algorithm involves five clustering algorithms: K-means, fuzzy C-means, self-organizing maps, hierarchical Euclidean-distance-based clustering, and hierarchical correlation-based clustering (Duda *et al.*, 2000; Jain, Dubes, 1988; Jain *et al.*, 1999; Theodoridis and Koutroumbasd, 1999). These are described in the appendix. Besides misclassification error, the inference analysis also provides related statistical and graphical output to assist in appreciation of the results.

A key motivation for the inference analysis is to examine the effect of replicates on clustering precision, relative to the template vectors and variances. Thus, the algorithm provides an error graph giving the

misclassification rate as a function of the number $N$ of replicates, for $N = 1, 2, 3, 4, 5, 10, 15, 20$. An experiment with $N$ replications is called an $N$-experiment. The chosen breakdown of $N$ lets one see the effect of a small number of replicates, which may be economically and technically feasible, as well as the long-run behavior of the clustering procedure as the number of replicates grows.

To reveal the nature of the clustering errors, as well as their overall number, a confusion matrix is provided. It is defined by labeling the matched clusters and congruency classes in corresponding order, and for each $i$ and $j$ giving the number of profiles in congruency class $i$ that were placed in cluster $j$.

Since cluster dispersion is key to clustering, a two-dimensional visualization of compressed clusters is shown. Compression is done by principle component analysis (PCA). As the number $n$ of time points grows, the cluster visualization using two principle components becomes less separated, but the visualization provides a reasonably good indication of separability for moderate $n$. PCA gives an optimal projection of the $n$-component vectors into a two-dimensional space spanned by statistically determined vectors.

A common way of displaying time-series ratio data from cDNA microarrays is to list the genes vertically and time points horizontally, and then used discrete pseudo-colored squares to indicate the ratio (Eisen *et al.*, 1998). Green indicates a ratio R/G less than one, red a ratio greater than one, and increasing color intensity reflects the degree to which the ratio is displaced from one (red labels for positive values of log R/G, and green labels for negatives values). This profile display can provide visual indication of clustering according to the behavior of the sample data over time. A central purpose of the inference analysis is to see to what degree the clusters given by the sample data reflect true clustering of the underlying stochastic processes (congruency classes). For each clustering, the various profile graphic displays are provided. The first display has the expressions for the full gene set in the initial order (unordered), with the true class of each gene indicated by a color in the associated tiny column. The second display has the genes ordered by cluster (as determined by the clustering algorithm). If the clustering algorithm is hierarchical, then the third display is the dendrogram resulting from the algorithm (associated with the second display). The fourth display is like the second, but partitioned into $m$ clusters, with the correct class for each gene indicated by the color in the associated tiny column. The last display shows the mean vectors of the congruency classes (of course, correctly grouped). The colors in the tiny column to the right of the last display indicate the class, and these colors can be compared with the colors of fourth column that result from the clustering algorithm. When the algorithm is not hierarchical, the second and third displays are absent.

Clusters of sample data are commonly measured according to the *validity* of the clusters (Jain *et al.*, 1988; Theodoridis and Koutroumbasd, 1999). Validity refers to the number of clusters present in the data. If it is decided that a clustering algorithm is to determine $m$ clusters, then it will determine $m$ clusters. Is the assumption of $m$ clusters valid? Various tests have been developed in the literature. The inference analysis provides the following validity measures for the sample data: the J1, J2, and J3 criteria, and Hubert's statistics with distance matrix. These are described in the website, along with references, and they are part of the output contained in the website.

In sum, the inference analysis uses the following input data:

1. $n$, the number of time points,
2. $m$, the number of congruency classes,
3. $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m$, the template vectors,
4. $r_1, r_2, \ldots, r_m$, the numbers of time series in each congruency class,
5. $\sigma_1^2, \sigma_2^2, \ldots, \sigma_m^2$, the class variances, and
6. the desired clustering algorithm.

The inference analysis produces the following outputs:

1. template graphs compared with means of clusters from clustering algorithm,
2. error table: the number of misclassifications in terms of $N$,
3. error graph: percentage of misclassification versus $N$,
4. confusion matrix,
5. 2D compressed data plot using principle component analysis,
6. cDNA-microarray profile graphics, and
7. validation measurements.

One may wish to center, scale, or transform the data before applying clustering. This is outside the scope of the algorithm and is the prerogative of the individual researcher.

To demonstrate the methodology, we first use the five synthetic templates given in the rows of the matrix

$$\begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_4 \\ \mathbf{u}_5 \end{pmatrix} = \begin{pmatrix} 3 & 2 & 1 & 0 & -1 & -2 & -3 \\ 3 & 2 & 1 & 0 & 1 & 2 & 3 \\ 0 & 2 & 2 & 0 & -2 & -2 & 0 \\ -3 & -2 & -1 & 0 & -1 & -2 & -3 \\ -3 & -2 & -1 & 0 & 1 & 2 & 3 \end{pmatrix}. \tag{5}$$

We assume $r_k = 50$ genes for each congruency class. This represents a total of $T = 250$ vectors (gene-expression profiles). A common variance $\sigma^2$ is assumed for each class. Figures 1 and 2 show 2D-PCA plots for simulated data with increasing and decreasing class overlapping for increasing $\sigma$ and increasing $N$, respectively.

For various variances and increasing $N$, the precision of the clustering algorithms is shown for fuzzy C-means, hierarchical correlation-based clustering, K-means, self-organizing map, and hierarchical Euclidean-based clustering in Fig. 3. Curves represent averages over 50 runs of the simulation. Except for K-means, replication helps in all cases; however, there are stark performance differences. In practice, $N$ will be small, often 1. The better performance of fuzzy C-means and Euclidean-distance-based hierarchical clustering should be noted in this regard. Note that error rates for K-means clustering do not get much below 20% even for large $N$.

Table 1 provides confusion matrices for fuzzy C-means and correlation-based hierarchical clustering for the cases $\sigma = 2.0$, $N = 2$ and $\sigma = 3.0$, $N = 1$. For $\sigma = 2.0$, $N = 2$, fuzzy C-means makes very few mistakes. For $\sigma = 3.0$, $N = 1$, there are some wrong assignments of class 3 into class 1, which seems not unlikely because of template similarity. For $\sigma = 2.0$, $N = 2$, hierarchical correlation-based clustering does not perform too badly except for class 3, which is consistently confused with class 1. For $\sigma = 3.0$, $N = 1$, the situation is much worse. Profiles in classes 2 and 4 are mis-clustered about half of the time, and half of those in class 3 are mis-clustered into class 1.

To illustrate the inconsistent behavior of the K-means algorithm, we consider two simulations in which K-means produces good results when the congruency classes are somewhat dispersed ($\sigma = 3.0$, $N = 10$) and bad results when they are tightly packed ($\sigma = 0.025$, $N = 50$). These are illustrated in Fig. 4, which shows random initialization and final clusters (partition of the space) determined by K-means: a) random initialization, loosely packed; b) final clusters; c) random initialization, tightly packed; d) final clusters with 50 errors. Owing to this kind of behavior, K-means can have unsatisfactory error rates even for large $N$. The results are what would be expected for hand-selected centers. These are shown in parts e through h of the figure. To test whether these kinds of results are dependent on the particular K-means implementation employed, other K-means implementations have been checked and similar results obtained.

For the templates in Equation 5, distances between cluster centers are fixed, and different variances and numbers of replications have been considered. We now treat clustering performance as a function of distance and variance, which together determine cluster separation. To make the spatial relations transparent and to avoid PCA compression, two-point templates are used. The four parts of Fig. 5 show data plots for the five classes at $\sigma = 0.15$, $\sigma = 0.50$, $\sigma = 0.80$, and $\sigma = 1.15$, with varying distances between the templates within each part. Performance curves ($N = 1$) for K-means, fuzzy C-means, and hierarchical Euclidean-based clustering are shown in Fig. 6.

According to the model of Equation 4, the variance vector for the noise $\mathbf{N}_k$ has equal variances, $\sigma_k^2 = (\sigma_k^2, \sigma_k^2, \ldots, \sigma_k^2)$. This condition is in accord with the assumption of Equations 3 and 4 that the noise is not time dependent. In practice, it means that model clusters are spherical. This condition can be relaxed so that the variance vector takes the form of vector $\sigma_k^2 = (\sigma_{k1}^2, \sigma_{k2}^2, \ldots, \sigma_{kn}^2)$. For application, as described in the next section, this would require estimation of more parameters, $nk$ instead of $n$. Since the purpose of the model is to check the inference capability of clustering algorithms for a data set, and clustering algorithms such as K-means and fuzzy C-means depend on $n$-dimensional Euclidean distance, the spherical assumption appears reasonable. Nonetheless, one could assume unequal variances within a class. To illustrate the effects of unequal variances, we use two-point templates. In addition, only two templates, (0, 0) and (0, 3), are

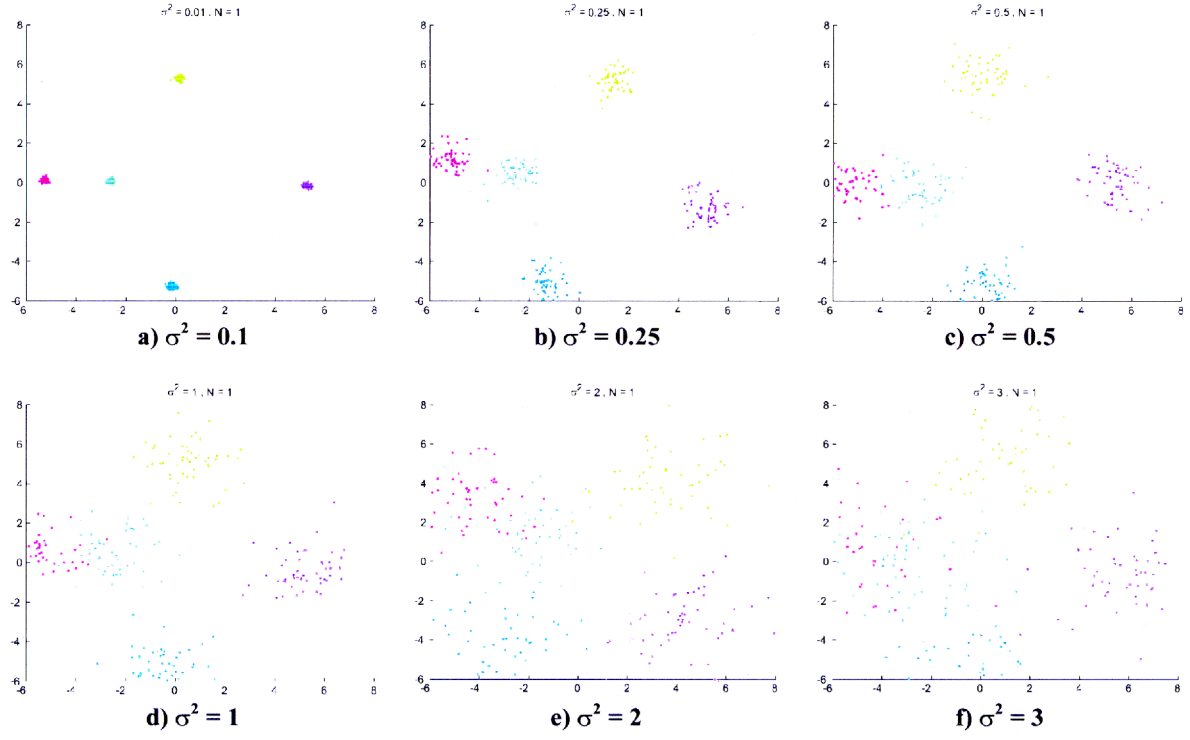



**FIG. 1.** Class overlapping for synthetic data: increasing variances.

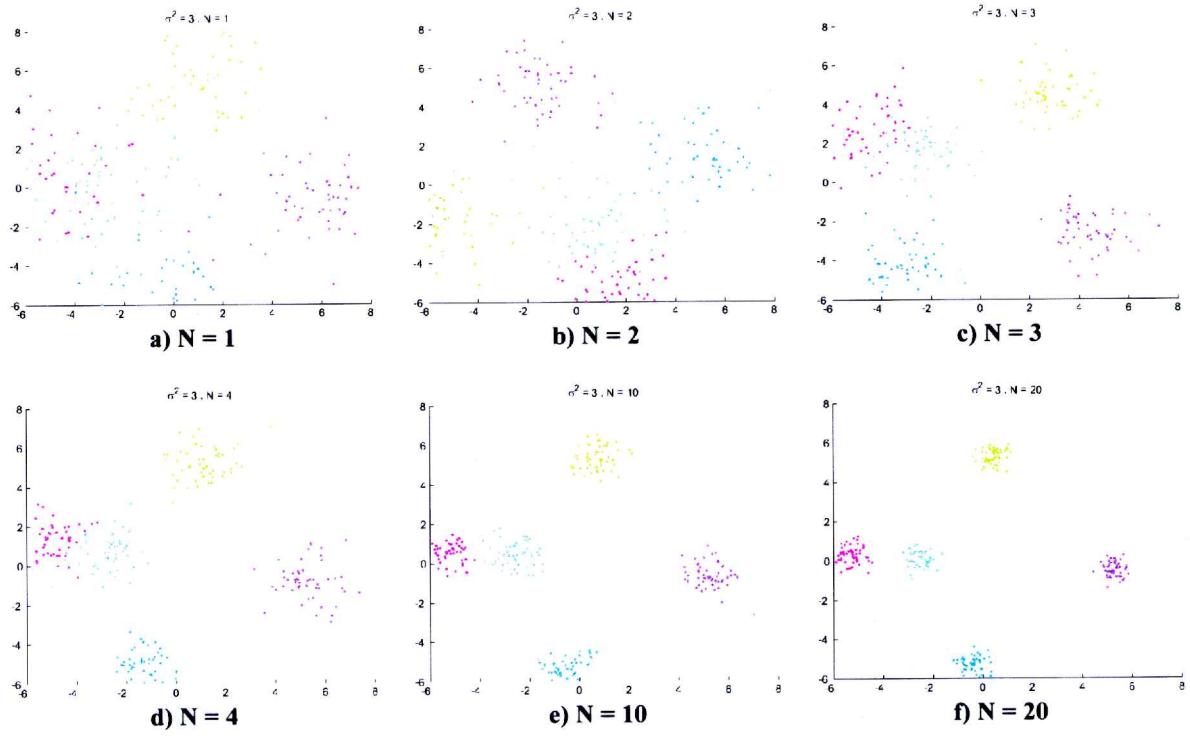



**FIG. 2.** Class overlapping for synthetic data: increasing N.

used so that the geometric effect of different variances is clear. The standard deviations used are 0.05, 0.25, 0.5, 0.75, and 1. Six pairs of covariance matrices are considered. Each yields a different clustering geometry. The six types are defined by the following pairs of covariance matrices:

$$\text{Type 1:} \quad \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_1^2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \sigma_2^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$$

$$\text{Type 2:} \quad \begin{pmatrix} 3\sigma_1^2 & 0 \\ 0 & \sigma_1^2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \sigma_2^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$$

$$\text{Type 3:} \quad \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & 3\sigma_1^2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \sigma_2^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$$

$$\text{Type 4:} \quad \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & 3\sigma_1^2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \sigma_2^2 & 0 \\ 0 & 3\sigma_2^2 \end{pmatrix}$$

$$\text{Type 5:} \quad \begin{pmatrix} 3\sigma_1^2 & 0 \\ 0 & \sigma_1^2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 3\sigma_2^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$$

$$\text{Type 6:} \quad \begin{pmatrix} 3\sigma_1^2 & 0 \\ 0 & \sigma_1^2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \sigma_2^2 & 0 \\ 0 & 3\sigma_2^2 \end{pmatrix}$$

Sample data for types 3 through 6 are shown in parts a through d of Fig. 7, respectively. Each part shows three cases, for the variances $\sigma_1 = \sigma_2 = 0.5$, $\sigma_1 = \sigma_2 = 0.75$, and $\sigma_1 = \sigma_2 = 1$. Parts a through d of Fig. 8 show performance curves for types 3 through 6, respectively, with each part containing curves for K-means, fuzzy C-means, and hierarchical Euclidean-based clustering. Sample-data plots and performance curves for types 1 and 2 have not been shown because type 1 involves equal variances and performance for type 2 is very similar to that of type 1. These are shown in the website. It is interesting to note that, when there are only two classes, we do not observe the inconsistent behavior of the K-means algorithm.

## APPLICATION

Application of the inference analysis to real data requires estimation of the model parameters from the data. Once these parameters have been estimated, the algorithm can be run to predict the expected numbers of errors based on the various algorithms and the number of replications. Intuitively, the algorithm gives the number of errors one would expect given the data. The problem here is that the algorithm requires the means and variances for the congruency classes, and the raw data does not include congruency classes. We proceed by applying a clustering algorithm to the raw data to form congruency classes with which to seed the algorithm. For instance, we might seed the model by applying fuzzy C-means to form *seed* congruency classes, and then apply clustering algorithms to the model based on those classes.

Suppose there are $q$ clustering algorithms $A_1, A_2, \ldots, A_q$, and $m$ congruency classes. A selected clustering algorithm $A_k$ is used to form $m$ clusters, which are then identified as *seed* congruency classes, $U_{k1}, U_{k2}, \ldots, U_{km}$. The model is seeded by computing the means and variances from $U_{k1}, U_{k2}, \ldots, U_{km}$. The inference analysis can then be run using the various clustering algorithms $A_1, A_2, \ldots, A_q$. Depending on the number $N$ of replications, the algorithm will produce expected error rates of $R_{k1}(N), R_{k2}(N), \ldots, R_{kq}(N)$ corresponding to $A_1, A_2, \ldots, A_q$, respectively. These error rates correspond to seeding by algo-

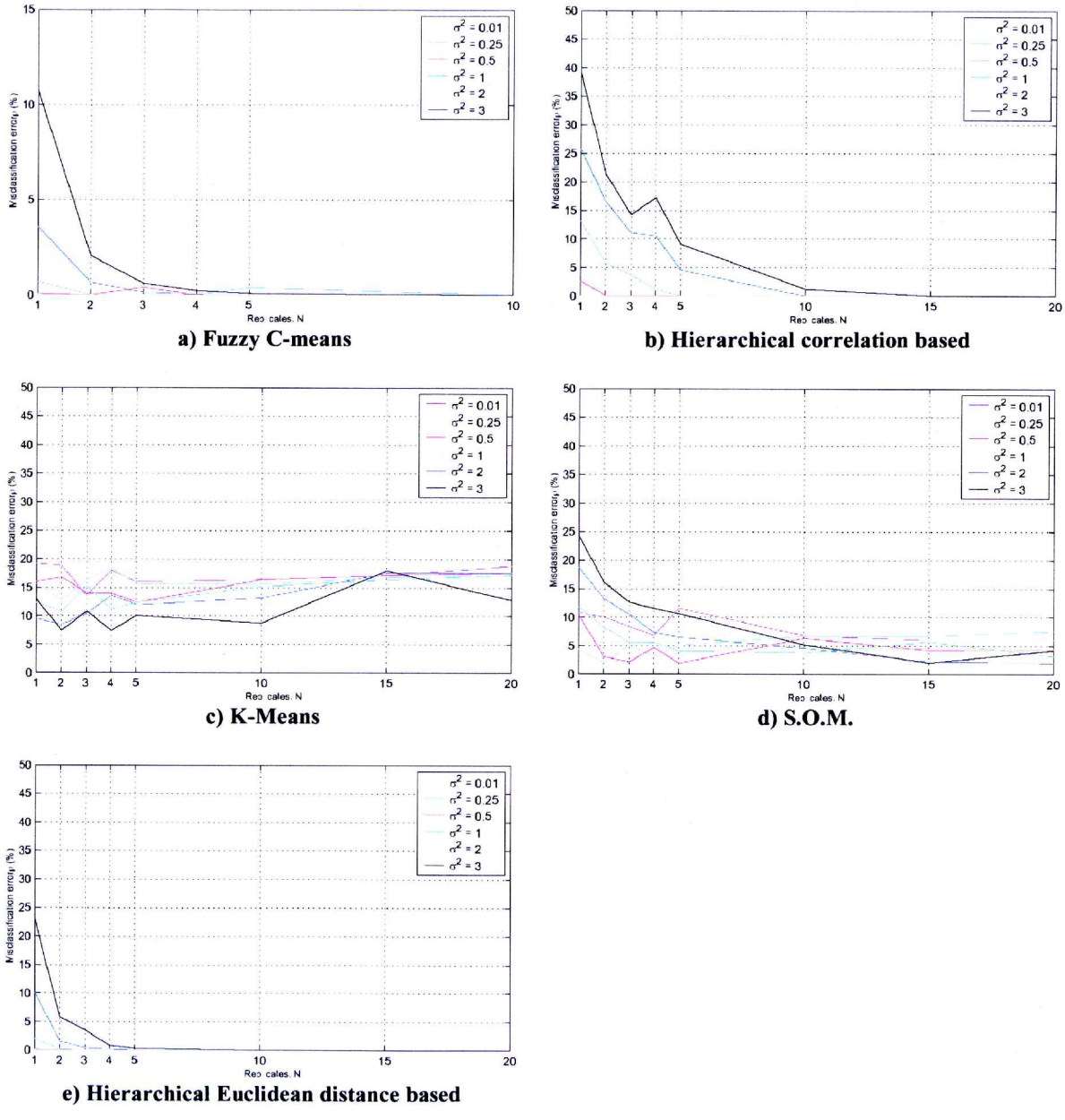



**FIG. 3.** Error graphs.

rithm $A_k$, and are dependent on this seeding. For each $k$, a graph can be computed as a function of $N$ to show the effect of the number of replications on the error rate. Altogether, the error rates form an error matrix

$$\mathbf{R}(N) = \begin{pmatrix} R_{11}(N) & R_{12}(N) & \cdots & R_{1q}(N) \\ R_{21}(N) & R_{22}(N) & \cdots & R_{2q}(N) \\ \vdots & \vdots & \ddots & \vdots \\ R_{q1}(N) & R_{q2}(N) & \cdots & R_{qq}(N) \end{pmatrix}. \tag{6}$$

The entry $R_{kj}(N)$ gives the error rate for algorithm $A_j$ under model seeding by $A_k$.

Table 1. Mean Confusion Matrices for Each Clustering Method for the Synthetic Data

(a) Fuzzy C-means with $\sigma^2 = 2$, $N = 2$

| *Cluster/class* | *1* | *2* | *3* | *4* | *5* | *SUM* |
|---|---|---|---|---|---|---|
| 1 | 49.34 | 0.04 | 0.86 | — | — | 50.24 |
| 2 | — | 49.96 | 0.02 | — | — | 49.98 |
| 3 | 0.66 | — | 49.12 | 0.02 | — | 49.8 |
| 4 | — | — | — | 49.98 | 0.06 | 50.04 |
| 5 | — | — | — | — | 49.94 | 49.94 |
| SUM | 50 | 50 | 50 | 50 | 50 | |

Mean Error = 1.66

(b) Fuzzy C-means with $\sigma^2 = 3$, $N = 1$

| *Cluster/class* | *1* | *2* | *3* | *4* | *5* | *SUM* |
|---|---|---|---|---|---|---|
| 1 | 45.16 | 0.4 | 10.34 | 0.3 | — | 56.2 |
| 2 | 0.46 | 48.46 | 3.12 | — | 0.66 | 52.7 |
| 3 | 3.2 | 0.46 | 32.97 | 0.7 | 0.12 | 37.45 |
| 4 | 1.18 | — | 2.82 | 48.1 | 0.72 | 52.82 |
| 5 | — | 0.68 | 0.74 | 0.9 | 48.5 | 50.82 |
| SUM | 50 | 50 | 49.99 | 50 | 50 | |

Mean Error = 26.8

(c) Correlation Hierarchical Clustering with $\sigma^2 = 2$, $N = 2$

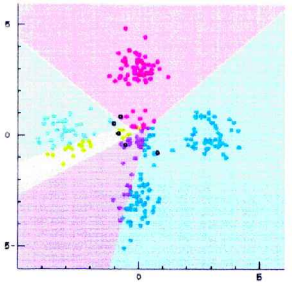
| *Cluster/class* | *1* | *2* | *3* | *4* | *5* | *SUM* |
|---|---|---|---|---|---|---|
| 1 | 46.28 | 0.94 | 30.2 | 0.78 | — | 78.2 |
| 2 | 0.14 | 46.56 | 0.22 | 0.24 | 0.08 | 47.24 |
| 3 | 3.58 | 0.66 | 18.89 | 0.57 | — | 23.7 |
| 4 | — | 0.18 | 0.66 | 47.26 | 0.26 | 48.36 |
| 5 | — | 1.66 | 0.02 | 1.13 | 49.66 | 52.47 |
| SUM | 50 | 50 | 49.99 | 49.98 | 50 | |

Mean Error = 41.34

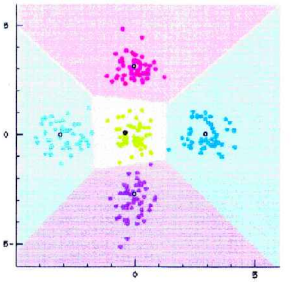
(d) Correlation Hierarchical Clustering with $\sigma^2 = 3$, $N = 1$

| *Cluster/class* | *1* | *2* | *3* | *4* | *5* | *SUM* |
|---|---|---|---|---|---|---|
| 1 | 36.82 | 6.02 | 19.66 | 10.28 | 0.14 | 72.92 |
| 2 | 3.9 | 29.14 | 5.96 | 1.5 | 2.36 | 42.86 |
| 3 | 6.84 | 4.38 | 18.66 | 4.28 | 1.13 | 35.29 |
| 4 | 2.34 | 1.84 | 3.9 | 23.8 | 2.24 | 34.12 |
| 5 | 0.1 | 8.61 | 1.82 | 10.14 | 44.12 | 64.79 |
| SUM | 50 | 49.99 | 50 | 50 | 49.99 | |

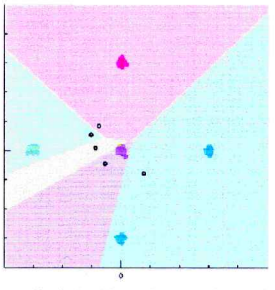
Mean Error = 97.46



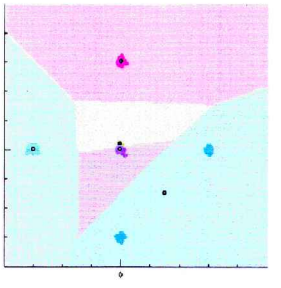
a) Random initialization, loosely packed



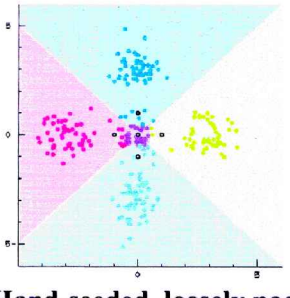
b) final clusters



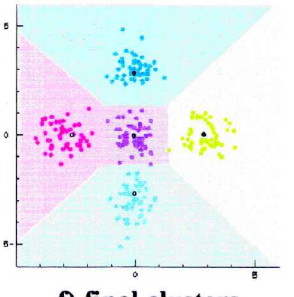
c) Random initialization, tightly packed



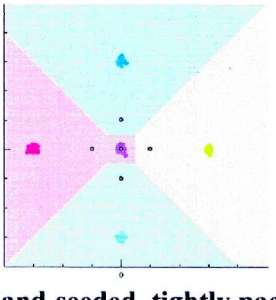
d) final clusters



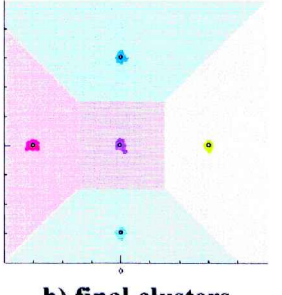
e) Hand-seeded, loosely packed



f) final clusters



g) Hand-seeded, tightly packed



h) final clusters

**FIG. 4.** Inconsistent behavior of the K-means algorithm.

It seems intuitive that the seeding algorithm should be favored when the clustering algorithms are applied to the model. Under fuzzy C-means seeding, one might think that fuzzy C-means will outperform K-means. While this initialization advantage is often the case, it may not be if the seeding algorithm has poor inference capability. K-means performs poorly in our model, and fuzzy C-means generally outperforms K-means when the model is seeded by K-means.

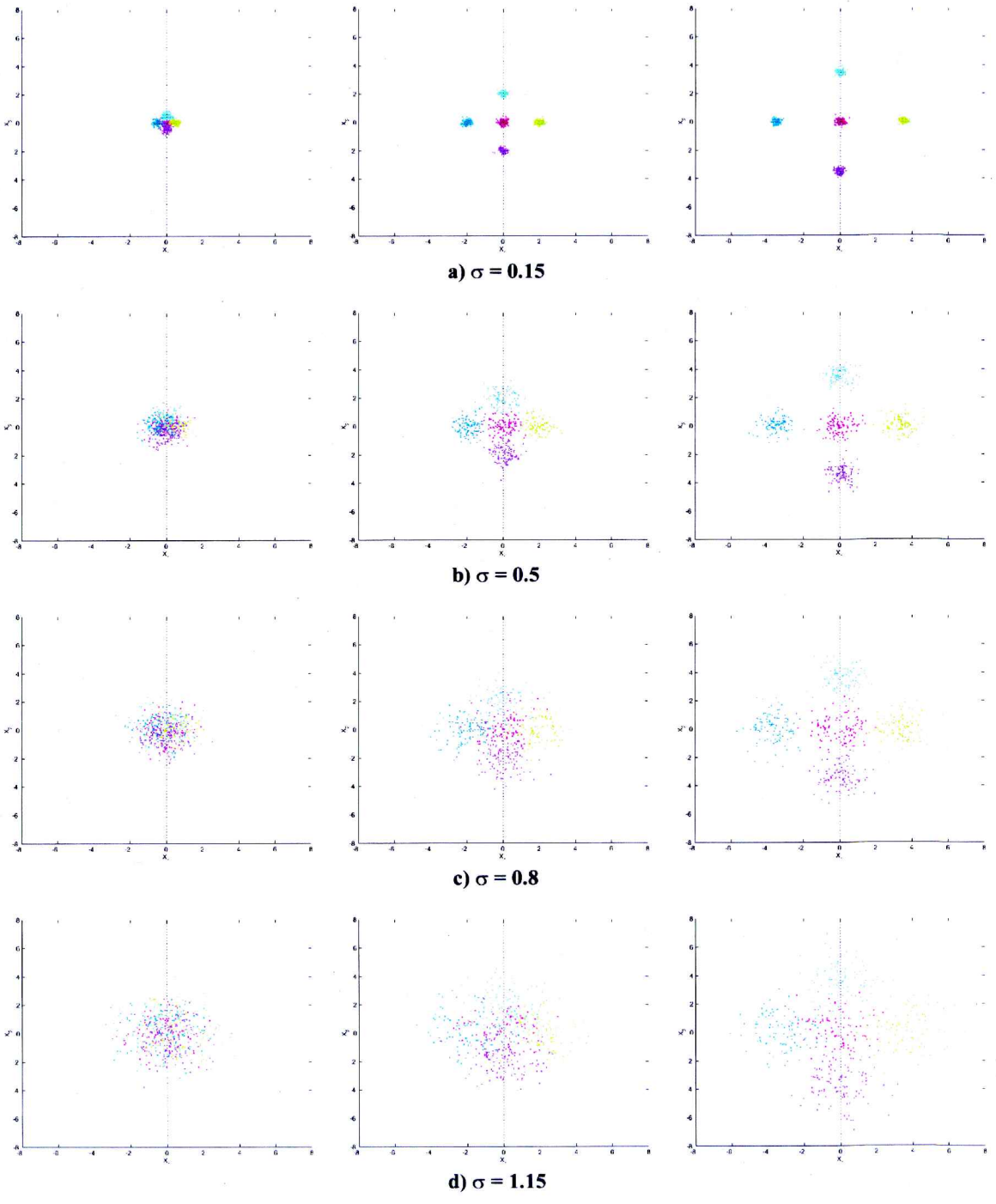



**FIG. 5.** Clusters as function of distance and variance.

To get an overall view of an algorithm's performance, one not so dependent on seeding, we can compute the various error rates for each seed and average the algorithm's performance over all seeds to obtain the global error rates $R_{\bullet 1}(N), R_{\bullet 2}(N), \ldots, R_{\bullet q}(N)$, where

$$R_{\bullet j}(N) = \frac{1}{q}\sum_{k=1}^{q} R_{kj}(N) \tag{7}$$

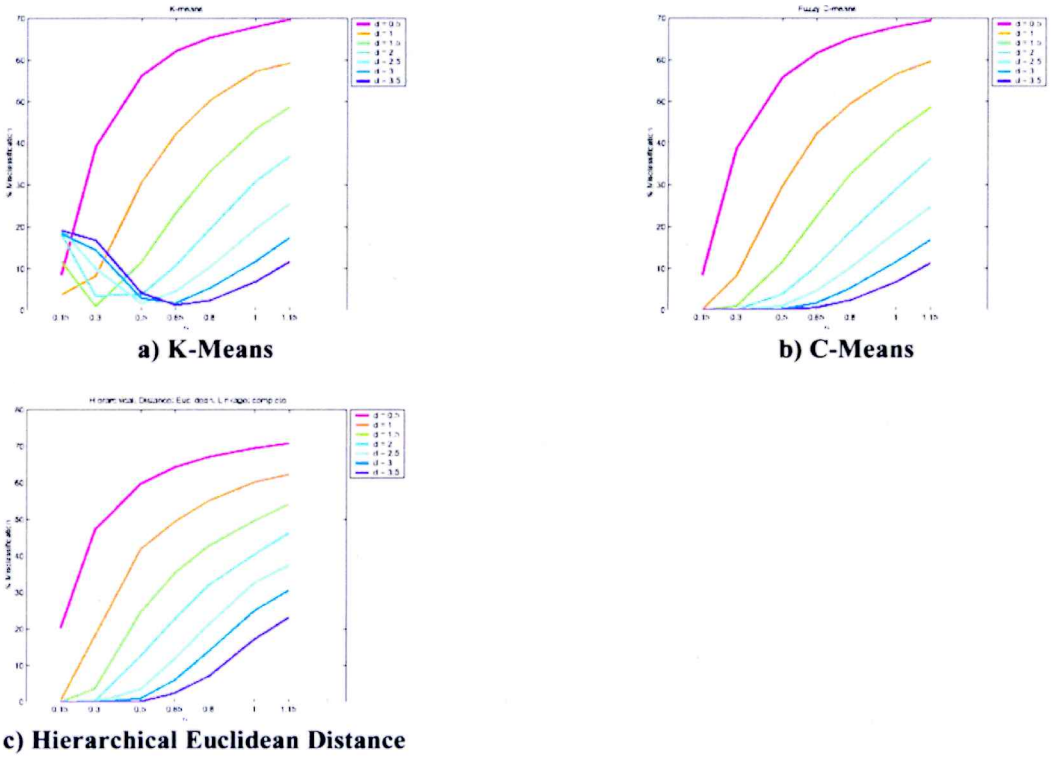



**FIG. 6.** Performance as function of distance and variance.

is the average performance of algorithm $A_j$ over all seeds. A slight modification occurs if one does not wish to average over all seeds, but only over a subcollection of seeds. For instance, owing to its generally poor performance, one might not wish to include seeding by K-means. In this case, the error rate corresponding to seeding by K-means is omitted from the average.

There are some considerations concerning variances for seed congruency classes. Consider the seed congruency class $U_{ki}$, the $i^{\text{th}}$ congruency class for the $k^{\text{th}}$ seeding algorithm. For the $n$ time points, there are $n$ means. Each of these is formed by the sample mean of the values at a time point of the profiles within the congruency class. Variances can similarly be formed from the sample variances at each time point. Alternatively, if some of the congruency classes are small, one can form the pooled variance over all time points. This results in all time points having the same model variance, but it avoids poor variance estimates for small classes.

To illustrate application, we use data published by Iyer *et al.* (1999) from an experiment to see the response of human fibroblasts to serum. The number of genes used in the original microarray is 8,613, which includes about 4,000 "named" human genes and another about 4,000 "anonymous" UniGene clusters on the basis of inclusion on the human transcript map and the lack of apparent homology to any other genes in the selected set. The original clustering analysis used only 517 genes from the microarray (and we use the same 517 genes). These were selected if either (i) their expression level deviated from that in quiescent fibroblasts by at least a factor of 2.20 in at least two of samples from serum-stimulated cells, or (ii) the standard deviation for the 13 time-point values of $\log_2$(expression-ratio) measured for the gene exceeded 0.7. In addition, observations in which the pixel-by-pixel correlation coefficients for the Cy3 and Cy5 fluorescence signals measured in a given array element were less than 0.6 were excluded.

We consider five and nine clusters, and seeds based on the five algorithms. For five clusters, Fig. 9 shows means (templates) for the seed congruency classes arising from self-organizing-map (SOM) clustering, along with the number of genes to be simulated in each seed class. To have sufficient data for small classes, simulations use twice the original cluster sizes. Time-point variances are pooled for each class. Simulated data based on the templates and their variances are shown in 2D-PCA space in Fig. 10 for differing numbers of replications. The first PCA plot ($N = 1$) portends the danger of not using replication.

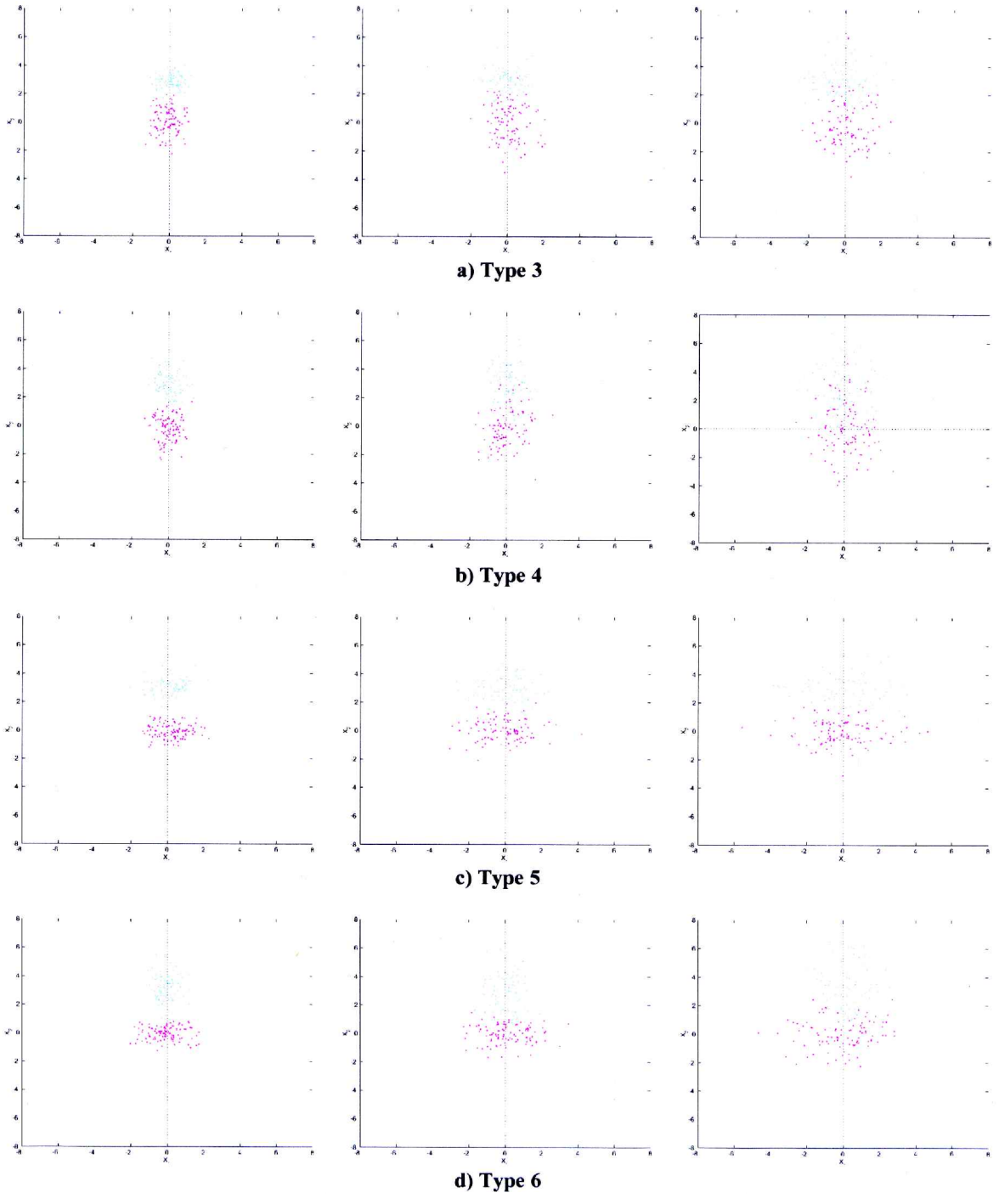



**FIG. 7.** Clusters as function of unequal variance.

This is verified in Fig. 11 where error curves (average of 50 simulations) are shown for the five clustering algorithms applied to the SOM seeds. Fuzzy C-means, K-means, and SOM perform tolerably for a single replication. Their errors are in the 10% range. The other two algorithms have errors in the range of 40%. The good news is that for only two replications, the errors for fuzzy C-means and SOM fall to only 5.5% and 2.5%, respectively. For three replications, the error for correlation-based hierarchical clustering remains high at 22%. Misclassification errors for the five algorithms averaged over all five seeds are shown

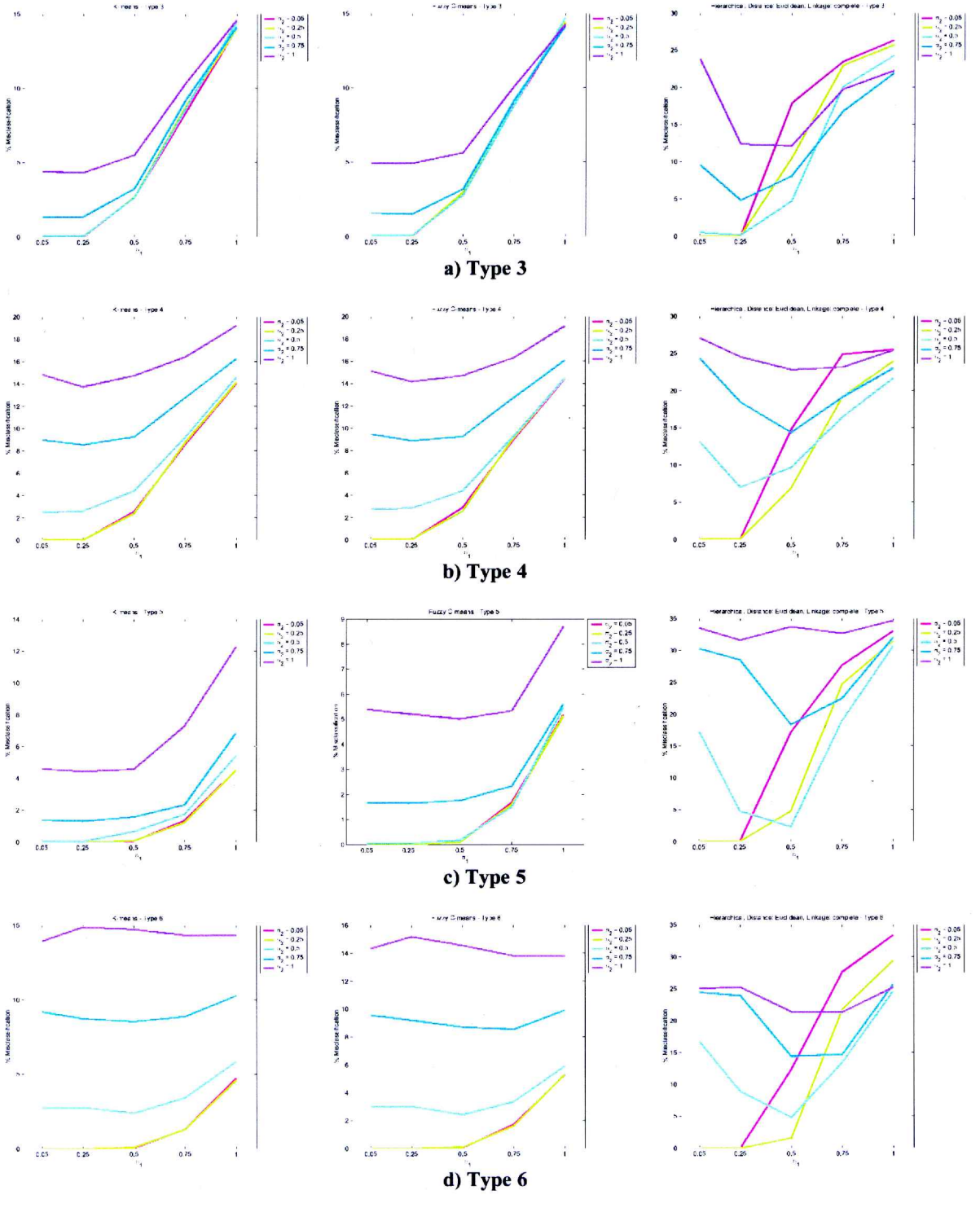



**FIG. 8.** Performance as function of unequal variance.

in Fig. 12. Once again, fuzzy C-means and SOM do well for very few replications. A similar graph using only fuzzy C-means, SOM, and hierarchical Euclidean-distance-based clustering is given in Fig. 13. The website contains confusion matrices for all five algorithms using SOM as the seed. It also includes the full analysis for all five seeds.

The issue of poor inference with decent-looking clusters is illustrated in Fig. 14. It shows an expression-profile dendrogram for hierarchical correlation-based clustering with SOM seeding for three replicates. The genes are listed vertically. Their expression ratios are listed horizontally. High ratios, low ratios, and ratios near 1 are indicated by strong red, strong green, and faded colors, respectively. In the next to last column, the graphic seems to visually indicate decent clustering. When the clusters are aligned with the congruency classes, many errors are seen. In fact, the error rate is 20.8%. Dendrograms for one and two replications are in the website. This kind of example shows that one must be cautious about drawing inferences from sample clusters.

## PATTERN RECOGNITION INTERPRETATION

The intent of this paper is to discuss clustering inference relative to congruency classes. From a pattern-recognition viewpoint, clustering is a kind of data-dependent partitioning. This section considers the analysis in terms of partitioning based on minimizing the Euclidean distance error.

The clustering model relates to separation of the distributions of the random vectors defining the congruency classes, and clustering is relative to sample points for the random vectors. The mixture distribution defined by these vectors (taken in proportion to the sizes of the congruency classes) characterizes the random vector $\mathbf{X}$ determining the partition of $n$-dimensional Euclidean space $\Re^n$, and the cluster error can be measured in terms of the distribution of $\mathbf{X}$. Given a random sample $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T$ for $\mathbf{X}$, the points $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$ are chosen to minimize the distance error

$$e_T^*(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m) = \frac{1}{T}\sum_{k=1}^{T} \min_{1 \le j \le m} \|\mathbf{X}_k - \mathbf{b}_j\|^2 \tag{8}$$

over all possible choices of the points $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m$. The points $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$ define a Voronoi partition, $\mathcal{V} = \{V_1, V_2, \ldots, V_m\}$, of $\Re^n$: a point lies in $V_k$ if its distance to $\mathbf{a}_k$ is no more than its distance to any other of the points $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$. Cluster $C_k$ consists of all sample points in $V_k$.

Now suppose the observations are labeled, meaning there is a joint probability vector $(\mathbf{X}, Y)$ defined on $\Re^n \times \{0, 1, \ldots, m\}$, and we wish to estimate the optimal classifier $\Psi[\Psi(\mathbf{X})$ being an estimator of $Y]$ based on the sample points $(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \ldots, (\mathbf{X}_T, Y_T)$. Having partitioned $\Re^n$ according to the minimization of $e_T^*$, a classifier $\Psi_T$ can be defined by majority vote. For any point $\mathbf{X}$, let $V_T(\mathbf{X})$ be the member of the partition containing $\mathbf{X}$ and define $\Psi_T(\mathbf{X})$ to be determined by voting among the labels for $(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \ldots, (\mathbf{X}_T, Y_T)$. Thus, a label is associated with each member of the partition. For each point $\mathbf{X}_k$, $\Psi_T(\mathbf{X}_k)$ is the label associated with the partition member containing the cluster including $\mathbf{X}_k$. This agrees with the way we have assigned clusters to congruency classes to compute the misclassification error. In that case, the cluster, $C_T(\mathbf{X}_k)$, containing $\mathbf{X}_k$ is assigned to the congruency class $U_j$ by voting among the congruency classes represented by points in $C_T(\mathbf{X}_k)$. The congruency classes determine the labels, a classifier $\Psi_T$ is determined, the values $\Psi_T(\mathbf{X}_1), \Psi_T(\mathbf{X}_2), \ldots, \Psi_T(\mathbf{X}_T)$ are determined, and the misclassification error, $\rho_T(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T)$, is the number of sample points for which $\Psi_T(\mathbf{X}_k) \ne Y_k$, where $Y_k$ is the label of the congruency class containing $\mathbf{X}_k$. The expression $\rho_T(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T)$ is the number of errors $\Psi_T$ makes on the sample data. It is the usual *resubstitution* estimate for the error of $\Psi_T$. It tends to underestimate the true error of $\Psi_T$ as an estimator of $Y$.

We focus the analysis on the case of two congruency classes, $U_0$ and $U_1$. In this case, the label space is $\{0, 1\}$, $\Psi$ is the binary Bayes classifier, and the error of $\Psi$ is the Bayes error, $\varepsilon^*$, for the distribution of $(\mathbf{X}, Y)$. The true error, $\varepsilon_T$, of $\Psi_T$ serves as an estimate for $\varepsilon^*$. The classifier $\Psi_T$ is strongly consistent as an estimator of $Y$ so long as the distribution of $\mathbf{X}$ has compact support and the number of clusters grows in a constrained manner in accordance with the number of samples. Specifically, let $k_T$ be the number of clusters for $T$ samples. If $k_T \to \infty$ and $k_T^2 T^{-1} \log T \to 0$ as $T \to \infty$, then $\varepsilon_T \to \varepsilon^*$ with probability one as $T \to \infty$ (Lugosi and Nobel, 1996). While powerful and interesting, this result does not appear to

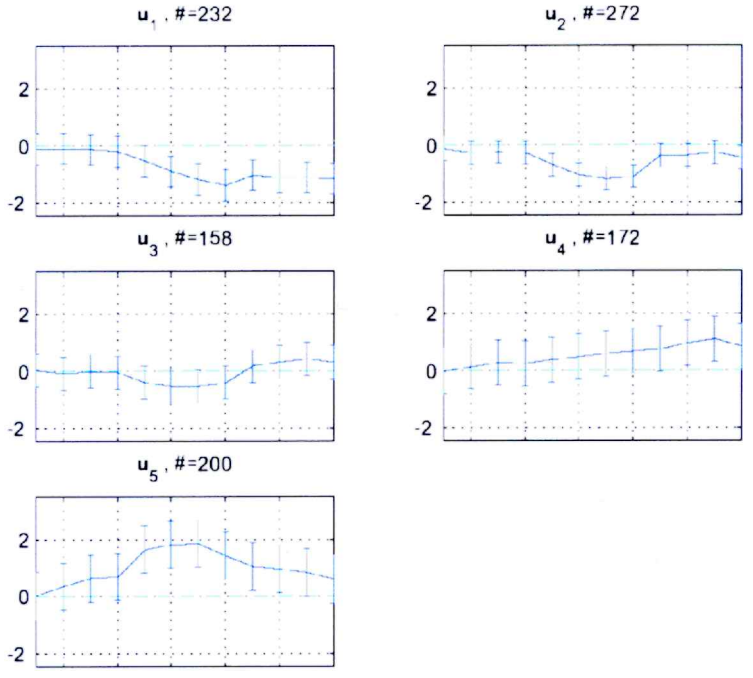



**FIG. 9.** Templates from S.O.M. Clustering.

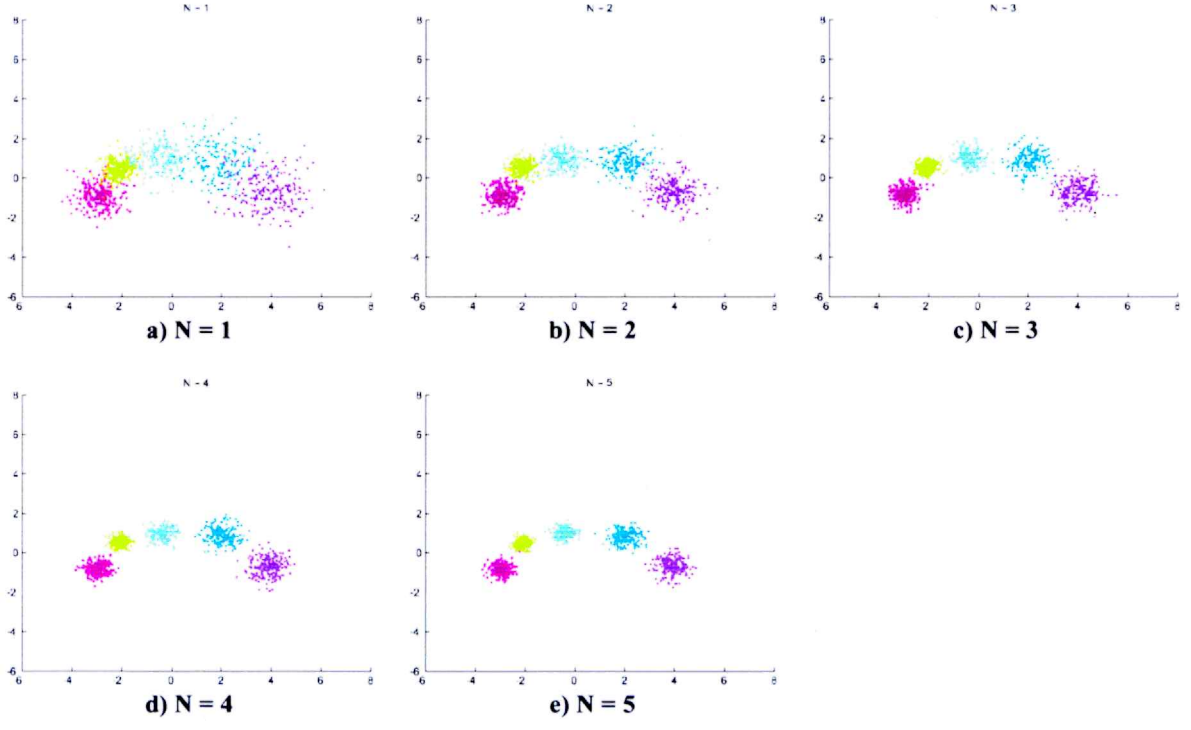



**FIG. 10.** Class overlapping (S.O.M. templates): increasing N.

be useful for our context. Not only is $T$ limited, but we do not necessarily want to increase the number of clusters as $n$ grows.

We have witnessed poor performance for the K-means algorithm. The K-means algorithm is designed to find points $\underline{\mathbf{a}}_1, \underline{\mathbf{a}}_2, \ldots, \underline{\mathbf{a}}_m$ that approximate the points $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$ that minimize the empirical Euclidean distance error of Equation 8. This approximation is dependent on seeding the algorithm, and performance depends on the particulars of the algorithm employed, as well as the data sets involved. Were the algorithm to actually find points $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$ that minimize the empirical error, then performance would be much better. For instance, in tightly packed and separated clusters that result from a large number of replications, one would do very well by selecting points interior to the clusters. The difficulty is in finding an algorithm to accomplish this end.

We close with a comment on the relationship between the empirical distance error of Equation 8 and the true clustering distance error. The true error is given in terms of the distribution of $\mathbf{X}$. For the points $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$ that minimize the empirical distance error, the true distance error is

$$e_T(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m) = E\Big[\min_{1 \le j \le m} \|\mathbf{X} - \mathbf{a}_j\|^2 | \mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T\Big]. \tag{9}$$

This expectation is conditional relative to the sample points and is approximated by the empirical distance error $e_T^*(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m)$. The empirical and true distance errors may differ significantly for small samples; however, if there exists a compact set $K$ such that $P(\mathbf{X} \subset K) = 1$, then their difference converges to zero with probability 1 as $T \to \infty$ (Linder *et al.*, 1994).

## CONCLUSION

Given either synthetic templates, or when seeded by data, the proposed statistical model can be used to evaluate clustering inference precision relative to variation and replication. This is a key issue when trying to draw inferences regarding similarity of behavior based on clusters. As might be expected, lower variance and more replications tend to yield greater precision. The procedure can be employed to determine which clustering algorithms appear to work well for the data at hand, as well has how many replications are necessary to achieve a desired level of performance. The entire toolbox is being made available in an interactive web-based implementation at the National Human Genome Research Institute that will allow a user to download data, choose a seeding algorithm (or algorithms), and get all of the output as it applies to the data.

## APPENDIX A: WEBSITE

Supplementary information is provided at http://gspsnap.tamu.edu/clustering/jcb/. To access the site, use *clustering* as both the account name and password. Detailed descriptions of algorithms and validation measures, annotation of graphics, a bibliography, and explanation of K-means behavior can be found by clicking on the Algorithms, Validations, Graphics, References, and K-means links, respectively.

The main web page consists of two simulations using real data, one with five templates (clusters) and the other with nine templates, and a third simulation with synthetic templates. For the first two simulations, as described in the paper, five different initial clustering methods have been used to generate initial templates, microarray data have been simulated for each template, and clustering analysis has been done using five different clustering methods. Output, including template means and variances, misclassification errors, confusion matrices, and validation measures, are tabulated and graphed in each section as the number, $N$, of replicates increases.

For each template generated by an initial clustering method, the Information page shows how the experiment is set up, templates with means and variances, and sample size. The Errors page shows the empirical expected values (average value over 50 repetitions) of misclassification errors, confusion matrices, and three different validation measures for each clustering method. The Examples page shows, for each clustering method, a clustering analysis including PCA projection maps, (dendrogram-like) clustering maps, templates recomputed after the clustering, confusion matrices, misclassification errors, and validation

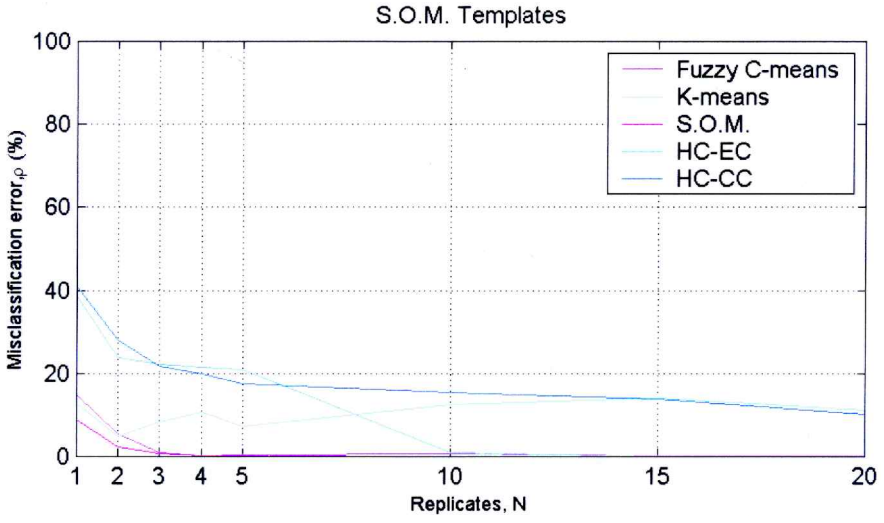



**FIG. 11.** Error curves for S.O.M. templates.

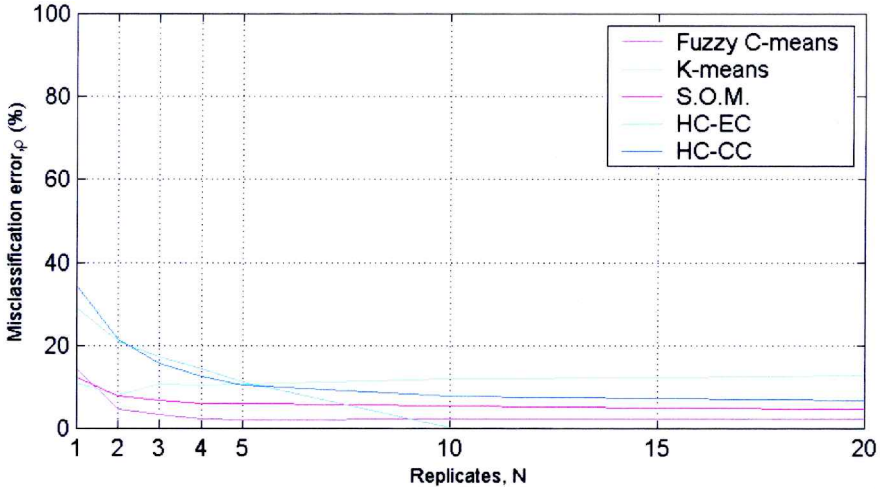



**FIG. 12.** Average error curves for the 5 template sets.

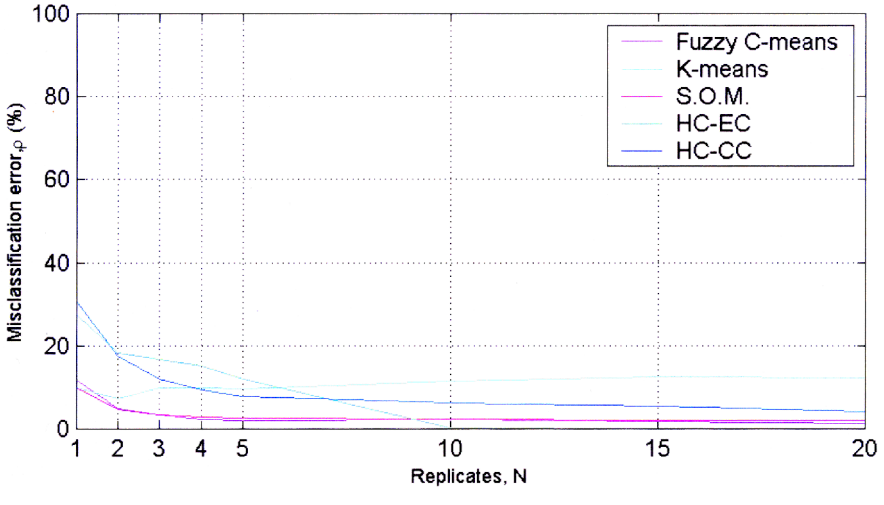



**FIG. 13.** Average error curves for template from Fuzzy C-means, S.O.M., and HC-EC.

measures for instances of simulated data, for different replicates. The Graphs page is a collection of all graphs, and the Export page is for users who want to download the input and output data to their local machines.

## APPENDIX B: CLUSTERING ALGORITHMS

### *K-means*

In the K-means algorithm, each sample point is placed into a unique cluster during each iteration, and the means are updated based on the classified samples. Given a set $S$ of $n$ sample points, those points are to be placed into $k$ clusters with $k$ means $m1, m2, m3, \ldots mk$. Algorithm implementation: For each point $s$ of $S$, calculate the distance $d(s, mi)$, for $i = 1, 2, \ldots, k$; let $m\_min$ be the nearest mean (i.e., $d(s, mi)$ is minimum); set $m\_min = (m\_min + s)/2$; and repeat until $mi$ does not change for $i = 1, \ldots, k$.

### *Fuzzy C-means*

Fuzzy C-means is a variation of K-means in which sample points have a degree of membership (or a probability of belonging) in each cluster, and the respective means are calculated based on these probabilities. Let $P(\omega_i : x_j)$ be the probability of the $j$-th sample belonging to the $i$-th cluster. For $c$ clusters and a parameter $b$, this is calculated from the trainning data by

$$P(\omega_i : x_j) = \frac{(1/\|x_j - m_i\|)^{1/(b-1)}}{\sum_{r=1}^{c}(1/\|x_j - m_r\|)^{1/(b-1)}} \tag{A1}$$

$$m_i = \frac{\sum_{j=1}^{n}[P(\omega_i : x_j)]^b x_j}{\sum_{j=1}^{n}[P(\omega_i : x_j)]^b} \text{ (mean of the } i\text{-th cluster).} \tag{A2}$$

Algorithm implementation (Duda *et al.*, 2000): For each point $s$ of $S$, compute the distance $d(s, mi)$, for $i = 1, 2, \ldots, k$; assign $s$ to the cluster associated to the nearest mean (i.e., $d(s, mi)$ is minimum); recompute means $m1, m2, m3, \ldots, mk$ according to Equation A2; recompute all probabilities $P(\omega_i : x_j)$ according to Equation A1; repeat until $mi$, for $i = 1, \ldots, k$, and $P(\omega_i : x_j)$ do not change.

### *Self-organizing map*

The SOM implements competive learning in neural networks. In competitive learning, the neurons receive identical input information and compete in their activities. The SOM defines a net of points that aproximates the density function of the input signal. There is a set of representatives $w_1, \ldots, w_n$, in the gene-expression space and a simple topology defines neighborhoods. A distance measure is used to assign points of the space to the nearest representative. For each sample point, the nearest representative $w_0$ is selected, after which $w_0$ and the representatives in a neighborhood of $w_0$ are updated. Algorithm implementation: set $t = 0$; randomly initialize the representatives $w_1, \ldots, w_m$; repeat the following procedure until convergence or $t > t_{max}$ and then assign each sample point to the representative closest to it: set $t = t + 1$; randomly select a sample point; determine the nearest representative $w_0$; update $w_1, \ldots, w_m$ according to the rule

$$w_i(t) = w_i(t-1) + \eta(t) f(D(w_i, w_0), t)(x - w_i(t-1)) \tag{A3}$$

where $\eta(t)$ is a variable learning rate, $D$ is a distance between cells defined by the topology of the net, and $f$ is a funtion defining the neighborhood. We have used the algorithm implemented in the toolbox

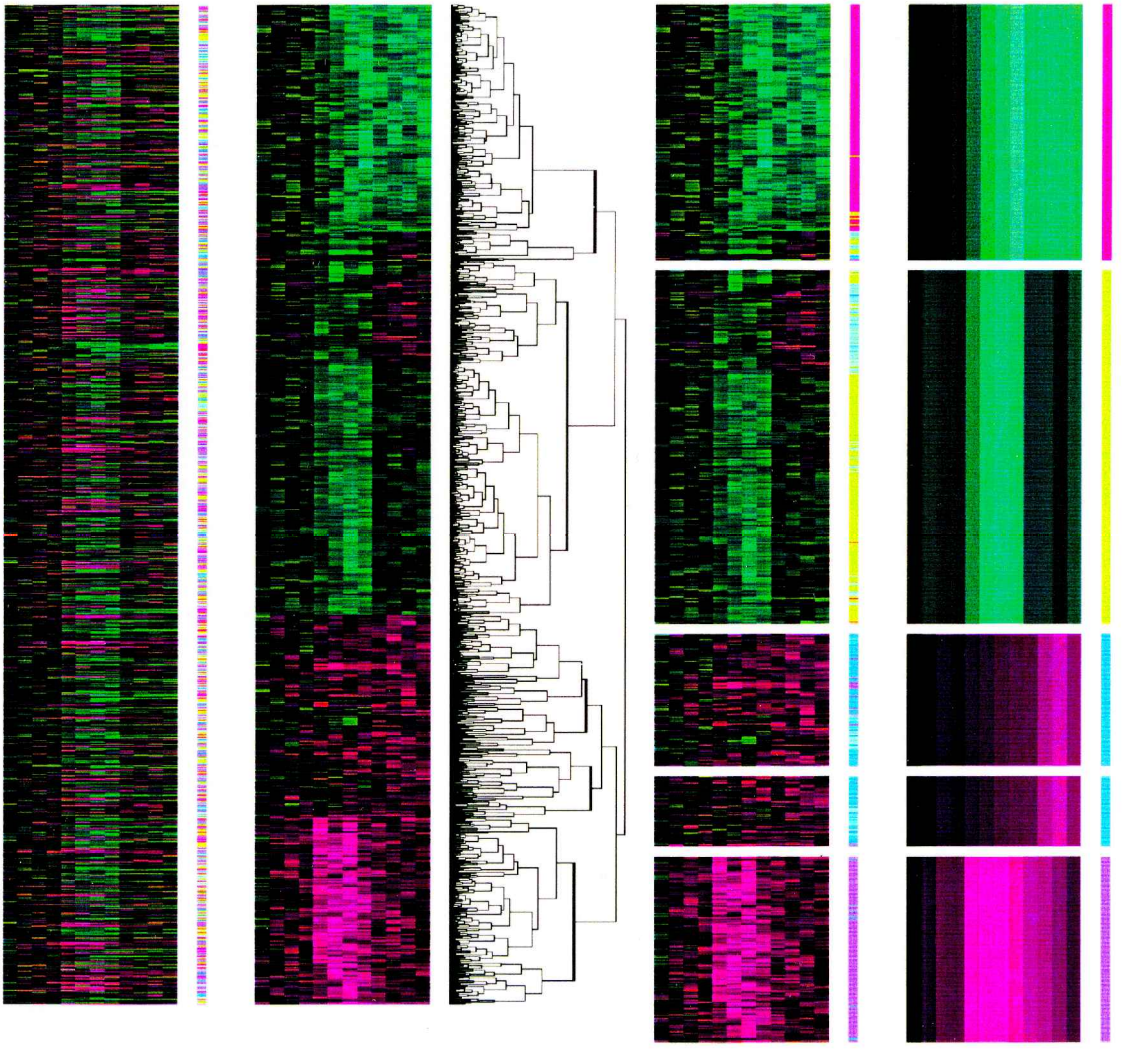


**FIG. 14.** Dendrogram for hierarchical correlation-based clustering ($N = 3$).

*nnet* of Matlab, using 2,500 iteractions to train the net. The topology used is a line, not a grid, with $N$ points, where $N$ is the number of clusters to be computed. The learning rate has an initial value of 0.9 and decreases 0.02 in each iteraction.

### *Hierarchical clustering*

Hierarchical clustering creates a hierarchy representing sample proximity in the feature space. It depends on the distance between samples and the distance between clusters. Three common cluster distances, yielding three variations of the algorithm, are given by:

$$\text{single-linkage algorithm:} \qquad d(Ci, Cj) = \min_{a \in Ci, b \in Cj} \{d(a, b)\},$$

$$\text{complete-linkage algorithm:} \qquad d(Ci, Cj) = \max_{a \in Ci, b \in Cj} \{d(a, b)\},$$

$$\text{average-linkage algorithm:} \qquad d(Ci, Cj) = \frac{1}{ni \times nj} \sum_{a \in Ci, b \in Cj} d(a, b),$$

where $ni$ and $nj$ are the number of samples of clusters $Ci$ and $Cj$, respectively. An alternative approach is to use the distance between the mean values of each cluster. In the experiments, we have used the three ways defined. Also, we have used three different distance measures: centered correlation, uncentered correlation, and Euclidean distance. Algorithm implementation: start with a feature space with $n$ samples; initialize $n$ clusters $Ci$, $i = 1, 2, \ldots, n$, each cluster consisting of one sample point; for $i = 1$ to $n - 1$, merge the nearer clusters $Ci$ and $Cj$.

## ACKNOWLEDGMENTS

The authors acknowledge the support of the National Human Genome Research Institute (USA) and the Fundacao de Amparo a Pesquisa Do Estado de Sao Paulo (Brazil), grant 98/15586-9.

## REFERENCES

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.

Duda, R.O., Hart, P.E., and Stork, D. 2000. *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York.

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* 95, 14863–14868.

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., and Dubes, R.C. 1988. *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ.

Jain, A.K., Duin, R.P.W., and Mao, J. 2000. Statistical pattern recognition: A review. *IEEE Trans. Pattern Analysis and Machine Intelligence* 22(1), 4–37.

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

Linder, T., Lugosi, G., and Zeger, K. 1994. Rates of convergence in the source coding theorem, empirical quantizer design, and universal lossy source coding. *IEEE Trans. Inform. Theory* 40, 1728–1740.

Lugosi, G., and Nobel, A. 1996. Consistency of data-driven histogram methods for density estimation and classification. *Ann. Statist.* 24, 687–706.

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

Theodoridis, S., and Koutroumbasd, K. 1999. *Pattern Recognition*, Academic Press, New York.

Address correspondence to:
*Edward R. Dougherty*
*Texas A&M University*
*Department of Electrical Engineering*
*3128 TAMU*
*College Station, TX 77843-3128*

*E-mail:* edward@ee.tamu.edu