Applications of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method

Jane Fridlyand and Sandrine Dudoit*

September 3, 2001

Technical report # 600

Address for correspondence:
Sandrine Dudoit
Division of Biostatistics
School of Public Health
University of California, Berkeley
140 Earl Warren Hall, # 7360
Berkeley, CA 94720-7360
E-mail: sandrine@stat.berkeley.edu

*Jane Fridlyand is a postdoctoral scientist in the Jain Lab at the UCSF Comprehensive Cancer Center, San Francisco, CA 94143-0128 (e-mail: janef@cc.ucsf.edu); Sandrine Dudoit is an Assistant Professor in the Division of Biostatistics, UC Berkeley, Berkeley, CA 94720-7360 (e-mail: sandrine@stat.berkeley.edu). This work was supported in part by a PMMB Burroughs-Wellcome graduate fellowship (JF) and a PMMB Burroughs-Wellcome postdoctoral fellowship (SD).

Abstract

The burgeoning field of genomics, and in particular microarray experiments, have revived interest in both discriminant and cluster analysis, by raising new methodological and computational challenges. The present paper discusses applications of resampling methods to problems in cluster analysis. A resampling method, known as bagging in discriminant analysis, is applied to increase clustering accuracy and to assess the confidence of cluster assignments for individual observations. A novel prediction-based resampling method is also proposed to estimate the number of clusters, if any, in a dataset. The performance of the proposed and existing methods are compared using simulated data and gene expression data from four recently published cancer microarray studies.

Keywords: Cluster analysis; discriminant analysis; unsupervised learning; supervised learning; number of clusters; resampling; bagging; microarray experiment; cancer; tumor classification.

1 Introduction

The burgeoning field of genomics, and in particular microarray experiments, have revived interest in both discriminant and cluster analysis, by raising new methodological and computational challenges. DNA microarrays are a new and promising biotechnology which allows the monitoring of expression levels in cells for thousands of genes simultaneously. Microarrays are being applied increasingly in biological and medical research to address a wide range of problems, such as the classification of tumors [1, 2, 16, 26, 27, 29]. A reliable and precise classification of tumors is essential for successful diagnosis and treatment of cancer. By allowing the monitoring of expression levels on a genomic scale, microarray experiments may lead to a more complete understanding of the molecular variations among tumors and hence to a finer and more reliable classification.

There are two main aspects to classification: discrimination and clustering. In *discriminant analysis*, also known as *supervised learning*, observations (*e.g.* tumor mRNA samples) are known to belong to prespecified classes, and the task is to build predictors for allocating new observations to these classes. By contrast, in *cluster analysis*, or *unsupervised learning*, the classes are unknown *a priori* and the task is to determine these classes from the data themselves, *i.e.*, to determine the number of classes and assign each observation to one of these classes. A detailed overview of statistical issues in discriminant and cluster analysis is presented in "Panel on discriminant analysis, classification and clustering" [25]. The present paper focuses on cluster analysis; for a discussion of discriminant analysis in the context of microarray experiments the reader is referred to Dudoit *et al.* [11].

Resampling methods such as bagging (Breiman [6]) and boosting (Breiman [7], Freund & Schapire [15]) have been applied successfully in the context of discriminant analysis to improve prediction accuracy. In the present paper, it is proposed to apply resampling techniques in the context of cluster analysis, to (i) estimate the number of clusters, if any, in a dataset, and (ii) improve and assess the accuracy of a given clustering procedure. Since the groups obtained from cluster analysis are often used for prediction purposes later on, the approach to (i) relies on ideas from discriminant analysis. For problem (ii), bagging is used to generate and aggregate multiple clusterings and to assess the confidence of cluster assignments for individual observations. Although the proposed resampling methods are applicable to general clustering problems, particular attention is given to the clustering of tumors using gene expression data.

The paper is organized as follows. The remainder of Section 1 introduces basic notions in cluster analysis and provides a motivation for the issues addressed in this paper. After a discussion of existing approaches, Section 2 presents a novel prediction-based resampling method, *Clest*, for estimating the number of clusters in a dataset. Section 3 proposes two resampling methods for improving the accuracy of a clustering method and for assessing the confidence of cluster assignments for individual observations. In Section 4, the proposed methods of Sections 2 and 3 are compared to existing approaches using simulated data. The same existing and new methods are applied in Section 5 to gene expression data from four recently published cancer microarray studies. Finally, Section 6 summarizes our findings and outlines open questions.

1.1 Motivation

The aim of cluster analysis is to group observational units on the basis of measurements and according to prespecified criteria. Important issues, which will only be briefly addressed in the present paper, include: the selection of observational units, the selection of variables for defining the groupings, the transformation and standardization of variables, the choice of a similarity or dissimilarity measure, the choice of a clustering method (Milligan [23]). The two main concerns in this paper are: (i) estimating the number of clusters, if any, in a dataset, and (ii) improving and assessing the accuracy of a given clustering procedure.

When a clustering algorithm is applied to a set of observations, a partition of the data is returned whether or not the data exhibit a true or "natural" clustering structure. This fact causes no problems if clustering is done to obtain a practical grouping of the given set of objects, for instance, for organizational purposes (*e.g.* hierarchical clustering for displaying large gene expression data matrices as in Eisen *et al.* [12]). However, if interest lies primarily in the recognition of an unknown classification of the data, an artificial clustering is not satisfactory, and clusters resulting from the algorithm must be investigated for their relevance and reproducibility. This task can be performed by descriptive and graphical exploratory methods, or by relying on probabilistic models and suitable statistical significance tests (*e.g.* model based clustering of Fraley & Raftery [14]).

Once novel classes are identified and cluster labels are assigned to the observations, the next step is often to build a classifier for predicting the class of future observations. The reproducibility of cluster assignments becomes very important in this context, and therefore provides a motivation for using ideas from discrimination to estimate the number of clusters. After a brief summary of existing methods in Section 2.1, a novel resampling method combining ideas from discriminant and cluster analysis is proposed in Section 2.2 for estimating the number of clusters in a dataset.

The ability to accurately allocate observations to clusters and assess the confidence of each cluster assignment is an essential aspect of the clustering problem. For example, in the context of tumor classification, the definition of new tumor classes is based on the clustering results and these classes are then used to build predictors for new tumor samples. Inaccurate cluster assignments could lead to erroneous diagnoses and unsuitable treatment protocols. Two bootstrap-based methods for improving the accuracy of a clustering procedure and assessing the confidence of cluster assignments are proposed in Section 3.

1.2 Clustering algorithm: Partitioning around medoids

The data are assumed to be sampled from a mixture distribution with K components corresponding to the K clusters to be recovered. Let $(X_1, \ldots, X_p)$ denote a random $1 \times p$ vector of explanatory variables and let $Y \in \{1, \ldots, K\}$ denote the component or cluster label. Given a sample of X's, the goal is to estimate the number of clusters K and to estimate, for each observation, its cluster label Y.

Suppose we have data $\mathbf{X} = (x_{ij})$ on p explanatory variables (*e.g.* genes) for n observations (*e.g.* tumor mRNA samples), where x_{ij} denotes the realization of variable j for observation i and $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$ denotes the data vector for observation i, $i = 1, \ldots, n$, $j = 1, \ldots, p$. We consider clustering algorithms that divide the learning set $\mathcal{L} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ into K clusters of observations that are "similar" to each other, where K is a user prespecified integer. More specifically, the clustering $\mathcal{P}(\cdot, \mathcal{L})$ assigns class labels $\mathcal{P}(\mathbf{x}_i, \mathcal{L}) = \hat{y}_i$ to each observation, where $\hat{y}_i \in \{1, \ldots, K\}$. Clustering algorithms generally operate on a matrix of pairwise *dissimilarities* (*similarities*) between the observations to be clustered, such as the Euclidean or Manhattan distance matrices (Mardia *et al.* [22]). A partitioning of the learning set can be produced directly by partitioning clustering methods (*e.g.* k-means, partitioning around medoid (PAM), self-organizing maps (SOM)) or by hierarchical clustering methods, by "cutting" the dendogram to obtain K "branches" or clusters.

In this report, the proposed resampling methods are illustrated using the *Partitioning Around Medoids* or *PAM* method of Kaufman & Rousseeuw [20]. As implemented in the `R` and `S-Plus` libraries `cluster`, the *PAM* function takes as its arguments a dissimilarity matrix (*e.g.* the Euclidean distance matrix as used here) and a prespecified number of clusters K. The *PAM* algorithm is based on the search for K representative objects, or *medoids*, among the observations to be clustered. After finding a set of K medoids, K clusters are constructed by assigning each observation to the nearest medoid. The goal is to find K medoids which minimize the sum of the dissimilarities of the observations to their closest medoid. The algorithm first looks for a good initial set of medoids, then finds a local minimum for the objective function, that is, a solution such that there is no single switch of an observation with a medoid that will decrease the objective.

The *PAM* algorithm tends to be more robust and computationally efficient than k-means. In addition, *PAM* provides a graphical display, the *silhouette plot*, which can be used to select the number of clusters and to assess how well individual observations are clustered. Let a_i denote the average dissimilarity between i and all other observations in the cluster to which i belongs. For any other cluster C, let $d(i, C)$ denote the average dissimilarity of i to all objects of C and let b_i denote the smallest of these $d(i, C)$. The *silhouette width* of observation i is $sil_i = (b_i - a_i)/\max(a_i, b_i)$ and the overall *average silhouette width* is simply the average of sil_i over all observations i, $\bar{sil} = \sum_i sil_i/n$. Intuitively, objects with large silhouette width sil_i are well clustered, while those with small sil_i tend to lie between clusters. Kaufman & Rousseeuw suggest estimating the number of clusters K by that which gives the largest average silhouette width, $\bar{sil}$.

2 Estimating the number of clusters

2.1 Existing methods

2.1.1 Null hypothesis

Suppose that the maximum possible number of clusters in the data is set to M, $2 \leq M \leq n$. One approach to estimating the number of clusters K is to look for $\hat{K}$, $1 < \hat{K} \leq M$, that provides the strongest significant evidence against the null hypothesis H_0 of $K = 1$, that is, "no clusters" in the data. Two commonly used parametric null hypotheses are the unimodality hypothesis and the uniformity hypothesis.

Under the *unimodality hypothesis* the data are thought to be a random sample from a multivariate normal distribution. This model typically gives a high probability of rejection of the null $K = 1$ if the data are sampled from a distribution with a lower kurtosis than the normal distribution, such as the uniform distribution (Sarle [30]).

The *uniformity hypothesis*, also referred to as *random position hypothesis*, states that the data are sampled from a uniform distribution in p-dimensional space (Bock [4], Hartigan [17], Jain & Dubes [19]). Methods based on the uniformity hypothesis tend to be conservative, *i.e.*, lead to few rejections of the null H_0, when the data are sampled from a strongly unimodal distribution such as the normal distribution. In two or more dimensions, and depending on the test statistic, the results can be very sensitive to the region of support of the reference distribution (Sarle [30]).

For both types of hypotheses, evidence against the null H_0 can be summarized formally under probability models for the data or more informally by using internal indices as described next.

2.1.2 Internal indices

Numerous methods have been proposed for testing the null hypothesis $K = 1$ and estimating the number of clusters in a dataset, however, none of them are completely satisfactory. Jain & Dubes [19] provide a general overview of such methods. The majority of existing approaches do not attempt to formally test the null hypothesis that $K = 1$, but rather look for the clustering structure under which a summary statistic of interest is optimal, being large or small depending on the statistic (Calinski and Harabasz [9], Davies & Bouldin [10], Krzanowski & Lai [21]). These statistics are typically functions of the within, and possibly between, clusters sums of squares, and belong to the class of so-called *internal indices*, in the sense that they are computed from the same observations that are used to create the clustering. Consequently, the distribution of these indices is intractable. In particular, since clustering methods attempt to maximize the separation between clusters, the ordinary significance tests such as analysis of variance F-tests are not valid for testing differences between the clusters. Milligan & Cooper [24] conducted an extensive Monte Carlo evaluation of thirty internal indices. Other approaches include modeling the data using Gaussian mixtures and applying a Bayesian criterion to determine the number of components in the mixture (Fraley & Raftery [14]). A recent proposal of Tibshirani *et al.* [31], called the *gap statistic method*, calibrates an internal index, such as the within clusters sum of squares, against its expectation under a suitably defined null hypothesis (note that gap tests have been used in another context in cluster analysis by Bock [4], p. 81, to test the null hypothesis of a "homogeneous" population against the alternative of "heterogeneity"). Tibshirani *et al.* conducted a comparative Monte Carlo study of the gap statistic and several of the internal indices which showed a better performance in the study of Milligan & Cooper [24]. These internal indices and the gap statistic are described in more detail next.

For a given partition of the learning set into $1 \leq k \leq M$ clusters, define $\mathbf{B}_k$ and $\mathbf{W}_k$ to be the $p \times p$ matrices of between and within k-clusters sums of squares and cross-products (Mardia *et al.* [22]). Note that $\mathbf{B}_1$ is not defined.

1. ***sil*** – Kaufman & Rousseeuw [20] suggest selecting the number of clusters $k \geq 2$ which gives the largest average silhouette width, $\bar{sil}_k$. Silhouette widths were defined in Section 1.2 with the clustering algorithm *PAM*.

2. ***ch*** – Calinski and Harabasz [9]. For each number of clusters $k \geq 2$, define the index

$$ch_k = \frac{\mathrm{tr}\mathbf{B}_k/(k-1)}{\mathrm{tr}\mathbf{W}_k/(n-k)},$$

   where tr denotes the trace of a matrix, *i.e.*, the sum of the diagonal entries. The estimated number of clusters is $\mathrm{argmax}_{k \geq 2} ch_k$.

3. ***kl*** – Krzanowski & Lai [21]. For each number of clusters $k \geq 2$, define the indices

$$diff_k = (k-1)^{2/p}\mathrm{tr}\mathbf{W}_{k-1} - k^{2/p}\mathrm{tr}\mathbf{W}_k \qquad \text{and} \qquad kl_k = |diff_k|/|diff_{k+1}|.$$

   The estimated number of clusters is $\mathrm{argmax}_{k \geq 2} kl_k$.

4. ***hart*** – Hartigan [18]. For each number of clusters $k \geq 1$, define the index

$$hart_k = \Big(\frac{\mathrm{tr}\mathbf{W}_k}{\mathrm{tr}\mathbf{W}_{k+1}} - 1\Big)(n-k-1).$$

   The estimated number of clusters is the smallest $k \geq 1$ such that $hart_k \leq 10$.

5. ***gap*** or ***gapPC*** – Tibshirani *et al.* [31]. This method compares an observed internal index, such as the within clusters sum of squares, to its expectation under a reference null distribution as follows. For each number of clusters $k \geq 1$, compute the within clusters sum of squares $\mathrm{tr}\mathbf{W}_k$. Generate B (here $B = 10$) reference datasets under the null distribution and apply the clustering algorithm to each, calculating the within clusters sums of squares $\mathrm{tr}\mathbf{W}_k^1, \ldots, \mathrm{tr}\mathbf{W}_k^B$. Compute the estimated *gap statistic*

$$gap_k = \frac{1}{b}\sum_b \log \mathrm{tr}\mathbf{W}_k^b - \log \mathrm{tr}\mathbf{W}_k,$$

and the standard deviation sd_k of $\log \mathrm{tr}\mathbf{W}_k^b$, $1 \le b \le B$. Let $\tilde{sd}_k = sd_k\sqrt{1+1/B}$. The estimated number of clusters is the smallest $k \ge 1$ such that $gap_k \ge gap_{k^*} - \tilde{sd}_{k^*}$, where $k^* = \mathrm{argmax}_{k \ge 1} gap_k$.

Tibshirani *et al.* [31] chose the uniformity hypothesis to create a reference null distribution and considered two approaches for constructing the region of support of the distribution. In the first approach, the sampling window for the j^{th} variable, $1 \le j \le p$, is the range of the observed values for that variable. In the second approach, following Sarle [30], the variables are sampled from a uniform distribution over a box aligned with the principal components of the centered design matrix (*i.e.*, the columns of $\mathbf{X}$ are first set to have mean 0 and the singular value decomposition of $\mathbf{X}$ is computed). The new design matrix is then back-transformed to obtain a reference dataset. While the first approach has the advantage of simplicity, the second takes into account the shape of the data distribution. Note that in both approaches the variables are sampled independently. The version of the gap method which uses the original variables to construct the region of support is referred to as *gap* and the second version as *gapPC*, where "*PC*" stands for *P*rincipal *C*omponents.

Note that of the above methods, only *hart*, *gap*, and *gapPC* allow the estimation of only one cluster in the data, *i.e.*, $\hat{K} = 1$.

2.1.3 External indices

The term "validation of a clustering procedure" usually refers to the ability of a given method to recover the true clustering structure in a dataset. There have been several attempts to assess validity on theoretical grounds (Bock [4], Hartigan [18]), however, such approaches turn out to be of little applicability in the context of high-dimensional complex datasets. In many validation studies, clustering methods are evaluated based on their performance on empirical datasets with *a priori* known cluster labels (Hartigan [18]) or, more commonly, based on simulation studies where true cluster labels are known. In order to assess the ability of a clustering algorithm to recover true cluster labels it is necessary to define a measure of agreement between two partitions; the first partition being the *a priori* known clustering structure of the data, and the second partition resulting from the clustering procedure. In the clustering literature, measures of agreement between partitions are referred to as *external indices*; several such indices are reviewed next.

Consider two partitions of n objects $\mathbf{x}_1, \ldots, \mathbf{x}_n$: the R-class partition $\mathcal{U} = \{u_1, \cdots, u_R\}$ and the C-class partition $\mathcal{V} = \{v_1, \cdots, v_C\}$. External indices of partitional agreement can be expressed in terms of a contingency table (Table 1), with entry n_{ij} denoting the number of objects that are both in clusters u_i and v_j, $i = 1, \ldots, R$, $j = 1, \ldots, C$ (Jain & Dubes [19]). Let $n_{i.} = \sum_{j=1}^C n_{ij}$ and $n_{.j} = \sum_{i=1}^R n_{ij}$ denote the row and column sums of the contingency table, respectively, and let $Z = \sum_{i=1}^R \sum_{j=1}^C n_{ij}^2$.

1. **Rand** – Rand [28]

$$Rand = 1 + \left(Z - (1/2)(\sum_{i=1}^R n_{i.}^2 + \sum_{j=1}^C n_{.j}^2)\right) / \binom{n}{2}.$$

Table 1: Contingency table for two partitions of n objects

$$
\begin{array}{c|cccc|c}
 & v_1 & v_2 & \cdots & v_C & \\
\hline
u_1 & n_{11} & n_{12} & \cdots & n_{1C} & n_{1.} \\
u_2 & n_{21} & n_{22} & \cdots & n_{2C} & n_{2.} \\
\vdots & \vdots & \vdots & & \vdots & \vdots \\
u_R & n_{R1} & n_{R2} & \cdots & n_{RC} & n_{R.} \\
\hline
 & n_{.1} & n_{.2} & \cdots & n_{.C} & n_{..} = n
\end{array}
$$

2. **Jaccard** – Jain & Dubes [19]

$$Jac = \big(Z - n\big)/\big(\sum_{i=1}^{R} n_{i.}^2 + \sum_{j=1}^{C} n_{.j}^2 - Z - n\big).$$

3. **FM** – Fowlkes & Mallows [13]

$$FM = \big(1/2\big)\big(Z - n\big)/\Big[\sum_{i=1}^{R}\binom{n_{i.}}{2}\sum_{j=1}^{C}\binom{n_{.j}}{2}\Big]^{\frac{1}{2}}.$$

Note that *Rand* and FM are linear functions of Z, and hence are linear functions of one another, conditional on the row and column sums in Table 1. If the row and column sums in Table 1 are fixed, but the partitions are selected at random, *i.e.*, if there is independence in the table, the hypergeometric distribution can be applied to determine the expected value of quantities such as Z. In particular

$$E\Big[\sum_{i=1}^{R}\sum_{j=1}^{C}\binom{n_{ij}}{2}\Big] = (1/2)E(Z-n) = \sum_{i=1}^{R}\binom{n_{i.}}{2}\sum_{j=1}^{C}\binom{n_{.j}}{2}\Big/\binom{n}{2}.$$

An external index S is often standardized in such a way that its expected value is 0 when the partitions are selected at random and 1 when they match perfectly. This amounts to computing a standardized external index

$$S' = \frac{S - E(S)}{S_{max} - E(S)},$$

where S_{max} is the maximum value of the statistic S and $E(S)$ is the expected value of S when partitions are selected at random. Accordingly, an often used correction for the *Rand* statistic is

$$Rand' = \frac{\sum_{i=1}^{R}\sum_{j=1}^{C}\binom{n_{ij}}{2} - [1/\binom{n}{2}]\sum_{i=1}^{R}\binom{n_{i.}}{2}\sum_{j=1}^{C}\binom{n_{.j}}{2}}{(1/2)[\sum_{i=1}^{R}\binom{n_{i.}}{2} + \sum_{j=1}^{C}\binom{n_{.j}}{2}] - [1/\binom{n}{2}]\sum_{i=1}^{R}\binom{n_{i.}}{2}\sum_{j=1}^{C}\binom{n_{.j}}{2}}.$$

The significance of an observed external index is usually assessed under the assumption that the two partitions to be compared are independent. This assumption does not hold for the resampling methods described in the following section, since the same data are used to produce the two partitions. Nevertheless, external indices are convenient tools for comparing two clusterings, and are used in the new resampling method *Clest*. In this context, one should think of these indices as *internal* rather than *external* measures.

2.2 A prediction-based resampling method, *Clest*

In this section, a new prediction-based resampling method, *Clest*, is proposed for estimating the number of clusters, if any, in a dataset. The idea behind *Clest* is very intuitive if one is concerned with reproducibility of cluster assignments.

It is proposed to estimate the number of clusters K by repeatedly randomly dividing the original dataset into two non-overlapping sets, a learning set $\mathcal{L}^b$ and a test set $\mathcal{T}^b$. For each iteration and for each number of clusters k, a clustering $\mathcal{P}(\cdot, \mathcal{L}^b)$ of the learning set $\mathcal{L}^b$ is obtained and a predictor $C(\cdot, \mathcal{L}^b)$ is built using the class labels from the clustering. The predictor $C(\cdot, \mathcal{L}^b)$ is then applied to the test set $\mathcal{T}^b$ and the predicted labels are compared to those produced by applying the clustering algorithm to the test set, using one of the external indices (or similarity statistics) described in Section 2.1.3. The number of clusters is estimated by comparing the observed similarity statistic for each k to its expected value under a suitable null distribution with $K = 1$. The estimated number of clusters is defined to be the $\hat{K}$ corresponding to the largest significant evidence against H_0 of $K = 1$.

An early version of this approach was introduced by Breckenridge [5] under the name of *replication analysis* and was designed to evaluate the stability of a clustering. In the original replication analysis, the number of clusters k is fixed, and the data are randomly divided into two samples. A clustering algorithm partitions both samples into k clusters, and the centroids of the clusters of the first sample are computed. A second set of labels is assigned to the observations in the second sample by assigning to each observation the cluster label of the closest centroid from the first sample. Finally, an external index is used to assess the agreement between the two partitions of the second sample. This measure reflects the stability of the clustering structure. The *Clest* algorithm proposed here generalizes and extends the work of Breckenridge [5].

Clest algorithm for estimating the number of clusters in a dataset.

Denote the maximum possible number of clusters by M, $2 \leq M \leq n$. For each number of clusters k, $2 \leq k \leq M$, perform steps 1–4.

1. Repeat the following B times:
   (a) Randomly split the original learning set $\mathcal{L}$ into two non-overlapping sets, a learning set $\mathcal{L}^b$ and a test set $\mathcal{T}^b$.
   (b) Apply a clustering algorithm $\mathcal{P}$ to the learning set $\mathcal{L}^b$ to obtain a partition $\mathcal{P}(\cdot, \mathcal{L}^b)$.
   (c) Build a classifier $\mathcal{C}(\cdot, \mathcal{L}^b)$ using the learning set $\mathcal{L}^b$ and its cluster labels.

(d) Apply the resulting classifier to the test set $\mathcal{T}^b$.

(e) Apply the clustering algorithm $\mathcal{P}$ to the test set $\mathcal{T}^b$ to obtain a partition $\mathcal{P}(\cdot, \mathcal{T}^b)$.

(f) Compute an external index $s_{k,b}$ comparing the two sets of labels for $\mathcal{T}^b$ obtained by clustering and prediction, respectively.

2. Let $t_k = \text{median}(s_{k,1}, \cdots, s_{k,B})$ denote the observed similarity statistic for the k–cluster partition of the data.

3. Generate B_0 datasets under a suitable null hypothesis. For each reference dataset, repeat the procedure described in steps 1 and 2 above, to obtain B_0 similarity statistics $t_{k,1}, \ldots, t_{k,B_0}$.

4. Let t_k^0 denote the average of these B_0 statistics, $t_k^0 = \frac{1}{B_0} \sum_{b=1}^{B_0} t_{k,b}$, and let p_k denote the proportion of the $t_{k,b}$, $1 \leq b \leq B_0$, that are at least as large as the observed statistic t_k, *i.e.*, the p-value for t_k. Finally, let $d_k = t_k - t_k^0$ denote the difference between the observed similarity statistic and its estimated expected value under the null hypothesis of $K = 1$.

Define the set K^- as

$$K^- = \Big\{2 \leq k \leq M : p_k \leq p_{max},\ d_k \geq d_{min}\Big\},$$

where p_{max} and d_{min} are preset thresholds (see Section 2.2.1). If this set is empty, estimate the number of clusters as $\hat{K} = 1$. Otherwise, let $\hat{K} = \text{argmax}_{k \in K^-} d_k$, *i.e.*, take the number of clusters $\hat{K}$ that corresponds to the largest significant difference statistic d_k.

2.2.1 Discussion of parameters

In this paper, the following decisions were made regarding the different parameters for the *Clest* algorithm.

Clest parameter	Value
Maximum number of clusters	$M = 10$ for microarray data
	$M = 5$ for simulated data
Number of learning/test set iterations	$B = 20$
Number of reference datasets	$B_0 = 20$
Size of learning sets $\mathcal{L}^b$	$2n/3$
Clustering algorithm	*PAM*
Classifier	linear discriminant analysis with diagonal covariance matrix – *DLDA*
Reference null distribution	uniformity hypothesis
External index	Fowlkes & Mallows [13] external index, *FM*
Maximum p-value	$p_{max} = 0.05$
Minimum difference statistic	$d_{min} = 0.05$

• *Clustering algorithm – partitioning around medoids, PAM.* The clustering algorithm *PAM* is used in this paper (see Section 1.2), but one should keep in mind that different clustering algorithms can generate different partitions of the same data, possibly leading to different inferences about the number of clusters.

• *Classifier – diagonal linear discriminant analysis, DLDA.* For multivariate normal class densities, *i.e.*, for $\mathbf{x}|y = k \sim \mathrm{N}(\mu_k, \mathbf{\Sigma}_k)$, the maximum likelihood (ML) discriminant rule is

$$\mathcal{C}(\mathbf{x}) = \text{argmin}_{1 \leq k \leq K} \left\{ (\mathbf{x} - \mu_k)\mathbf{\Sigma}_k^{-1}(\mathbf{x} - \mu_k)' + \log |\mathbf{\Sigma}_k| \right\}.$$

When the class densities have the same diagonal covariance matrix $\mathbf{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_p^2)$, the discriminant rule is linear and given by

$$\mathcal{C}(\mathbf{x}) = \text{argmin}_{1 \leq k \leq K} \sum_{j=1}^{p} \frac{(x_j - \mu_{kj})^2}{\sigma_j^2}.$$

For the corresponding sample ML discriminant rules, the population mean vectors and covariance matrices are estimated from a learning set by the sample mean vectors and covariance matrices, respectively: $\hat{\mu}_k = \bar{\mathbf{x}}_k$ and $\hat{\mathbf{\Sigma}}_k = \mathbf{S}_k$. For the constant covariance matrix case, the pooled estimate of the common covariance matrix is used: $\hat{\mathbf{\Sigma}} = \sum_k (n_k - 1)\mathbf{S}_k/(n - K)$, where n_k denotes the number of observations in class k and n is the total sample size. *DLDA* is a very simple classifier but it has been shown to perform well in complex situations, in particular, in an extensive study of discrimination methods for the classification of tumors using gene expression data (Dudoit *et al.* [11]).

• *Reference null distribution.* The reference datasets are generated under the uniformity hypothesis as in the *gap* statistic method (see Section 2.1.1).

• *External index.* All of the external indices described in Section 2.1.3 were considered. The Fowlkes & Mallows [13] *FM* index was found to be superior to the other indices when reference datasets are generated under the uniformity hypothesis (data not shown).

• *Threshold parameters, p_{max} and d_{min}.* This rule is *ad hoc* and can likely be improved upon. Nevertheless, it gives a satisfactory performance and is used in the absence of a better choice.

• *Number of iterations and reference datasets.* The *Clest* procedure is robust to the choice of B and B_0 (data not shown).

3 Improvement of clustering accuracy

For a given number of clusters K, the goal is to estimate for each observation its cluster label and, if possible, get a measure of confidence for this cluster assignment.

In discriminant analysis, it is well known that gains in accuracy can be obtained by aggregating predictors built from perturbed versions of the learning set [6, 7, 8, 15]. In the *b*ootstrap *agg*regat*ing* or *bagging* procedure (Breiman [6]), perturbed learning sets of the same size as the original learning set are formed by drawing at random with replacement from the learning set, *i.e.*, by forming non-parametric bootstrap replicates of the learning set. Predictors are built for each perturbed dataset and aggregated by plurality voting. A useful by-product of the voting are the *prediction votes*, which may be used to assess the confidence of predictions for individual observations (Dudoit *et al.* [11]). It is of interest to see whether the application of aggregation procedures can also improve the partitions created by an arbitrary clustering method.

Two applications of bagging, denoted by *Bag1* and *Bag2*, are considered here. In the first application, the clustering algorithm is repeatedly applied to each bootstrap sample and the final partition is obtained by plurality voting, *i.e.*, by taking the majority class label for each observation. A valuable by-product of this bootstrap procedure are the cluster votes for individual observations. The second bagging approach forms a new dissimilarity matrix by recording for each pair of observations the proportion of time they were clustered together in the bootstrap clusters (Breiman, pers. comm.). This new dissimilarity matrix is then used as an input to a clustering algorithm and the resulting partition is considered final. The partitioning clustering algorithm *PAM* (see Section 1.2) is used here, but *Bag1* and *Bag2* can be applied to an arbitrary clustering procedure.

3.1 Bagging a clustering algorithm, *Bag1*

For a fixed number of clusters K

1. Apply the clustering algorithm $\mathcal{P}$ to the original learning set $\mathcal{L}$ to obtain cluster labels $\mathcal{P}(\mathbf{x}_i, \mathcal{L}) = \hat{y}_i$ for each observation, $i = 1, \ldots, n$.

2. Form the bth bootstrap sample $\mathcal{L}^b = (\mathbf{x}_1^b, \ldots, \mathbf{x}_n^b)$.

3. Apply the clustering algorithm $\mathcal{P}$ to the perturbed learning set $\mathcal{L}^b$ and obtain cluster labels $\mathcal{P}(\mathbf{x}_i^b, \mathcal{L}^b)$ for each observation in $\mathcal{L}^b$.

4. Permute the cluster labels assigned to the perturbed learning set $\mathcal{L}^b$ so that there is maximum overlap with the original clustering of these observations. More specifically, let S_K denote the set of all permutations of the integers $1, \ldots, K$. Find the permutation $\tau^b \in S_K$ such that

$$\sum_{i=1}^{n} I\Big(\tau^b(\mathcal{P}(\mathbf{x}_i^b, \mathcal{L}^b)) = \mathcal{P}(\mathbf{x}_i^b, \mathcal{L})\Big) = \max_{\tau \in S_K} \sum_{i=1}^{n} I\Big(\tau(\mathcal{P}(\mathbf{x}_i^b, \mathcal{L}^b)) = \mathcal{P}(\mathbf{x}_i^b, \mathcal{L})\Big),$$

   where $I(\cdot)$ is the indicator function, equaling 1 if the condition in parentheses is true, and 0 otherwise, and let

$$\hat{y}_i^b = \tau^b(\mathcal{P}(\mathbf{x}_i^b, \mathcal{L}^b))$$

   denote the cluster label for the ith observation of the bth bootstrap sample.

5. Repeat steps 2 – 4 B times (here $B = 20$), and assign a bagged cluster label for each observation by majority vote, that is, let

$$\hat{y}_i^* = \text{argmax}_{1\leq k\leq K} \sum_{\{b:\ \mathbf{x}_i\in\mathcal{L}^b\}} I\Big(\tau^b(\mathcal{P}(\mathbf{x}_i, \mathcal{L}^b)) = k\Big).$$

   Also record a *cluster vote*, which is the proportion of votes in favor of the "winning" cluster assignment, that is,

$$CV(\mathbf{x}_i) = \frac{\max_{1\leq k\leq K} \sum_{\{b:\ \mathbf{x}_i\in\mathcal{L}^b\}} I\Big(\tau^b(\mathcal{P}(\mathbf{x}_i, \mathcal{L}^b)) = k\Big)}{\#\{b : \mathbf{x}_i \in \mathcal{L}^b\}}.$$

Note the alignment in Step 4 of the *Bag1* procedure: the labels of the observations from each of the perturbed datasets are permuted in such a way that there is the least disagreement between these labels and the original labels from the clustering applied to the entire dataset. The method described next bypasses this alignment step by considering pairs of observations, rather than individual observations, and by building a new dissimilarity matrix.

3.2 Bagging a clustering algorithm, *Bag2*

For a fixed number of clusters K

1. Initialize two $n \times n$ matrices $\mathbf{A} = (a_{ij})$ and $\mathbf{M} = (m_{ij})$ to zero.

2. Form the bth bootstrap sample $\mathcal{L}^b = (\mathbf{x}_1^b, \ldots, \mathbf{x}_n^b)$.

3. Apply the clustering algorithm $\mathcal{P}$ to the perturbed learning set $\mathcal{L}^b$ and obtain cluster labels $\mathcal{P}(\mathbf{x}_i^b, \mathcal{L}^b)$ for each observation in $\mathcal{L}^b$.

4. For each pair of observations, update the matrices $\mathbf{A}$ and $\mathbf{M}$ as follows:

$$a_{ij} \leftarrow a_{ij} + I\Big(\mathbf{x}_i \in \mathcal{L}^b,\ \mathbf{x}_j \in \mathcal{L}^b,\ \mathcal{P}(\mathbf{x}_i, \mathcal{L}^b) = \mathcal{P}(\mathbf{x}_j, \mathcal{L}^b)\Big),$$

$$m_{ij} \leftarrow m_{ij} + I\Big(\mathbf{x}_i \in \mathcal{L}^b,\ \mathbf{x}_j \in \mathcal{L}^b\Big).$$

5. Repeat steps 2 – 4 B times (here $B = 20$), and define a new dissimilarity matrix $\mathbf{D} = (d_{ij})$, by $d_{ij} = 1 - a_{ij}/m_{ij}$.

6. Cluster the n original observations on the basis of this new dissimilarity matrix.

Note that the clustering algorithm applied in Step 6 need not be the same as the algorithm applied in Step 3. Also, note that procedure *Bag2* does not directly produce cluster votes as *Bag1* does. Nevertheless, it is possible to assess the confidence of the refined cluster assignments using, for example, the silhouette widths in *PAM*.

4 Simulated data

The proposed methods for estimating the number of clusters and for cluster accuracy improvement are applied to simulated datasets. In Section 4.1, the *Clest* procedure is compared to existing approaches for estimating the number of clusters. In Section 4.2, the cluster bagging procedures *Bag1* and *Bag2* are applied to simulated data and compared, in terms of the accuracy of cluster assignments, to a single application of the clustering algorithm *PAM*. In addition, for *Bag1*, it is investigated how well the cluster vote of an observation reflects the accuracy of its assigned cluster label.

4.1 Estimating the number of clusters

4.1.1 Simulation models

Procedures for estimating the number of clusters in a dataset are evaluated using simulated data from a variety of models, including those considered by Tibshirani *et al.* [31]. The models used for comparison contain different numbers of overlapping and non-overlapping clusters, different numbers of variables, and a wide range of covariance matrix structures. In addition, a variable number of irrelevant or "noise" variables are included in the models. A noise variable is a variable whose distribution does not depend on the cluster label, and such variables are added to obscure the underlying clustering structure which is to be recovered.

• **Model 1**: *1 cluster in 10 dimensions.* $n = 200$ observations are simulated from the uniform distribution over the unit hypercube in $p = 10$ dimensions.

• **Model 2**: *3 clusters in 2 dimensions.* The observations in each of the three clusters are independent bivariate normal random variables with means (0,0), (0,5), and (5,-3), respectively, and identity covariance matrix. There are 25, 25, and 50 observations in each of the 3 clusters, respectively.

• **Model 3**: *4 clusters in 10 dimensions, 7 noise variables.* Each cluster is randomly chosen to have 25 or 50 observations and the observations in a given cluster are independently drawn from a multivariate normal distribution with identity covariance matrix. For each cluster, the cluster means for the first three variables are randomly chosen from a $\mathrm{N}(\mathbf{0}_3, 25\mathbf{I}_3)$ distribution, where $\mathbf{0}_p$ denotes a $1 \times p$ vector of zeros and $\mathbf{I}_p$ denotes the $p \times p$ identity matrix. The means for the remaining seven variables are 0. Any simulation where the Euclidean distance between the two closest observations belonging to different clusters is less than 1 is discarded.

• **Model 4**: *4 clusters in 10 dimensions.* Each cluster is randomly chosen to contain 25 or 50 observations, with means randomly chosen as $\mathrm{N}(\mathbf{0}_{10}, 3.6\mathbf{I}_{10})$. The observations in a given cluster are independently drawn from a normal distribution with identity covariance matrix and appropriate mean vector. Any simulation where the Euclidean distance between the two closest observations belonging to different clusters is less than 1 is discarded.

• **Model 5**: *2 elongated clusters in 3 dimensions.* Cluster 1 contains 100 observations generated as follows. Set $x_1 = x_2 = x_3 = t$, with t taking on equally spaced values from -0.5 to 0.5. Gaussian noise with standard deviation of .1 is then added to each variable. Cluster 2 is generated in the same way except that the value 10 is added to each variable. This results in two elongated clusters, stretching out along the main diagonal of a three-dimensional cube, with 100 observations each.

• **Model 6**: *2 elongated clusters in 10 dimensions, 7 noise variables.* The clusters are generated as in **Model 5**, but, in addition, 7 noise variables are simulated independently from a normal distribution with mean 0 and variance v^2 for the vth variable, $4 \leq v \leq 10$.

• **Model 7**: *2 overlapping clusters in 10 dimensions, 9 noise variables.* Each cluster contains 50 observations. The first variable in each of the two clusters is normally distributed with mean 0 and 2.5, respectively, and with variance 1. The remaining 9 variables are simulated from the $\mathrm{N}(\mathbf{0}_9, \mathbf{I}_9)$ distribution (independently of the first variable).

• **Model 8**: *3 overlapping clusters in 13 dimensions, 10 noise variables.* Each cluster contains 50 observations. The first three variables have a multivariate normal distribution with mean vectors (0,0,0), (2,-2,2), and (-2,2,-2), respectively, and covariance matrix $\mathbf{\Sigma}$, where $\sigma_{ii} = 1, 1 \leq i \leq 3$, and $\sigma_{ij} = 0.5, 1 \leq i \neq j \leq 3$. The remaining 10 variables are simulated independently from the $\mathrm{N}(\mathbf{0}_{10}, \mathbf{I}_{10})$ distribution.

Note that **Models 1**, **2**, **4**, and **5** were considered in Tibshirani *et al.* [31]. **Model 3** is similar to model 3 in [31], with the addition of seven noise variables. **Model 6** is the same as **Model 5**, with the addition of seven noise variables.

Fifty datasets were simulated from each model and the methods described in Section 2 were applied to the resulting datasets. We are primarily interested in comparing the percentage of simulations for which each procedure recovers the correct number of clusters, as this quantity reflects the accuracy of the procedure. However, for the purpose of future applications, it is useful to also know whether a method tends to underestimate or overestimate the true number of clusters. Hence, the full distribution of the number of clusters estimated by each method is presented in Table 2. Note that only the methods *Clest*, *gap*, *gapPC*, and *hart* have the capability to identify one cluster in the data.

4.1.2 Results

Figure 1 displays barplots for the percentage of simulations for which a given method correctly recovered the number of clusters for each of the eight models. Table 2 provides a more detailed account of the simulation results for each of the procedures. It can be seen that *Clest* gave uniformly good results over the range of models, its worst performance being for **Model 7** with two overlapping clusters. The rest of the methods failed for at least one of the eight models considered. The *gap* procedure failed twice (**Models 5** and **6**) and *gapPC* failed once (**Model 6**). Neither *gap* nor *gapPC* were able to identify the presence of the two clusters for **Model 6**, which is a model with two drawn-out clusters and seven noise variables with varying variances. Both *gap* and *gapPC* consistently estimated one cluster for this model, perhaps because both methods are based on the within clusters sums of squares and consequently are more affected by the variables with larger variances. In a majority of the simulations from **Model 7** *Clest*, *gap*, and *gapPC* failed to distinguish between one and two clusters. The simple *hart* index did well for this model. The rest of the procedures do not have by definition the ability to estimate one cluster and hence generally identified the two clusters. Interestingly, for **Model 8** with three overlapping clusters, *sil* and *ch* performed poorly, choosing two clusters in a majority of the simulations, while *hart* and *Clest* showed the best performance. Overall, most methods tended to underestimate more often than they overestimated the number of clusters, but the situation was reversed for *hart* and *kl*. For **Model 1** it is only fair to compare *Clest*, *gap*, *gapPC*, and *hart*, as the other methods only estimate $\hat{K} \geq 2$.

In summary, for the simulation models considered here, *Clest* was the most robust and accurate method, whereas *hart* showed the worst performance. *gapPC* was better than *gap* and the rest of the methods showed similar performance.

For a given model, it is of interest to consider the median value of the statistics used by each method to estimate the number of clusters. For each number of clusters k, the plots of the median values, over the 50 simulated datasets, of the *Clest* d-statistic, $gapPC$, and $\bar{sil}$ statistics are shown in Figures 2, 3, and 4, respectively. The d-statistic does not generally have local maxima except for **Model 5**. There, a local maximum appears at $K = 4$ clusters, but the global maximum occurs at $K = 2$. It can be seen that the ability of *Clest* to distinguish between one and two clusters is very low for **Model 7**; the median of the d_2 values is less than the significance cut-off d_{min} used in the *Clest* algorithm. Indeed, the results in Table 2 show that *Clest* identified two clusters for only 30% of the datasets simulated from **Model 7**. The figures suggest that for the majority of the models, the global maximum of the median d-statistic is more pronounced than the global maxima of the median $gapPC$ and $\bar{sil}$ statistics, respectively. This again suggests good robustness and accuracy properties for the *Clest* method.

4.2 Improvement of clustering accuracy

4.2.1 Simulation models

By and large, given the true number of clusters K, a single application of *PAM* was able to recover the true partitions in the datasets simulated from the models of the previous section. These models are thus unsuitable for comparing methods aimed at improving the accuracy of cluster assignments, and data should be simulated from models with a sufficient amount of overlap between the clusters in order to provide room for a possible improvement in accuracy.

In this section, observations for each cluster are generated independently from multivariate normal distributions. That is, for each cluster k, n_k independent observations are generated from $\mathrm{N}(\mu_k, \mathbf{\Sigma}_k)$, where μ_k and $\mathbf{\Sigma}_k$ denote respectively the $1 \times p$ mean vector and $p \times p$ covariance matrix for cluster k, $k = 1, \ldots, K$. The parameters of the models are set in such a way that the clusters are overlapping to a certain degree. Eight types of models, with varying number of variables, covariance matrix structure, and number of clusters are considered and listed in Table 3.

One hundred datasets were simulated for each model. For each dataset, three sets of cluster labels were obtained by applying the *PAM* clustering algorithm as well as the *Bag1* and *Bag2* bagging procedures with *PAM* and $B = 20$ bootstrap samples. The three partitions were compared to the true partition as follows. The assigned cluster labels of the observations were permuted in order to minimize the proportion of observations with cluster labels disagreeing with the true class labels (see Step 4 of the algorithm *Bag1* in Section 3.1). The resulting disagreement rate is referred to as the *clustering error rate* and the distribution of the error rates over the 100 realizations was compared between the three methods for each of the simulation models.

For the *Bag1* procedure we also investigated how well the cluster votes relate to the accuracy of individual cluster assignments. To this end, observations were binned by their cluster votes and in each bin the percentage of correctly and incorrectly labeled observations was examined. The bins corresponding to high cluster votes should contain a high percentage of correctly classified observations. The observations were also grouped according to the correctness of their assigned labels and the distributions of the cluster votes in the two groups were compared.

4.2.2 Results

Improvement of clustering accuracy. For each simulation model, Figure 5 displays boxplots of the clustering error rates computed over 100 simulations. For all models but **Model II** the results are shown for one value of the parameter Δ only. The clusterings produced by bagging procedures *Bag1* and *Bag2* were in general at least as accurate and often substantially more accurate than the clusterings resulting from a single application of the *PAM* algorithm. It can also be seen from Figure 5 that for most models considered, the *Bag1* procedure was slightly superior to *Bag2*.

To quantify the improvement of bagging over a single application of *PAM*, improvement statistics i_1 and i_2 were defined to represent the percentage change of the clustering error rate relative to a single application of *PAM*. That is, the *improvement statistic* i_j for *Bagj*, $j = 1, 2$, is defined as the ratio $(e_o - e_j)/e_0$, where e_0, e_1, and e_2 denote the median clustering error rates for *PAM*, *Bag1*, and *Bag2*, respectively, over the 100 simulated datasets. The improvement statistics are displayed above the boxplots in Figure 5.

Both bagging procedures showed the largest improvement over a single application of *PAM* for **Model II** with $\Delta = 6$. This model contains a large number of noise variables (99), with complete overlap between the clusters, and only one variable with no overlap between the clusters. A single application of *PAM* did not perform well in the presence of a large number of noise variables, while aggregation by bagging greatly improved the accuracy of the clustering. The improvement statistics for the bagging procedures were very small and sometimes negative for **Model II** with $\Delta = 3$ and **Model V** with $\Delta = 2$. For these models, aggregation had no impact on the quality of the partitions. In general, the improvement statistic rises as the separation between the clusters increases, unless the performance of a single application of *PAM* is nearly optimal (data not shown).

Cluster votes. Recall that cluster votes CV can be obtained as by-products of the plurality voting in the *Bag1* procedure. For each model, the observations were stratified according to whether they were correctly classified by *Bag1* or not, and the distributions of the cluster votes between the two types of observations were compared using boxplots in Figure 6. It can be seen that the cluster votes for correctly allocated observations are higher than those for incorrectly allocated ones. Another way to evaluate whether the cluster votes are good indicators of the accuracy of cluster assignments is to group the observations according to the value of their cluster votes and to consider the fraction of correctly allocated observations within each bin. The barplot of Figure 7 displays the proportion of correct allocations as a function of cluster votes for **Model VI**; this model was used for demonstration purposes because of the diversity of its cluster votes. Figure 7 shows that the proportion of misclassifications for an observation is inversely related to its cluster vote. Cluster votes are thus good indicators of the accuracy of a cluster assignment.

5 Microarray data

DNA microarrays are a new and promising biotechnology which allows the monitoring of expression levels in cells for thousands of genes simultaneously. Microarrays are being applied increasingly in biological and medical research to address a wide range of problems, such as the classification of tumors [1, 2, 16, 26, 27, 29]. A reliable and precise classification of tumors is essential for successful diagnosis and treatment of cancer. Current methods for classifying human malignancies rely on a variety of morphological, clinical, and molecular variables. In spite of recent progress, there are still uncertainties in diagnosis. Furthermore, it is likely that the existing classes are heterogeneous and comprise diseases which are molecularly distinct and follow different clinical courses. By allowing the monitoring of expression levels on a genomic scale, microarray experiments may lead to a more complete understanding of the molecular variations among tumors and hence to a finer and more reliable classification.

There are three main types of statistical problems associated with tumor classification: (i) the identification of new tumor classes using gene expression profiles - *cluster analysis*; (ii) the classification of malignancies into known classes - *discriminant analysis*; and (iii) the identification of "marker" genes that characterize the different tumor classes - *variable selection*. Microarray data present a "large p, small n" problem, that is, a very large number of variables (genes) relative to the number of observations (tumor samples). The publicly available datasets typically contain expression data on 5,000-10,000 genes for less than 100 tumor samples. Both numbers are expected to grow, the number of genes reaching on the order of 30,000-40,000, an estimate for the total number of genes in the human genome.

Applications of clustering methods to microarray data can be found in Alizadeh *et al.* [1], Alon *et al.* [2], Golub *et al.* [16], Ross *et al.* [29], Tibshirani *et al.* [31], and van der Laan & Bryan [33]. We refer the reader to Dudoit *et al.* [11] for a discussion of discriminant analysis in the context of microarray experiments.

In this section, our proposed clustering resampling methods are applied to gene expression data from four recently published cancer microarray studies: the lymphoma dataset of Alizadeh *et al.* [1], the leukemia (ALL/AML) dataset of Golub *et al.* [16], the 60 cancer cell line (NCI 60) dataset of Ross *et al.* [29], and the melanoma dataset of Bittner *et al.* [3]. Note that the expression levels are in general highly processed data: the raw data in a microarray experiment consist of image files, and important pre-processing steps include image analysis of these scanned images and normalization. Because we chose to use publicly available datasets, most of these decisions were beyond our control, and one should bear in mind that different pre-processing decisions can have a large impact on the measured expression levels (Yang *et al.* [34, 35]).

5.1 Data and pre-processing

5.1.1 Description of the datasets

Lymphoma. This dataset comes from a study of gene expression in the three most prevalent adult lymphoid malignancies: B-cell chronic lymphocytic leukemia (B-CLL), follicular lymphoma (FL), and diffuse large B-cell lymphoma (DLBCL) (see Alizadeh *et al.* [1] and `http://genome-www.stanford.edu/lymphoma` for a detailed description of the experiments). Gene expression levels were measured using a specialized cDNA microarray, the Lymphochip, containing genes that are preferentially expressed in lymphoid cells or which are of known immunological or oncological importance. In each hybridization, fluorescent cDNA targets were prepared from a tumor mRNA sample (red-fluorescent dye Cy5) and a reference mRNA sample derived from a pool of 9 different lymphoma cell lines (green-fluorescent dye Cy3). The cell lines in the common reference pool were chosen to represent diverse expression patterns, so that most spots on the array would exhibit a non-zero signal in the Cy3 channel. This study produced gene expression data for $p = 4,682$ genes in $n = 81$ mRNA samples. The mRNA samples comprise 29 cases of B-CLL, 9 cases of FL, and 43 cases of DLBCL. Alizadeh *et al.* [1] further demonstrated that the DLBCL class is heterogeneous and comprises two distinct subclasses of tumors with different clinical behaviors. The gene expression data are summarized by an $81 \times 4,682$ matrix $\mathbf{X} = (x_{ij})$, where x_{ij} denotes the base-2 logarithm of the Cy5/Cy3 background-corrected and normalized fluorescence intensity ratio for gene j in lymphoma sample i. The mean percentage of missing observations per array is 6.6% and missing data were imputed as outlined in Section 5.1.2. The data were standardized as described in Section 5.1.3.

Leukemia. The leukemia dataset is described in Golub *et al.* [16] and available at `http://waldo.wi.mit.edu/MPR/data_set_ALL_AML.html`. This dataset comes from a study of gene expression in two types of acute leukemias: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Gene expression levels were measured using Affymetrix high-density oligonucleotide arrays containing $p = 6,817$

human genes. The data comprise 47 cases of ALL (38 B-cell ALL and 9 T-cell ALL) and 25 cases of AML. Following Golub *et al.* (Pablo Tamayo, pers. comm.), three pre-processing steps were applied to the normalized matrix of intensity values available on the website (after pooling the 38 mRNA samples from the learning set and the 34 mRNA samples from the test set): (i) thresholding: floor of 100 and ceiling of 16,000; (ii) filtering: exclusion of genes with $\max / \min \leq 5$ or $(\max - \min) \leq 500$, where max and min refer respectively to the maximum and minimum intensities for a particular gene across the 72 mRNA samples; (iii) base-10 logarithmic transformation. The data are then summarized by a $72 \times 3,571$ matrix $\mathbf{X} = (x_{ij})$, where x_{ij} denotes the expression level for gene j in mRNA sample i. There are no missing values and the data were standardized as described in Section 5.1.3. Note that this standardization differs from the one described in Golub *et al.* [16].

NCI 60. In this study, cDNA microarrays were used to examine the variation in gene expression among the 60 cell lines from the National Cancer Institute's (NCI 60) anti-cancer drug screen (Ross *et al.* [29], `http://genome-www.stanford.edu/nci60`). The cell lines were derived from tumors with different sites of origin: 7 breast, 6 central nervous system (CNS), 7 colon, 6 leukemia, 8 melanoma, 9 non-small-cell-lung-carcinoma (NSCLC), 6 ovarian, 2 prostate, 8 renal, 1 unknown (ADR-RES). Gene expression was studied using microarrays with 9,703 spotted cDNA sequences. In each hybridization, fluorescent cDNA targets were prepared from a cell line mRNA sample (red-fluorescent dye Cy5) and a reference mRNA sample obtained by pooling equal mixtures of mRNA from 12 of the cell lines (green-fluorescent dye Cy3). To investigate the reproducibility of the entire experimental procedure (cell culture, mRNA isolation, labeling, hybridization, scanning, *etc.*), a leukemia (K562) and a breast cancer (MCF7) cell line were analyzed by three independent microarray experiments. Ross *et al.* screened out genes with missing data in more than two arrays. In addition, because of their small class size, the two prostate cell lines and the unknown cell line were excluded from our analysis. The data are summarized by a $61 \times 5,244$ matrix $\mathbf{X} = (x_{ij})$, where x_{ij} denotes the base-2 logarithm of the Cy5/Cy3 background-corrected and normalized fluorescence intensity ratio for gene j in cell line i. The mean percentage of missing observations per array is 3.3% and missing data were imputed as outlined in Section 5.1.2. The data were standardized as described in Section 5.1.3.

Melanoma. The melanoma dataset is described in the recent paper of Bittner *et al.* [3] and available at `http://www.nhgri.nih.gov/DIR/Microarray`. There are 31 melanoma samples and 7 control samples. Gene expression levels were measured using cDNA microarrays of 8,150 spots, representing 6,971 unique genes. In each hybridization, fluorescent cDNA targets were prepared from a melanoma or control mRNA sample (red-fluorescent dye Cy5) and a common reference mRNA sample (green-fluorescent dye Cy3). The following pre-processing steps were applied by Bittner *et al.*: (i) a gene was excluded from the analysis if its average mean intensity above background for the least intense signal (Cy3 or Cy5) across all experiments was $\leq 2,000$ or its average spot size across all experiments was ≤ 30 pixels; and (ii) a floor and ceiling of .02 and 50, respectively, were applied to the individual red and green intensities. This initial screening resulted in a dataset of 3,613 genes. Finally, Bittner *et al.* did not include the 7 control samples in their analysis. The data are summarized by a $31 \times 3,613$ matrix $\mathbf{X} = (x_{ij})$, where x_{ij} denotes the base-2 logarithm of the Cy5/Cy3 background-corrected and normalized fluorescence intensity ratio for gene j in mRNA sample i. There were no *a priori* known classes for this dataset, but the analysis of Bittner *et al.* suggests that two classes may be present in the data, with observations in one of the classes (Group A in the figures) being more tightly clustered. There were no missing values and the data were standardized as described in Section 5.1.3. Note that this standardization is slightly different from the one described in Bittner *et al.* [3].

5.1.2 Imputation of missing data

For the lymphoma and NCI 60 datasets, each array contains a number of genes with fluorescence intensity measurements that were flagged by the experimenter and recorded as missing data points. Missing data were imputed by a simple k nearest neighbor algorithm, in which the neighbors are the genes and the distance between neighbors is based on the correlation between their gene expression levels across arrays. For each gene with missing data: (i) compute its correlation with all other $p-1$ genes, and (ii) for each missing array, identify the k nearest genes having data for this array and impute the missing entry by the average of the corresponding entries for the k neighbors. A value of $k = 5$ neighbors was used for the lymphoma and NCI 60 datasets. For a detailed study of imputation methods in microarray experiments the reader is referred to the recent work of Troyanskaya *et al.* [32] which suggests that a nearest neighbor approach provides accurate and robust estimates of missing values.

5.1.3 Standardization

The gene expression data were standardized so that the observations (arrays) have mean 0 and variance 1 across variables (genes). Standardizing the data in this fashion achieves a location and scale normalization of the different arrays. In a study of normalization methods, we have found scale adjustment to be desirable in some cases, in order to prevent the expression levels in one particular array from dominating the average expression levels across arrays (Yang *et al.* [35]). Furthermore, this standardization is consistent with the common practice in microarray experiments of using the correlation between the gene expression profiles of two mRNA samples to measure their similarity [1, 26, 29].

5.1.4 Preliminary gene selection

Expression levels were monitored for thousands of genes in each of the four studies. However, the majority of the genes exhibit near constant expression levels, as measured by the variance of the expression levels across tumor samples. Genes showing nearly constant expression levels are not likely to be useful for classification purposes, therefore, we chose to exclude low variance genes from the clustering process.

Figure 8 displays for each dataset the individual gene variances divided by the maximum variance over all genes. All four variance curves show a sharp drop-off which gradually flattens. The plots are remarkably similar for all the datasets, with the melanoma dataset having the fastest drop-off. In this report, the $p = 100$ most variable genes were used to analyze the leukemia, lymphoma, and melanoma datasets, and the $p = 200$ most variable genes were used for the NCI 60 dataset as it contains more classes. Increasing the number of genes to $p = 300 - 400$ or decreasing the number of genes to $p = 50$ did not have much effect on the results (data not shown).

5.1.5 Correlation matrices

The following is not part of the cluster analysis *per se*, but is an interesting side-step which may be predictive of the results of the forthcoming analysis. Recall that for the first three datasets, tumor classes were known *a priori*, and for the melanoma dataset two classes were proposed by Bittner *et al.* [3]. For each dataset, images of the $n \times n$ correlation matrix for the n mRNA samples are displayed in Figures 9, 10, 11, and 12, with observations grouped according to their *a priori* known or proposed classes. Note that if observations are highly correlated within classes, the correlation image in this representation should exhibit bright red squares along the diagonal.

Lymphoma. The existence of three well separated classes for the lymphoma dataset is reflected in Figure 9 for both sets of genes, the classes being more clearly separated when the majority of the genes are screened out. Recall, that gene expression levels were measured using a specialized cDNA microarray, the Lymphochip, enriched in genes that are involved in the immune system. This may partly account for the clear separation of the classes even when the correlation matrix is computed using the full set of genes. When *PAM* is applied to the lymphoma dataset using the 100 genes with the largest variance, the $K = 2, 3, 4, 5$ partitions are as follows. For $K = 2$, one cluster consists of the FL and DLBCL classes combined and the other consists of the CLL class. This could reflect differences in tissue sampling, as the CLL mRNA samples were obtained from peripheral blood cells as opposed to lymph node biopsy specimens for the FL and DLBCL samples. For $K = 3$, all three classes (CLL, FL, DLBCL) are recovered as distinct clusters. For $K = 4$, the largest DLBCL class is divided into two clusters of approximately equal size and the remaining two classes (CLL and FL) are recovered as two distinct clusters. The two DLBCL clusters have a 75% overlap with the proposed subclasses of Alizadeh *et al.* [1]. Finally, for $K = 5$, the smallest class FL is divided into two clusters and the rest of the clusters are as with $K = 4$. Based on this analysis, we do not expect to recover more than 4 classes in the lymphoma data.

Leukemia. Images of the correlation matrix for the leukemia dataset are displayed in Figure 10. The three classes corresponding to the ALL T-cell, ALL B-cell, and AML samples clearly stand out in the image of the correlation matrix for the 100 genes with the largest variance, but are indistinguishable in the image of the correlation matrix based on all genes. When the *PAM* algorithm is applied to the leukemia dataset using the 100 genes with the largest variance, the results are as follows. For $K = 2$, 8 ALL T-cell observations are misallocated with the AML observations. For $K = 3$, one ALL B-cell sample is clustered with the ALL T-cell tumors and the rest of the observations are allocated correctly. For $K = 4$, the ALL B-cell samples are partitioned into two clusters. Finally, for $K = 5$, the AML samples are partitioned into two clusters. Based on the correlation matrix, one would expect to identify three tumor classes in this dataset.

NCI 60. For the NCI 60 cell line dataset, the classes are not clearly distinguishable from the images of the correlation matrix. The colon, leukemia, and melanoma cell lines display the strongest correlations within class, while the breast, NSCLC, and ovarian cell lines seem to be the most heterogeneous classes. When the *PAM* algorithm is applied to the NCI 60 dataset using the 200 genes with the largest variance and varying the number of clusters $K \leq 8$, only five types of cell lines tend to cluster together (CNS, colon, leukemia, melanoma, and renal cell lines). Based on this observation, one should not expect to recover more than 5 classes.

Melanoma. Finally, for the melanoma dataset, the image of the correlation matrix for the $p = 100$ most variable genes (Figure 12) could possibly suggest the existence of a subclass of tumors which includes the Group A samples of Bittner *et al.* [3]. However, some observations in this cluster (the first one from the left in particular) were not identified by Bittner *et al.* as being part of the tight cluster. Indeed, when *PAM* is applied to the melanoma dataset using the 100 genes with the largest variance, four additional observations are joined to the 19 observation cluster (Group A) proposed by Bittner *et al.*. Dividing the data into three clusters results in a split of the 19 observations into two clusters. One would expect to identify at most two or three classes for this dataset because of the small sample size.

5.2 Estimating the number of clusters

The existing and new methods of Section 2 were applied to estimate the number of clusters for each of the four microarray datasets; the results are presented in Table 4. Note the quotation marks for the "known" column in the table: the DLBCL class for the lymphoma dataset is likely to contain two subclasses and the two melanoma classes in Bittner *et al.* [3] were proposed but not confirmed.

The methods *Clest* and *sil* correctly estimated the presumed number of classes for all but the NCI 60 dataset, where both methods identified three clusters only. The *gap* and *gapPC* methods overestimated the number of clusters for all datasets, with the exception of *gapPC* identifying 8 clusters for the NCI 60 dataset. The *ch* method estimated 2 clusters for each of the four datasets, while *kl* and *hart* identified 4 classes for the lymphoma dataset.

For *Clest*, *gapPC*, and *sil*, we further investigated how the strength of the evidence for the estimated number of clusters varied between datasets. Figure 13 displays plots of the d_k, $gapPC_k$, and $\bar{sil}_k$ statistics *vs.* the number of clusters k. Error bars for d_k and $gapPC_k$ are based on the standard deviations of t_k and $\log \mathrm{tr}\mathbf{W}_k$ under their respective null distributions (Section 2). While the evidence for the existence of clusters is very strong for the lymphoma, leukemia, and NCI 60 datasets, the evidence for the two clusters in the melanoma dataset is much weaker. In particular, for *Clest*, the maximum value of the d_k statistic barely reaches the d_{min} threshold of .05. For the leukemia dataset, the d_k statistic clearly peaks at $k = 3$

clusters and drops off abruptly; for the lymphoma and NCI 60 datasets the decrease is more gradual. Note that according to *Clest* there was not enough evidence to identify the two DLBCL subclasses. Alizadeh *et al.* [1] identified these subclasses using subject matter knowledge to select the genes for the clustering procedure; here the genes were selected in an unsupervised manner.

5.3 Improvement of clustering accuracy

Recall that mRNA samples in the lymphoma, leukemia, and NCI 60 datasets were assigned class labels from the laboratory analyses of the tumor samples or from *a priori* knowledge of the cell lines. For the melanoma dataset, tumor class labels were obtained from the statistical analysis described in Bittner *et al.* [3]. In the discussion that follows, these class labels are treated as known.

The cluster bagging methods *Bag1* and *Bag2* of Section 3 were applied to the four microarray datasets. For the lymphoma, leukemia, and melanoma datasets, the number of clusters estimated by *Clest* agreed with the "known" number of clusters and that number was used as an input to *Bag1* and *Bag2*. In addition, for the lymphoma dataset, we investigated how the cluster votes changed when the number of clusters is increased to $K = 4$. The NCI 60 dataset comprises cell lines from 8 different sites of origin, but the methods *Clest* and *sil* estimated the number of clusters to be only 3. Therefore, the methods *Bag1* and *Bag2* were applied to the NCI 60 dataset with 3 and 8 clusters. The resulting cluster assignments and cluster votes for all four datasets are discussed next.

5.3.1 Lymphoma

The clustering algorithm *PAM* and the *Bag1* and *Bag2* procedures were applied to the lymphoma dataset with $K = 3$ clusters. All three clustering procedures recovered the known tumor classes (data not shown for *Bag2*). Figure 14 displays barplots for the cluster votes and silhouette widths, where the observations are color-coded by class and by whether the cluster assignment matched the known tumor class. The cluster votes are very high for all samples. Note that the silhouette widths are much more variable than the cluster votes. Figure 14 also displays cluster votes for *Bag1* with $K = 4$ clusters. While the cluster votes stay unchanged for the CLL and FL observations, they decrease for the DLBCL observations. The DLBCL class is split into two clusters, similar to the subclasses reported in Alizadeh *et al.* [1], and these do not seem as stable as the three original tumor classes.

5.3.2 Leukemia

The *PAM*, *Bag1*, and *Bag2* procedures were applied to the leukemia dataset with $K = 3$ clusters. A single application of *PAM* clustered one of the AML cases with the ALL T-cell cases; *Bag1* clustered one ALL T-cell case with the ALL B-cell cases and one ALL B-cell case with the AML cases; finally, *Bag2* misallocated the same two samples as *Bag1* and one of the samples misallocated by *PAM*. Note that the misallocated cases are the same as the ones that were hard to predict in the study of discrimination methods of Dudoit *et al.* [11].

Figure 15 displays barplots of the cluster votes and silhouette widths. Again, the silhouette widths are more variable than the cluster votes. Recall that a negative silhouette width indicates that the corresponding observation is closer to observations in a cluster other than its own, *i.e.*, its label is suspicious. A few observations that were correctly classified by a single application of *PAM* have negative or very small silhouette widths, and two of these observations were mislabeled by the procedure *Bag1* and carried low cluster votes. This raises the possibility that the tumors were misdiagnosed in the laboratory.

5.3.3 NCI 60

Recall that for this dataset three of the cell line classes (breast, NSCLC, and ovarian) are heterogeneous and cannot be identified with a single application of *PAM*. Here, the clustering algorithm *PAM* and the *Bag1* and *Bag2* procedures were applied with $K = 3$ and $K = 8$ clusters. For $K = 3$, all three procedures resulted in nearly identical partitions, and the cluster votes of the cell lines belonging to the 3 heterogeneous classes were lower than those of the cell lines belonging to the 5 more homogeneous classes. In particular, the eight melanoma cell lines had the highest cluster votes and the NSCLC cell lines had the lowest cluster votes (figure not shown).

Interestingly, when *Bag1* was applied to the NCI 60 dataset with $K = 8$ clusters, the final partition contained only 2 clusters. Although each application of PAM to a bootstrap learning set produced 8 clusters, the plurality voting eliminated unstable clusters. This suggests that the *Bag1* procedure may be able to correct for a misspecified number of clusters through the voting step.

5.3.4 Melanoma

The *PAM*, *Bag1*, and *Bag2* procedures were applied to the 31 melanoma observations with $K = 2$ clusters and the results compared to the cluster assignments of Bittner *et al.* [3]. Figure 16 displays barplots of the cluster votes and silhouette widths. The *Bag1* and *Bag2* partitions were identical. In general, the cluster votes for the melanoma dataset were lower than the cluster votes for the lymphoma and leukemia datasets. Several observations allocated by Bittner *et al.* to the small cluster were reclassified to the large cluster by *PAM*, *Bag1*, and *Bag2*. For instance, the first observation from the left was assigned the highest cluster vote and placed into the large cluster. Bittner *et al.* reclassified this observation as well in a later analysis (Radmacher, pers. comm.).

To date, the existence of the two melanoma classes and the correctness of the allocations have not been experimentally or clinically verified. Survival data is available on 15 patients as well as other clinical information. However, these data do not carry enough power to validate the allocation of the observations into the two clusters.

6 Discussion

Resampling methods such as bagging and boosting have been applied successfully in the context of discriminant analysis to improve prediction accuracy. In this paper, we have proposed resampling methods to address two main problems in cluster analysis: (i) estimating the number of clusters, if any, in a dataset; (ii) improving and assessing the accuracy of a given clustering procedure. Since the groups obtained from cluster analysis are often used later on for prediction purposes, the approaches to these two problems rely on and extend ideas from discriminant analysis. Although the methods are applicable to general clustering problems, particular attention is given to the clustering of tumors using gene expression data. The performance of the proposed and existing methods were compared using simulated data and gene expression data from four recently published cancer microarray studies.

6.1 Estimating the number of clusters

For problem (i), we proposed a prediction-based resampling method, *Clest*, which estimates the number of clusters K based on the reproducibility of cluster assignments. In the comparison studies of Sections 4 and 5, *Clest* was generally found to be more robust and accurate than six existing methods. For the simulated datasets, *Clest* performed well across a wide range of models with varying numbers of overlapping and non-overlapping clusters, different numbers of variables and covariance matrix structures. Unlike methods based on between or within clusters sums of squares, the resampling method seems robust to the varying covariance structure of the variables. For the microarray datasets, *Clest* and *sil* correctly estimated the number of clusters (as determined from *a priori* known or putative tumor and cell line classes) for three out of the four datasets; the performance of other methods was significantly worse.

A number of decisions were made regarding the different parameters of the *Clest* algorithm. The clustering (*PAM*) and prediction methods (*DLDA*) considered in this paper focus on similar features of the data, namely, the distance of the observations from cluster "centers". More work is needed to investigate the robustness of *Clest* to these choices. In particular, it would be interesting to consider prediction methods (*e.g.* classification trees) which focus on different aspects of the data than the clustering method. While it may appear that having a classifier as a further parameter of the algorithm creates more room for error, we have found that this is not the case in practice. When the classifier in *Clest* performs poorly, other methods for estimating the number of clusters also perform poorly. Another important choice in the *Clest* algorithm is the reference null distribution used to calibrate the observed similarity statistics t_k for different numbers of clusters. The uniformity hypothesis was used here, a natural alternative would be to consider random permutations of the variables, *i.e.*, permutations of the entries of the design matrix within columns. In *Clest*, the observed similarity statistics t_k are compared across numbers of clusters k by considering their distance from their expected value t_k^0 under the null distribution. A more sensitive calibration may be achieved by taking scale into account, *i.e.*, by dividing the difference statistic d_k by the standard deviation of t_k under the null, or even by considering p-values p_k for t_k. We briefly considered these refinements and found that on their own they did not allow good discrimination between the different ks. The *Clest* method does use however the idea of p-value in combination with the differences d_k, as it imposes an upper limit on the p-value p_k. Finally, the choice of cut-off parameters d_{min} and p_{max} were rather *ad hoc* and could be fine tuned.

Note that we have not considered model-based methods, such as the Bayesian approach of Fraley & Raftery [14]; we are currently setting up a new comparison study including such methods. Another issue which was only briefly addressed in this paper is the selection of variables on which to base the clusterings. For the microarray datasets, genes were selected based on the variance of their expression levels across samples and it was found that the clusterings were fairly robust to the number of genes.

6.2 Improvement of clustering accuracy

For problem (ii), a resampling method known as bagging in discriminant analysis is used to generate and aggregate multiple clusterings. Two applications of bagging were considered. In the first application, *Bag1*, the clustering algorithm is repeatedly applied to each bootstrap sample and the final partition is obtained by plurality voting. The second bagging procedure, *Bag2*, forms a new dissimilarity matrix by recording for each pair of observations the proportion of time they were clustered together in the bootstrap clusters. This new dissimilarity matrix is then used as an input to a clustering algorithm and the resulting partition is considered final.

For the microarray and simulated datasets considered in this study, the clusterings produced by bagging procedures *Bag1* and *Bag2* were in general at least as accurate and often substantially more accurate than the clusterings resulting from a single application of the *PAM* algorithm. Although the bagging procedures were illustrated using *PAM*, *Bag1* and *Bag2* are applicable to any clustering algorithm and it would be worthwhile to evaluate the improvement in accuracy for methods such as k-means or self-organizing maps. We suspect that, as in prediction, the increase in accuracy observed with *PAM* is due to a decrease in variability achieved by aggregating multiple clusterings. It would be interesting to carry out a more thorough study of the bias and variance properties of different clustering methods, as was done for classifiers in Breiman [7]. Other ongoing research directions include the investigation of different resampling schemes, similar in spirit to the adaptive resampling schemes used in boosting.

A valuable by-product of the *Bag1* procedure are the cluster votes which can be used to assess the confidence of cluster assignments for individual observations. Our study indicates that cluster votes are generally good indicators of the accuracy of a cluster assignment. In the context of tumor microarray data, samples with low cluster votes could be "flagged" and sent for new laboratory analyses. The cluster votes could also be used as weights when building predictors from the classes obtained by clustering. Note that we could also compute for a given observation the distribution of the cluster votes for each cluster and interpret the results as in *fuzzy clustering* (see Kaufman & Rousseeuw [20] for a discussion of fuzzy clustering).

An interesting feature of the *Bag1* procedure was raised in the application to the NCI 60 dataset using $K = 8$ clusters. Although each application of PAM to a bootstrap learning set produced 8 clusters, the plurality voting reduced the number of clusters to 2. This suggests that *Bag1* may be able to correct for a misspecified number of clusters by eliminating unstable clusters through the voting step. To investigate this more thoroughly one would need to carry out a simulation study in which the wrong number of clusters is given as an input to *Bag1*. We are also exploring other methods for "aligning" the original and bootstrap cluster labels in step 4 of the algorithm.

For most models considered in this study, the *Bag1* procedure was found to be slightly superior to *Bag2*. We are further exploring the general idea of creating a new dissimilarity matrix by resampling as in *Bag2*. Resampling could lead to dissimilarity matrices that are more robust to the initial choice of dissimilarity matrix and pre-processing decisions such as standardization. The new dissimilarity matrices could be used as inputs to other clustering algorithms than the one used on the bootstrap samples.

References

[1] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson Jr, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt. Different types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[2] U. Alon, N. Barkai D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci.*, 96:6745–6750, 1999.

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

[4] H-H Bock. On some significance tests in cluster analysis. *Journal of Classification*, 2:77–108, 1985.

[5] J. Breckenridge. Replicating cluster analysis: Method, consistency and validity. *Multivariate Behavioral Research*, 24:147–161, 1989.

[6] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[7] L. Breiman. Arcing classifiers. *Annals of Statistics*, 26:801–824, 1998.

[8] P. Buhlmann and B. Yu. Explaining bagging. *Annals of Statistics*, (Accepted).

[9] R. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3:1–27, 1974.

[10] D. Davies and D. Bouldin. A cluster separation measure. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1979.

[11] S. Dudoit, J. Fridlyand, and T. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, accepted for publication in 2001.

[12] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci.*, 95:14863–14868, 1998.

[13] E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78:553–584, 1983.

[14] C. Fraley and A. Raftery. How many clusters? which clustering method? – answers via model-based cluster analysis. Technical Report 329, University of Washington, Department of Statistics, 1998.

[15] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

[16] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M.L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[17] J. Hartigan. Asymptotic distributions for clustering criteria. *Annals of Statistics*, 6:117–131, 1978.

[18] J. A. Hartigan. Statistical theory in clustering. *Journal of Classification*, 2:63–76, 1985.

[19] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

[20] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York., 1990.

[21] W. Krzanowski and Y. Lai. A criterion for determining the number of groups in a dataset using sum of squares clustering. *Biometrics*, 44:23–34, 1985.

[22] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, Inc., San Diego, 1979.

[23] G. W. Milligan. Clustering validation: Results and implications for applied analyses. In P. Arabie, L. J. Hubert, and G. De Soete, editors, *Clustering and Classification*, pages 341–375. World Scientific Publishing Co., River Edge, NJ, 1996.

[24] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50:159–179, 1985.

[25] Panel on Discriminant Analysis, Classification, and Clustering. Discriminant analysis and clustering. *Statistical Science*, 4:34–69, 1989.

[26] C. M. Perou, S. S. Jeffrey, M. van de Rijn, C. A. Rees, M. B. Eisen, D. T. Ross, A. Pergamenschikov, C. F. Williams, S. X. Zhu, J. C. F. Lee, D. Lashkari, D. Shalon, P. O. Brown, and D. Botstein. Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proc. Natl. Acad. Sci.*, 96:9212–9217, 1999.

[27] J. R. Pollack, C. M. Perou, A. A. Alizadeh, M. B. Eisen, A. Pergamenschikov, C. F. Williams, S. S. Jeffrey, D. Botstein, and P. O. Brown. Genome-wide analysis of DNA copy-number changes using cDNA microarrays. *Nature Genetics*, 23:41–46, 1999.

[28] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[29] D. T. Ross, U. Scherf, M. B. Eisen, C. M. Perou, P. Spellman, V. Iyer, S. S. Jeffrey, M. Van de Rijn, M. Waltham, A. Pergamenschikov, J. C. F. Lee, D. Lashkari, D. Shalon, T. G. Myers, J. N. Weinstein, D. Botstein, and P. O. Brown. Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics*, 24:227–234, 2000.

[30] W. Sarle. Cubic clustering criterion. Technical Report A-108, SAS Institute, Inc., 1983.

[31] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic. Technical report, Stanford University, Department of Biostatistics, March 2000.

[32] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, In press.

[33] M. J. van der Laan and J. Bryan. Gene expression analysis with the parametric bootstrap. *Biostatistics*, In press.

[34] Y. H. Yang, M. J. Buckley, S. Dudoit, and T. P. Speed. Image processing on cDNA microarray data. *Journal of Computational and Graphical Statistic*, Accepted.

[35] Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed. Normalization for cDNA microarray data. In Michael L. Bittner, Yidong Chen, Andreas N. Dorsel, and Edward R. Dougherty, editors, *Microarrays: Optical Technologies and Informatics*, volume 4266 of *Proceedings of SPIE*, May 2001.

Table 2: *Estimating the number of clusters, results for simulated data.* For each simulation model, the distribution of the estimated number of clusters is recorded for each method. The true number of clusters is denoted by "*" and the modes for the distribution of the 50 estimates are indicated in bold for each method. Note that *sil*, *ch*, and *kl* do not have the ability to estimate $\hat{K} = 1$ cluster.

Method	Number of clusters, $\hat{K}$					
Model 1						
	1*	2	3	4	5	> 5
Clest	**48**	2	0	0	0	0
gap	**48**	0	1	1	0	0
gapPC	**50**	0	0	0	0	0
sil	–	**37**	6	4	3	0
ch	–	**42**	7	1	0	0
kl	–	**12**	**14**	**11**	**13**	0
hart	0	5	**22**	16	7	0
Model 2						
	1	2	3*	4	5	> 5
Clest	0	1	**49**	0	0	0
gap	0	0	**50**	0	0	0
gapPC	0	0	**50**	0	0	0
sil	–	5	**45**	0	0	0
ch	–	0	**50**	0	0	0
kl	–	0	**41**	2	7	0
hart	0	0	0	2	2	**46**
Model 3						
	1	2	3	4*	5	> 5
Clest	0	1	20	**29**	0	0
gap	0	1	16	**33**	0	0
gapPC	0	1	12	**37**	0	0
sil	–	17	**24**	9	0	0
ch	–	8	**20**	**22**	0	0
kl	–	3	11	**35**	1	0
hart	0	0	8	**42**	0	0
Model 4						
	1	2	3	4*	5	> 5
Clest	0	0	1	**49**	0	0
gap	0	0	0	**50**	0	0
gapPC	0	0	1	**49**	0	0
sil	–	5	8	**37**	0	0

continued on next page

continued from previous page						
Method	Number of clusters, $\hat{K}$					
ch	–	5	7	**38**	0	0
kl	–	0	1	**49**	0	0
hart	0	0	0	**50**	0	0
Model 5						
	1	2*	3	4	5	> 5
Clest	0	**44**	0	6	0	0
gap	0	0	0	19	**31**	0
gapPC	0	**50**	0	0	0	0
sil	–	**50**	0	0	0	0
ch	–	3	0	**47**	0	0
kl	–	**50**	0	0	0	0
hart	0	0	0	0	0	**50**
Model 6						
	1	2*	3	4	5	> 5
Clest	0	**43**	7	0	0	0
gap	**47**	3	0	0	0	0
gapPC	**43**	5	1	1	0	0
sil	–	**41**	5	4	0	0
ch	–	**43**	5	2	0	0
kl	–	**16**	9	**17**	8	0
hart	0	1	0	5	14	**30**
Model 7						
	1	2*	3	4	5	> 5
Clest	**26**	15	6	3	0	0
gap	**25**	**22**	2	1	0	0
gapPC	**31**	17	2	0	0	0
sil	–	**42**	6	1	1	0
ch	–	**39**	10	0	1	0
kl	–	**13**	**15**	**10**	**12**	0
hart	6	**39**	5	0	0	0
Model 8						
	1	2	3*	4	5	> 5
Clest	0	16	**34**	0	0	0
gap	0	22	**28**	0	0	0
gapPC	0	**28**	21	1	0	0
sil	–	**50**	0	0	0	0
ch	–	**50**	0	0	0	0
kl	–	**25**	17	4	4	0
hart	0	3	**43**	4	0	0

Table 3: *Improvement of clustering accuracy, description of simulation models.*

Model	Cluster mean vectors	Cluster covariance matrices	Cluster sizes	Parameter Δ
Model I				
$K=2$	$\mu_1=(0,0)$	$\mathbf{\Sigma}=\mathbf{I}_2$	$n_1=50$	$1,3,6$
$p=2$	$\mu_2=(0,\Delta)$		$n_2=50$	
Model II				
$K=2$	$\mu_1=(0,\mathbf{0}_{99})$	$\mathbf{\Sigma}=\mathbf{I}_{100}$	$n_1=50$	$3,6$
$p=100$	$\mu_2=(\Delta,\mathbf{0}_{99})$		$n_2=50$	
Model III				
$K=3$	$\mu_1=\mathbf{0}_{13}$		$n_1=50$	$1.5,2$
$p=13$	$\mu_2=(\Delta,-\Delta,\Delta,\mathbf{0}_{10})$	$\mathbf{\Sigma}=\begin{pmatrix}\mathbf{A}_3 & \mathbf{0}_{3,10}\\ \mathbf{0}_{10,3} & \mathbf{I}_{10}\end{pmatrix}$	$n_2=50$	
	$\mu_3=-\mu_2$		$n_3=50$	
Model IV				
$K=3$	$\mu_1=\mathbf{0}_{13}$		$n_1=50$	2
$p=13$	$\mu_2=(\Delta,-\Delta,\Delta,\mathbf{0}_{10})$	$\mathbf{\Sigma}=\mathbf{A}_{13}$	$n_2=50$	
	$\mu_3=-\mu_2$		$n_3=50$	
Model V				
$K=3$	$\mu_1=\mathbf{0}_{16}$		$n_1=50$	2
$p=16$	$\mu_2=(\Delta,0,\Delta,0,\Delta,0,\mathbf{0}_{10})$	$\mathbf{\Sigma}=\begin{pmatrix}\mathbf{A}_6 & \mathbf{0}_{6,10}\\ \mathbf{0}_{10,6} & \mathbf{I}_{10}\end{pmatrix}$	$n_2=50$	
	$\mu_3=(0,-\Delta,0,-\Delta,0,-\Delta,\mathbf{0}_{10})$		$n_3=25$	
Model VI				
$K=3$	$\mu_1=\mathbf{0}_{15}$	$\mathbf{\Sigma}=\begin{pmatrix}\mathbf{B}_5 & \mathbf{0}_{5,10}\\ \mathbf{0}_{10,5} & \mathbf{I}_{10}\end{pmatrix}$	$n_1=50$	$1.5,2$
$p=15$	$\mu_2=(\Delta,\Delta,\Delta,\Delta,\Delta,\mathbf{0}_{10})$		$n_2=50$	
	$\mu_3=-\mu_2$		$n_3=50$	
Model VII				
$K=3$	$\mu_1=\mathbf{0}_{15}$	$\mathbf{\Sigma}=\begin{pmatrix}\mathbf{C} & \mathbf{0}_{5,10}\\ \mathbf{0}_{10,5} & \mathbf{I}_{10}\end{pmatrix}$	$n_1=50$	2
$p=15$	$\mu_2=(\Delta,\Delta,\Delta,\Delta,\Delta,\mathbf{0}_{10})$		$n_2=50$	
	$\mu_3=-\mu_2$		$n_3=50$	
Model VIII				
$K=2$	$\mu_1=\mathbf{0}_{15}$	$\mathbf{\Sigma}_1=\begin{pmatrix}\mathbf{C} & \mathbf{0}_{5,10}\\ \mathbf{0}_{10,5} & \mathbf{I}_{10}\end{pmatrix}$	$n_1=50$	2
$p=15$	$\mu_2=(\Delta,\Delta,\Delta,\Delta,\Delta,\mathbf{0}_{10})$	$\mathbf{\Sigma}_2=\begin{pmatrix}\mathbf{D} & \mathbf{0}_{5,10}\\ \mathbf{0}_{10,5} & \mathbf{I}_{10}\end{pmatrix}$	$n_2=50$	

Here, $\mathbf{0}_{m,n}$ is an $m\times n$ matrix of zeros; $\mathbf{A}_p$ is the $p\times p$ matrix such that $a_{ii}=1$, and

$a_{ij} = 0.5, i \neq j$; $\mathbf{B}_p$ is the $p \times p$ matrix such that $b_{ii} = 1$, $b_{i\,i+1} = b_{i\,i-1} = 0.5$, and $b_{ij} = 0.1, j \neq i-1, i, i+1$;

$$\mathbf{C} = \begin{pmatrix} 0.5 & 0.5 & -0.1 & -0.1 & -0.1 \\ 0.5 & 1.0 & 0.5 & -0.1 & -0.1 \\ -0.1 & 0.5 & 1.5 & 0.5 & -0.1 \\ -0.1 & -0.1 & 0.5 & 1.0 & 0.5 \\ -0.1 & -0.1 & -0.1 & 0.5 & 0.5 \end{pmatrix}, \text{ and } \mathbf{D} = \begin{pmatrix} 1.0 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 2.0 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 1.0 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 2.0 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 1.0 \end{pmatrix}.$$

Table 4: *Estimating the number of clusters, results for microarray data.*

Dataset	"known"	*Clest*	*gap*	*gapPC*	*sil*	*ch*	*kl*	*hart*
Lymphoma	**3**	3	10	8	3	2	4	4
Leukemia	**3**	3	10	5	3	2	3	3
NCI 60	**8**	3	10	8	3	2	6	2
Melanoma	**2**	2	9	4	2	2	8	1

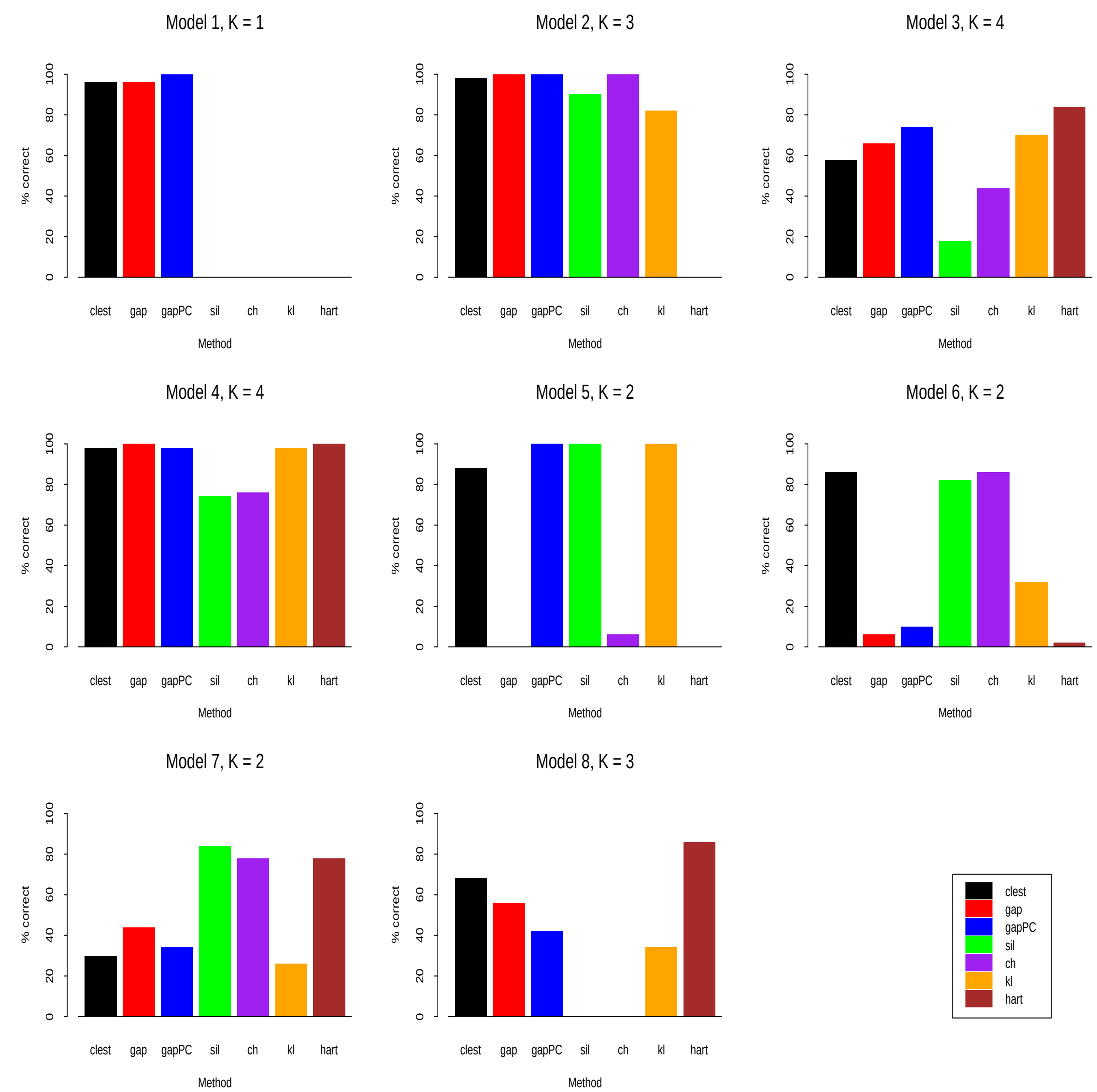


Figure 1: *Estimating the number of clusters, results for simulated data* – For each of the eight simulation models, the barplots represent the percentage of simulations for which the number of clusters was correctly estimated by each method (out of 50 simulations).

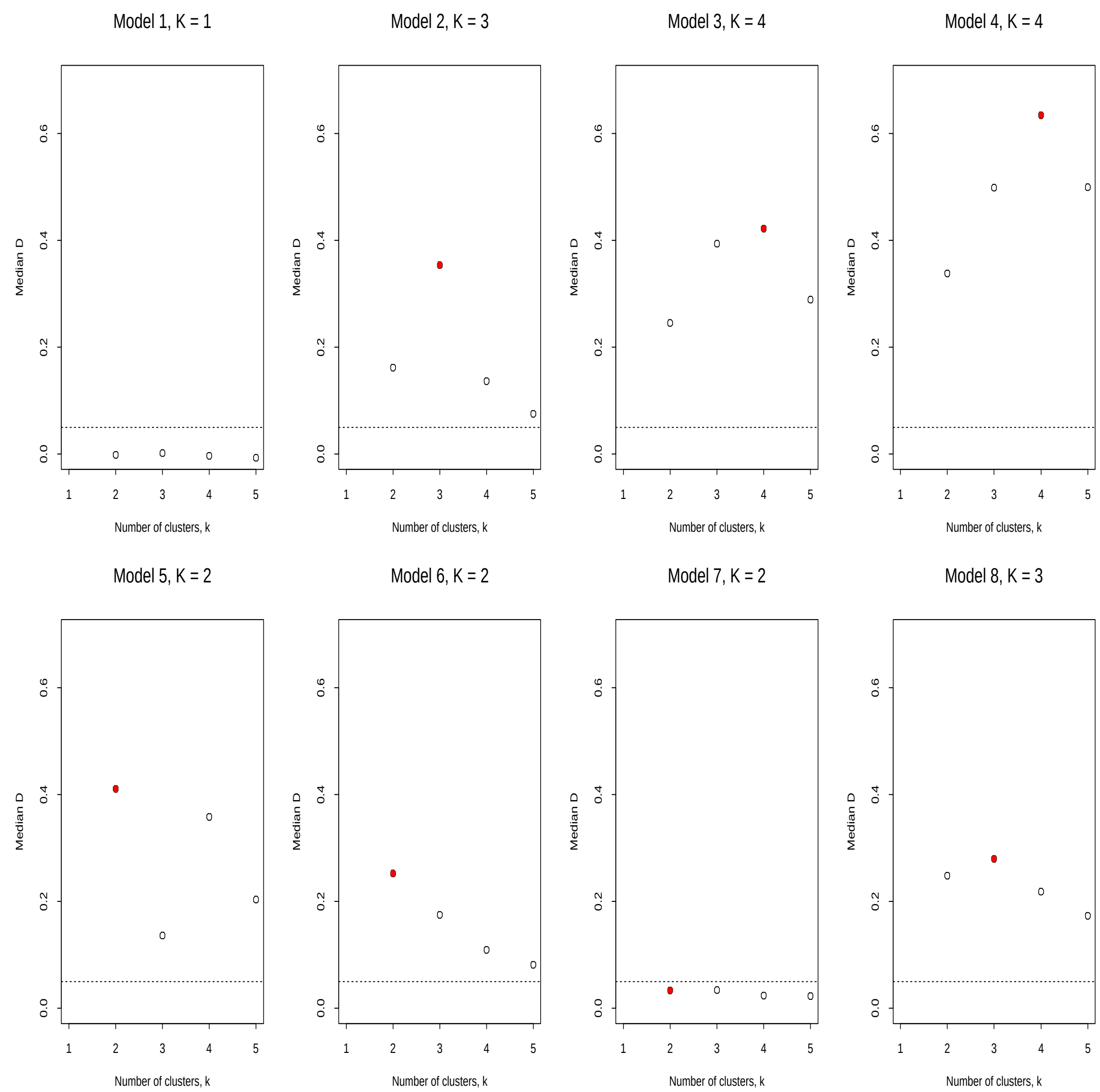


Figure 2: *Estimating the number of clusters, results for simulated data* – For the *Clest* procedure, plots of median d_k *vs.* k for each simulation model (medians are computed over 50 simulations). The horizontal line corresponds to the d_{min} cut-off of 0.05, and the true number of clusters is indicated by a filled plotting symbol.

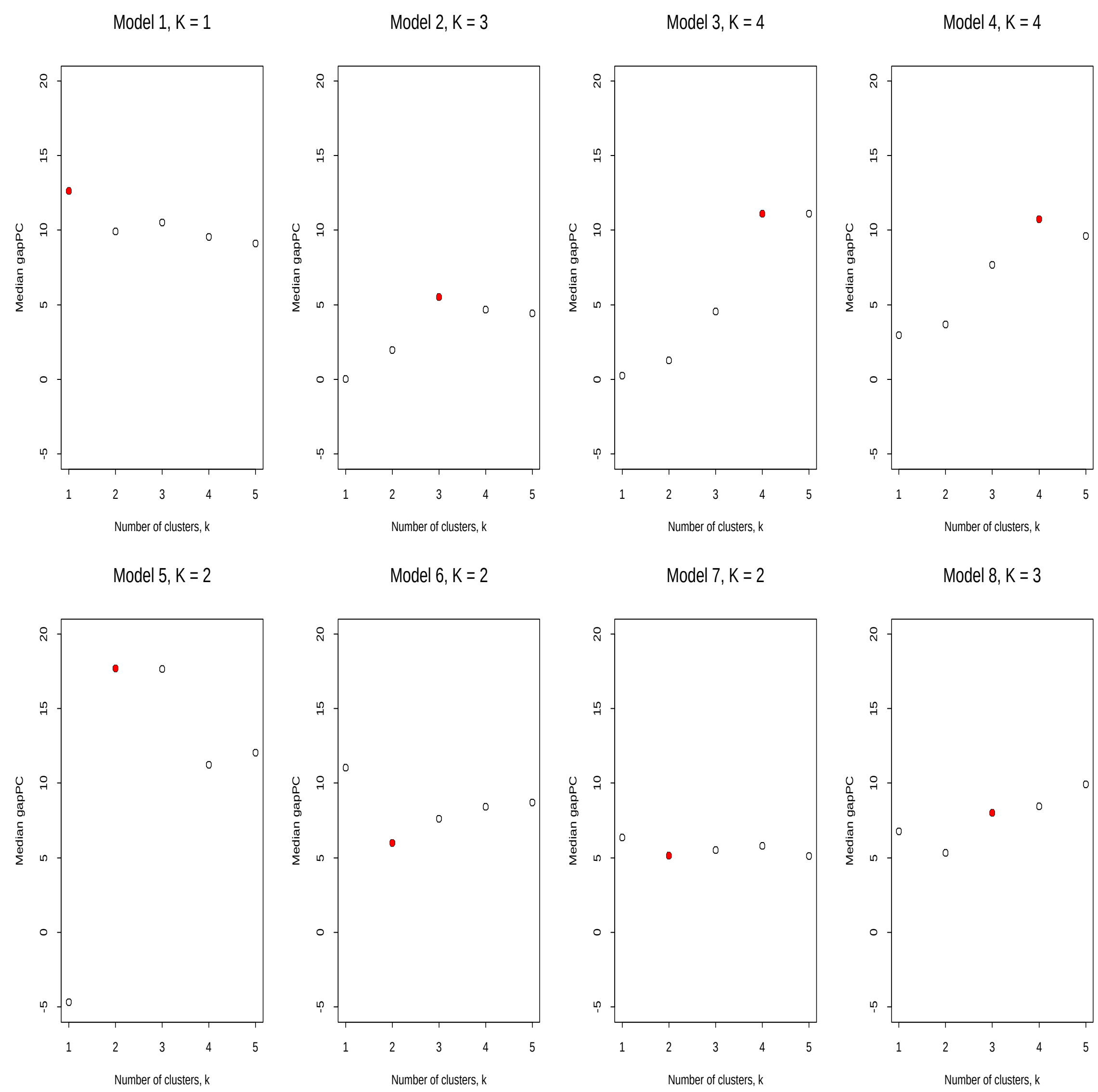


Figure 3: *Estimating the number of clusters, results for simulated data* – For the *gapPC* procedure, plots of median $gapPC_k$ *vs.* k for each simulation model (medians are computed over 50 simulations). The true number of clusters is indicated by a filled plotting symbol.

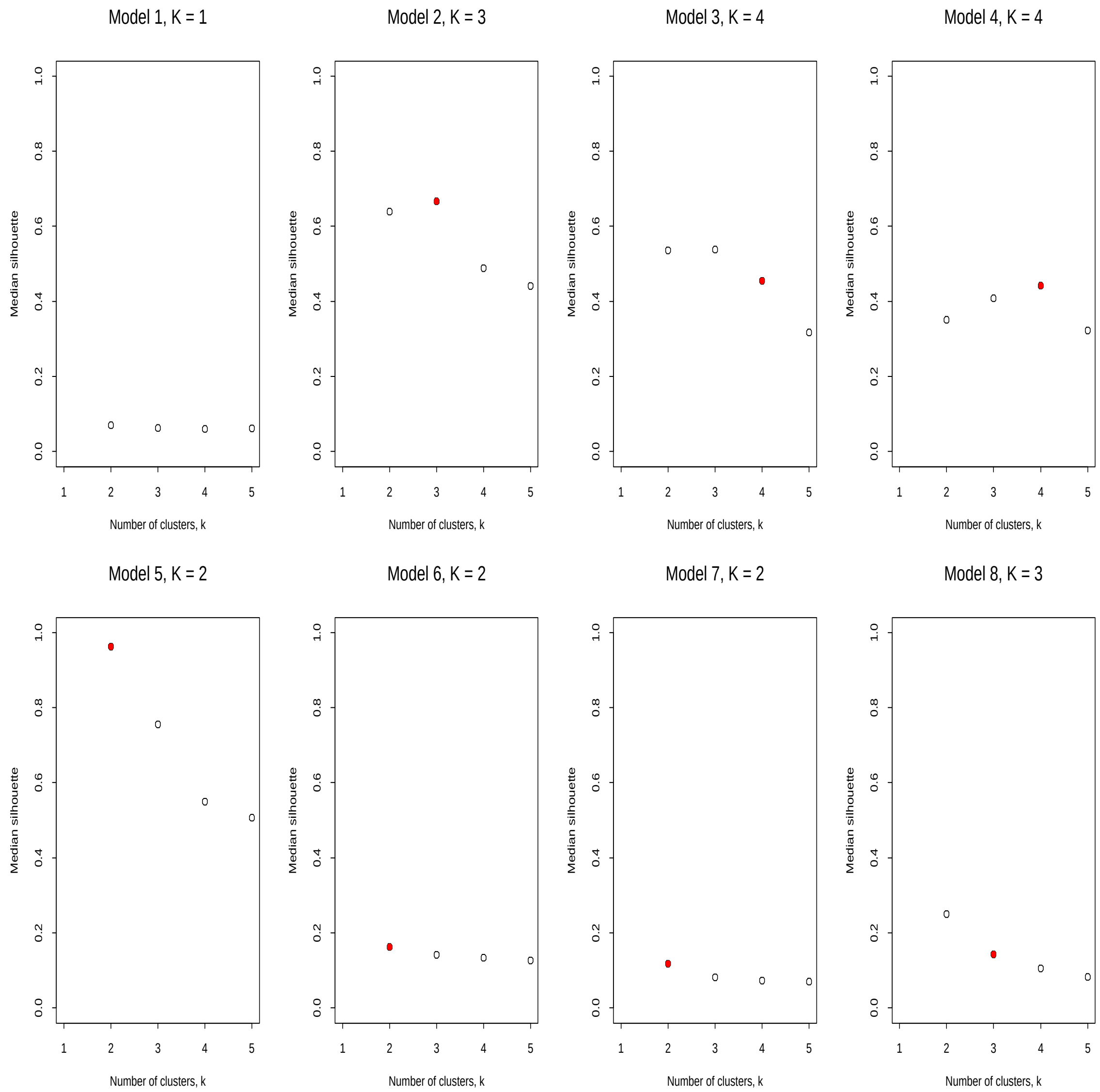


Figure 4: *Estimating the number of clusters, results for simulated data* – For the *sil* procedure, plots of median $\bar{sil}_k$ *vs.* k for each simulation model (medians are computed over 50 simulations). The true number of clusters is indicated by a filled plotting symbol.

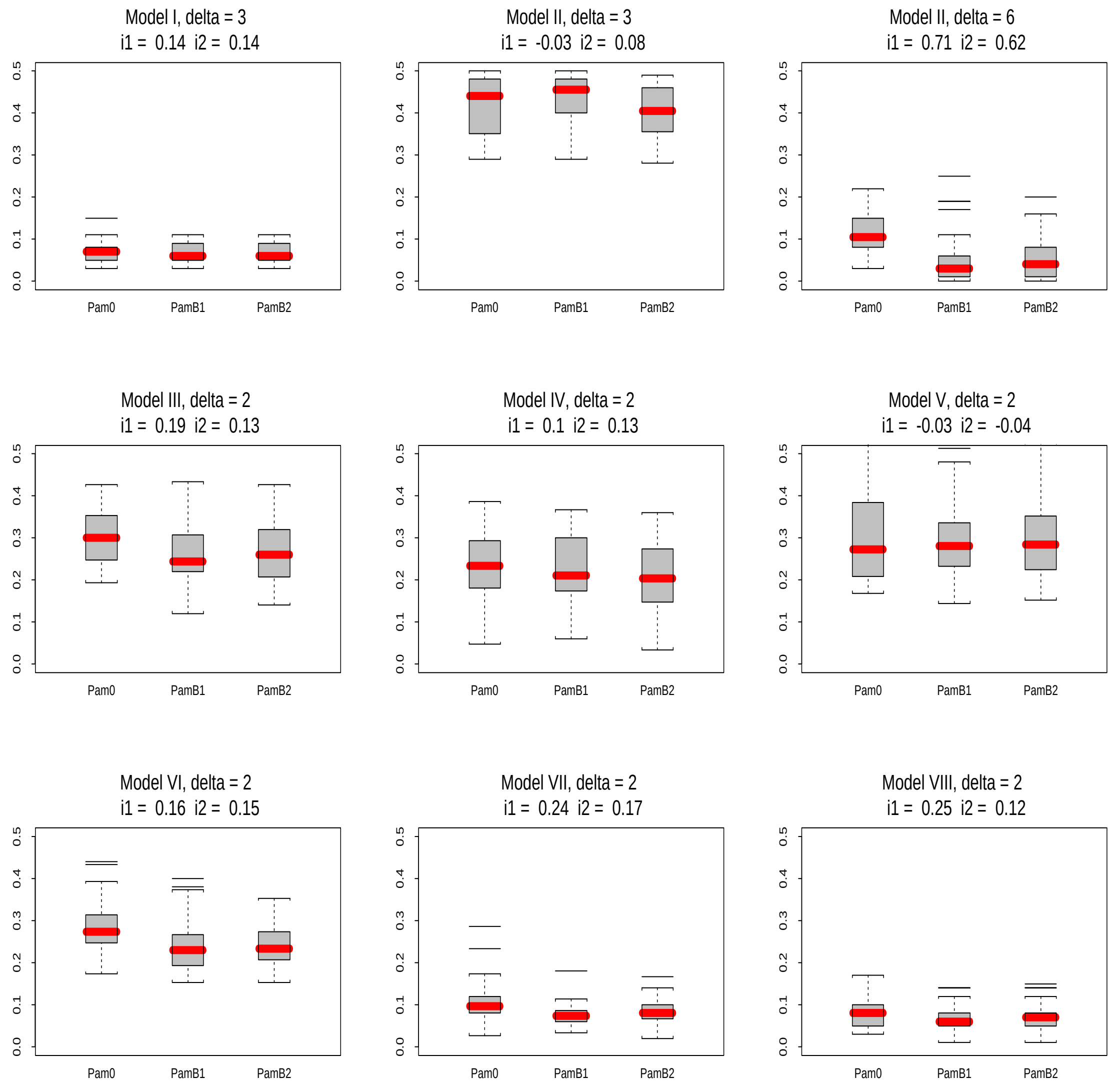


Figure 5: *Improvement of clustering accuracy, results for simulated data* – Boxplots of the clustering error rates (over 100 simulations) for a single application of *PAM* (Pam0), and bagging procedures *Bag1* (PamB1) and *Bag2* (PamB2). Clustering error rates and improvement statistics i_1 and i_2 are defined in Section 4.2.

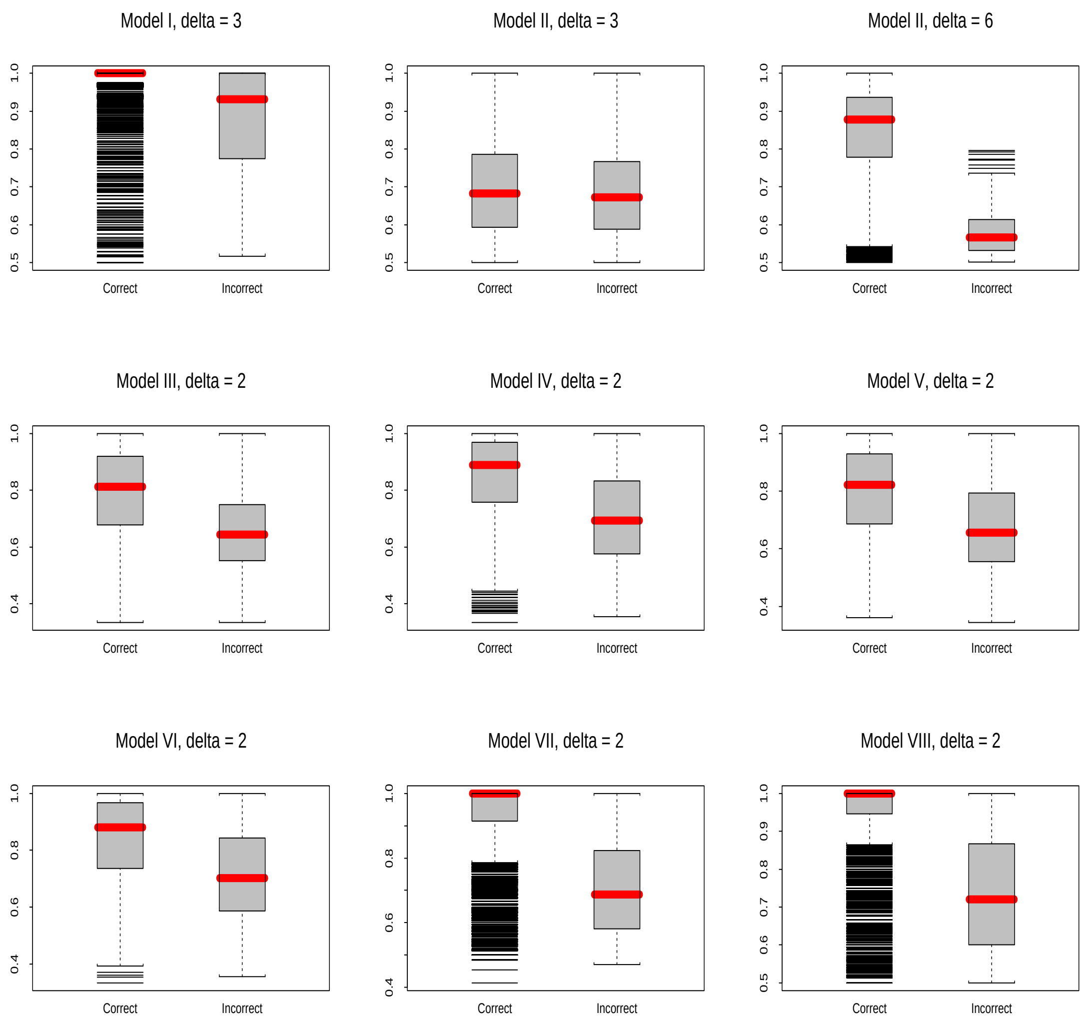


Figure 6: *Cluster votes, results for simulated data* – Boxplots of cluster votes stratified according to correct and incorrect allocations. (The number of cluster votes considered for each model is equal to the number of observations n times the number of simulations, here 100.)

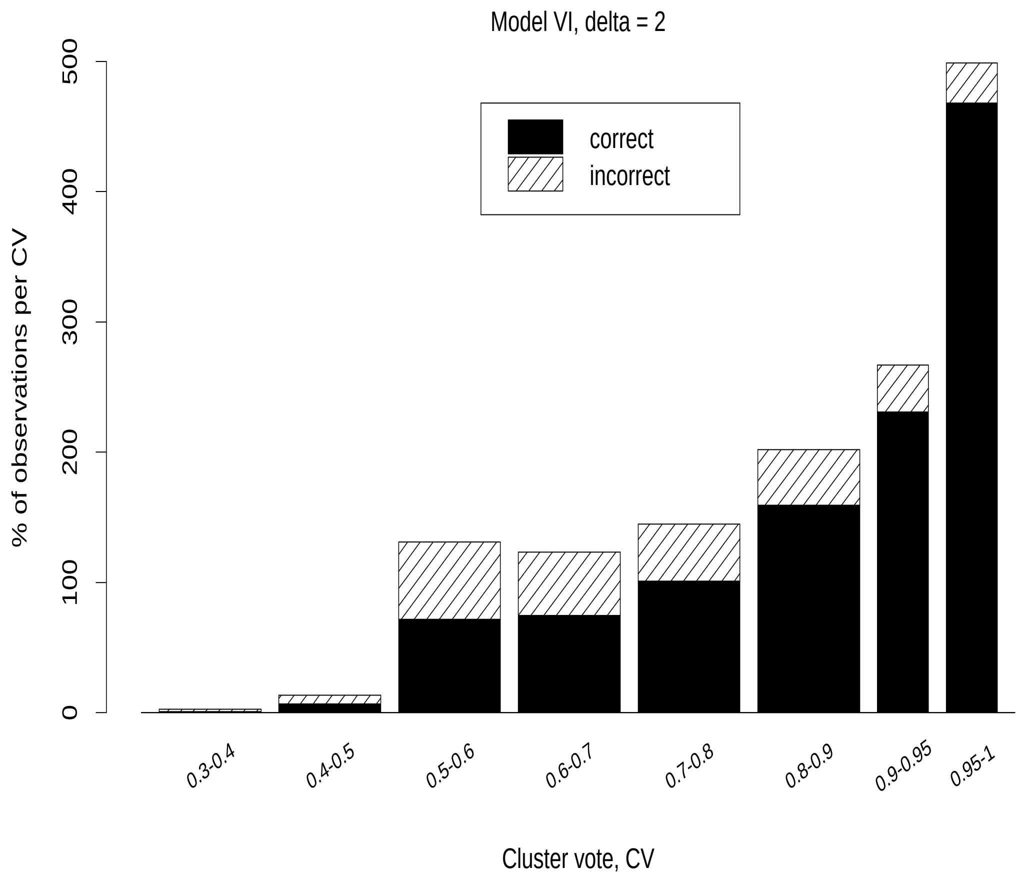


Figure 7: *Cluster votes, results for simulated data* – Barplot of cluster votes for **Model VI** with $\Delta = 2$; correct cluster allocations are represented with black bars, incorrect cluster allocations with shaded bars . (The total number of cluster votes is equal to the number of observations $n = 150$ times the number of simulations, here 100.)

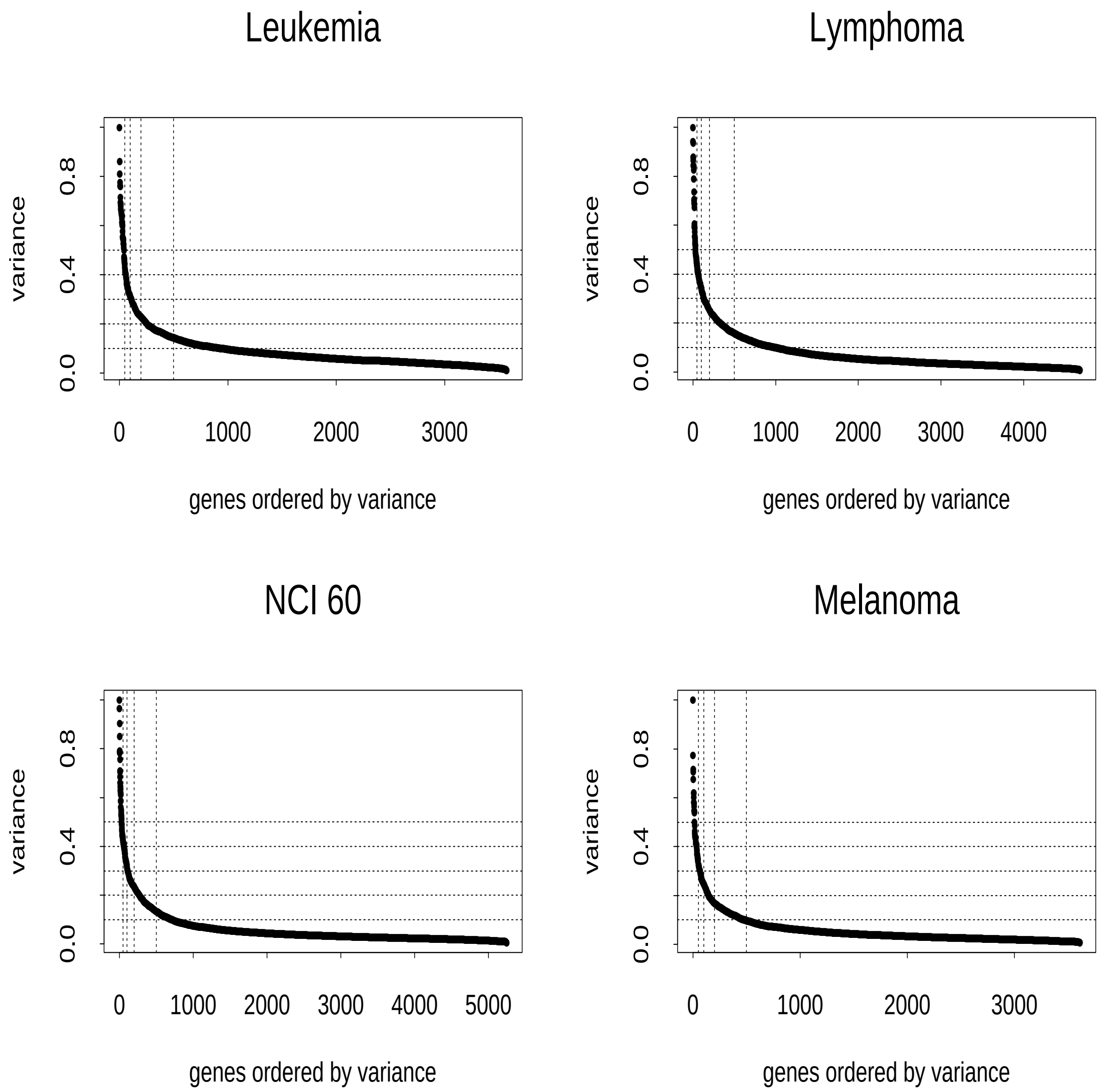
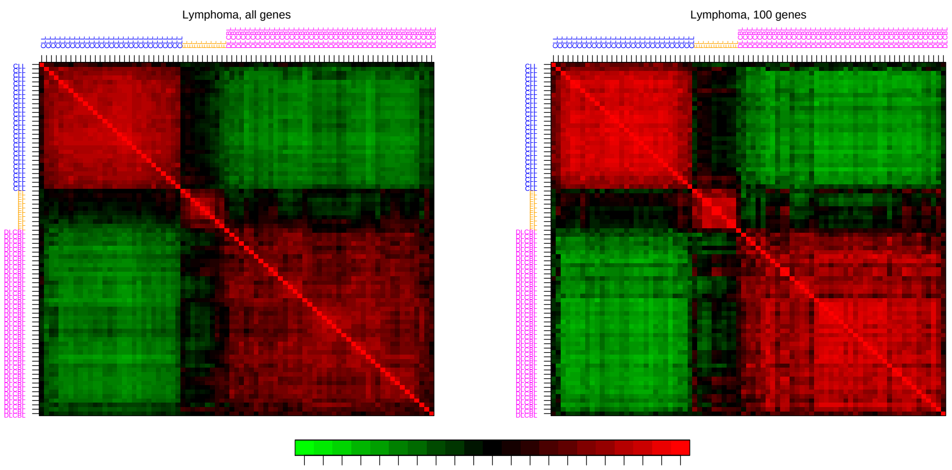
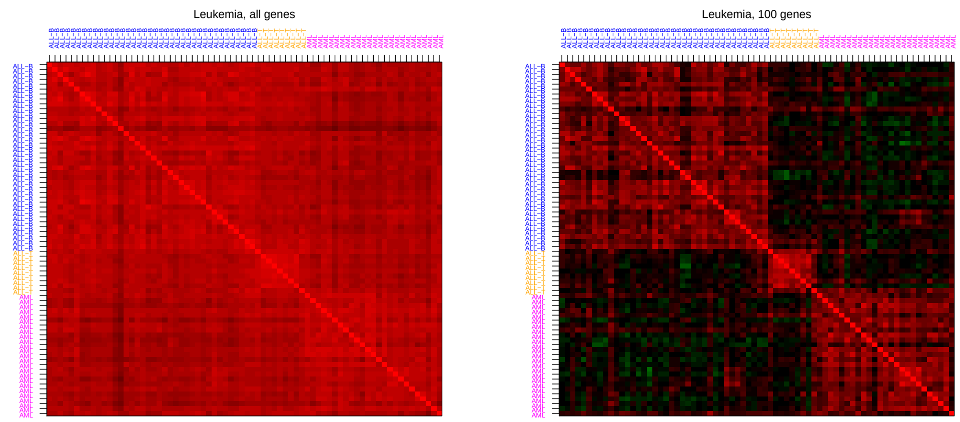


Figure 8: *Gene variances for microarray datasets* – Plots of the variance of the expression levels of each gene across mRNA samples. The variances are scaled by the maximum variance over all genes and the genes are ordered by variance in descending order. The vertical lines correspond to 50, 100, 200, and 500 genes, and the horizontal lines correspond to ratios of variances of 0.1, 0.2, 0.3, 0.4, and 0.5.



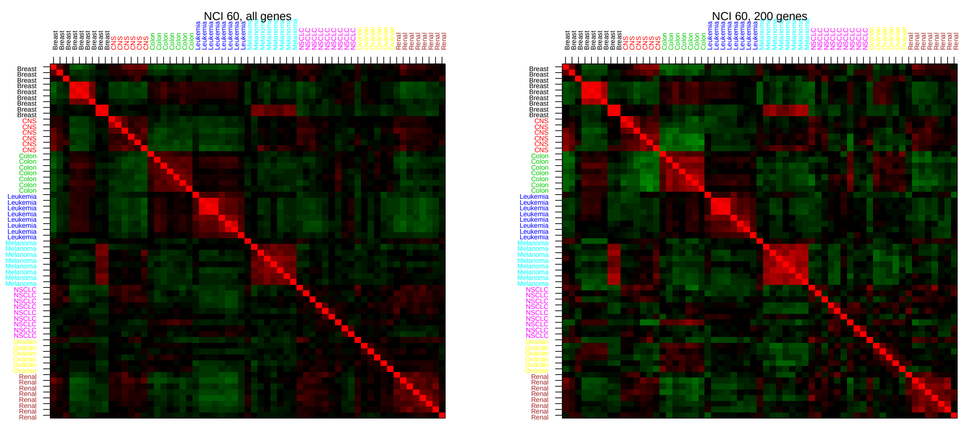
(a) (b)

Figure 9: *Correlation matrix, lymphoma dataset* – Images of the correlation matrix for the 81 B-CLL, FL, and DLBCL samples based on expression profiles for all $p = 4,682$ genes (panel (a)) and for the $p = 100$ genes with the largest variance (panel (b)). The mRNA samples are ordered by class, first B-CLL (blue), then FL (orange), and finally DLBCL (magenta). Correlations of zero are represented in black, increasingly positive correlations are represented with reds of increasing intensity, and increasingly negative correlations are represented with greens of increasing intensity. The color bar below the images may be used for calibration purposes.



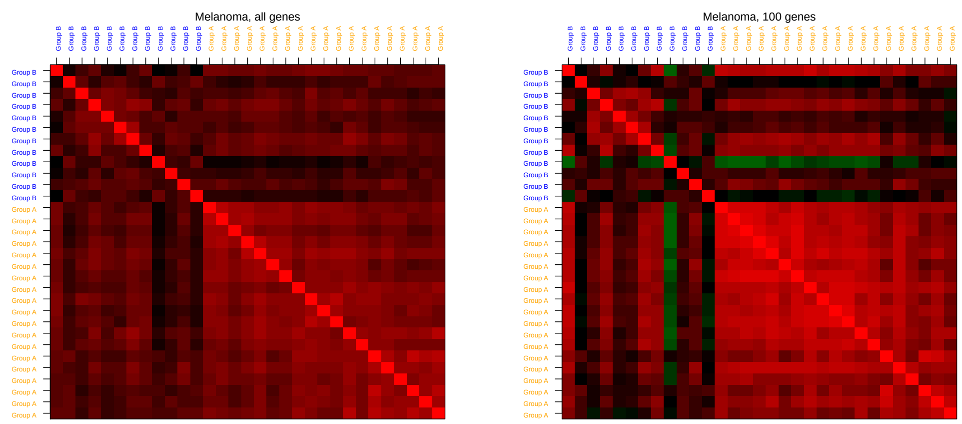
(a) (b)

Figure 10: *Correlation matrix, leukemia dataset* – Images of the correlation matrix for the 72 ALL B-cell, ALL T-cell, and AML samples based on expression profiles for all $p = 3,571$ genes (panel (a)) and for the $p = 100$ genes with the largest variance (panel (b)). The mRNA samples are ordered by class, first ALL B-cell (blue), then ALL T-cell (orange), and finally AML (magenta).



(a) (b)

Figure 11: *Correlation matrix, NCI 60 dataset* – Images of the correlation matrix for the 61 cell line mRNA samples based on expression profiles for all $p = 5,244$ genes (panel (a)) and for the $p = 200$ genes with the largest variance (panel (b)). The mRNA samples are ordered by class: 7+2 breast, 6 CNS, 7 colon, 6+2 leukemia, 8 melanoma, 9 NSCLC, 6 ovarian, 8 renal.



(a) (b)

Figure 12: *Correlation matrix, melanoma dataset* – Images of the correlation matrix for the 31 melanoma mRNA samples based on expression profiles for all $p = 3,613$ genes (panel (a)) and for the $p = 100$ genes with the largest variance (panel (b)). The mRNA samples are ordered by class, as proposed in Bittner *et al.* [3], first Group B (blue), then Group A (orange).

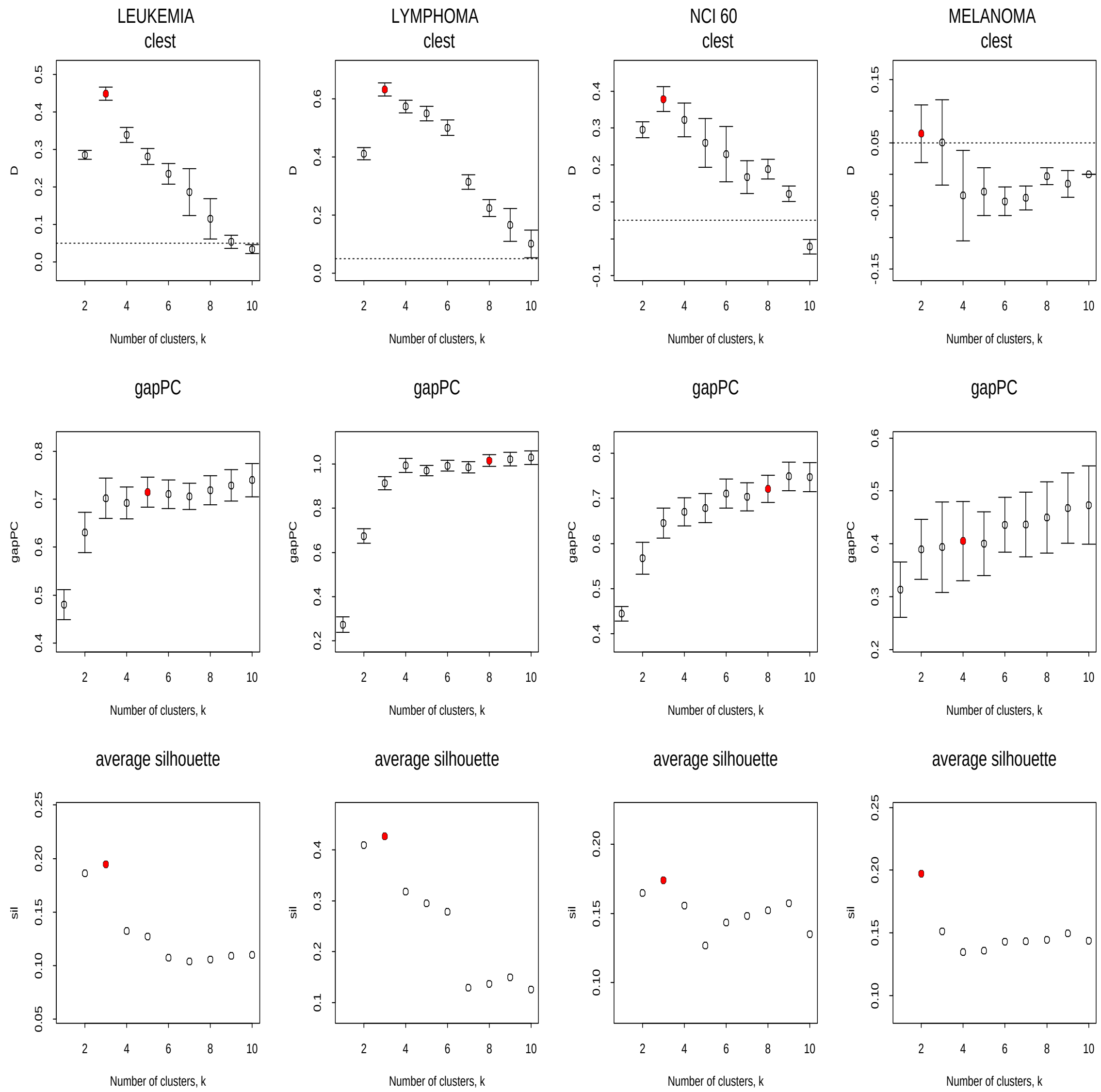


Figure 13: *Estimating the number of clusters, results for microarray data* – Plots of d_k, $gapPC_k$, and $\bar{sil}_k$ *vs.* k, with error bars based on the standard deviations for the first 2 statistics computed as in Section 5.2. The horizontal lines for the d_k plots correspond to a d_{min} cut-off of 0.05. The estimates for the number of clusters are indicated by filled plotting symbols.

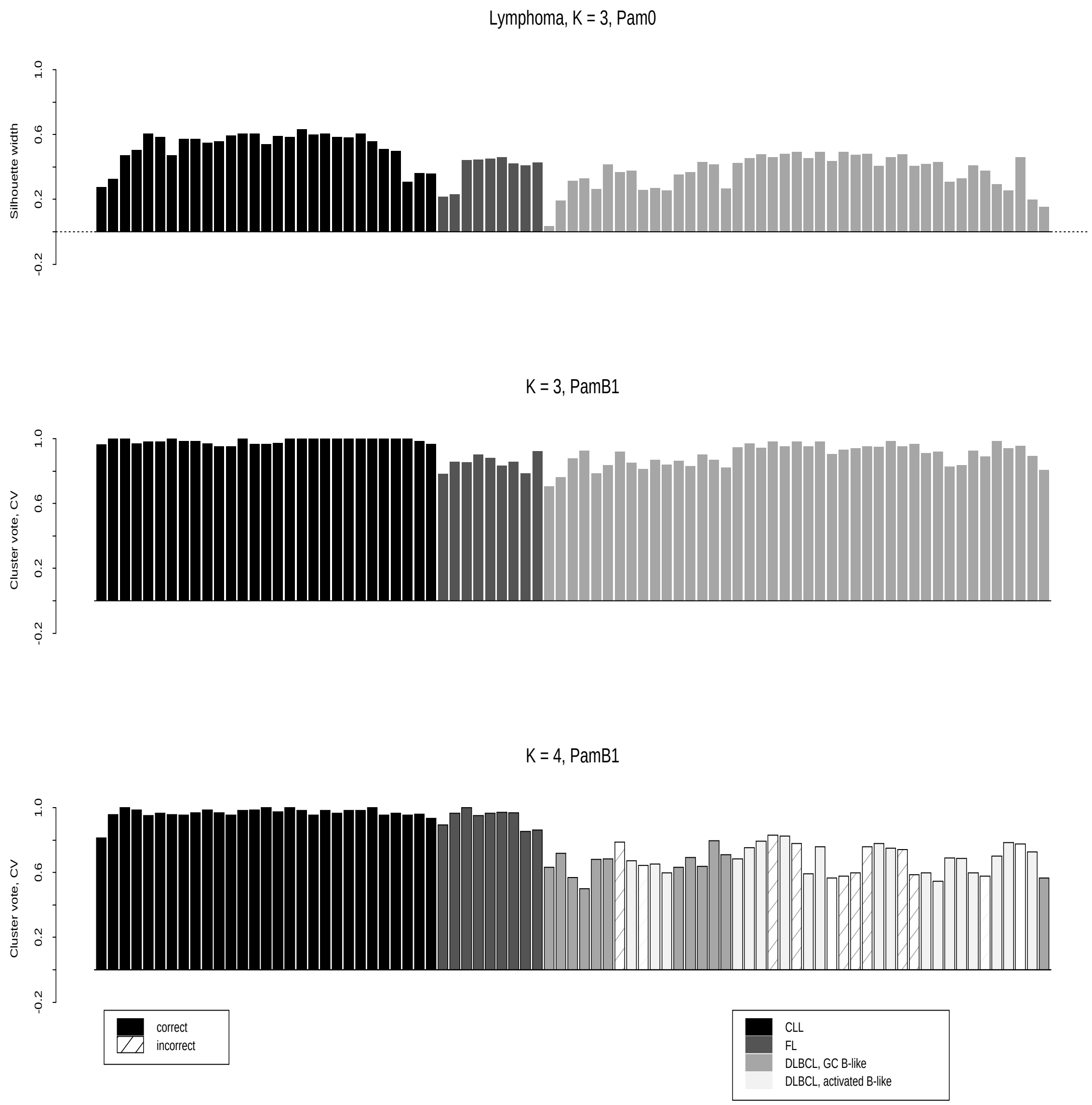


Figure 14: *Cluster votes and silhouette widths, lymphoma dataset,* $K = 3, 4$ *clusters* – Silhouette plot for a single application of *PAM* (Pam0) with $K = 3$ and plot of cluster votes for *Bag1* (PamB1) with $K = 3$ and 4. The "known" classes of the observations are represented by different shades of gray and incorrect cluster assignments are indicated by hatching. For $K = 4$, the DLBCL subclasses reported in Alizadeh *et al.* [1] are taken as the "true" classes. Observations are ordered as in Figure 9.

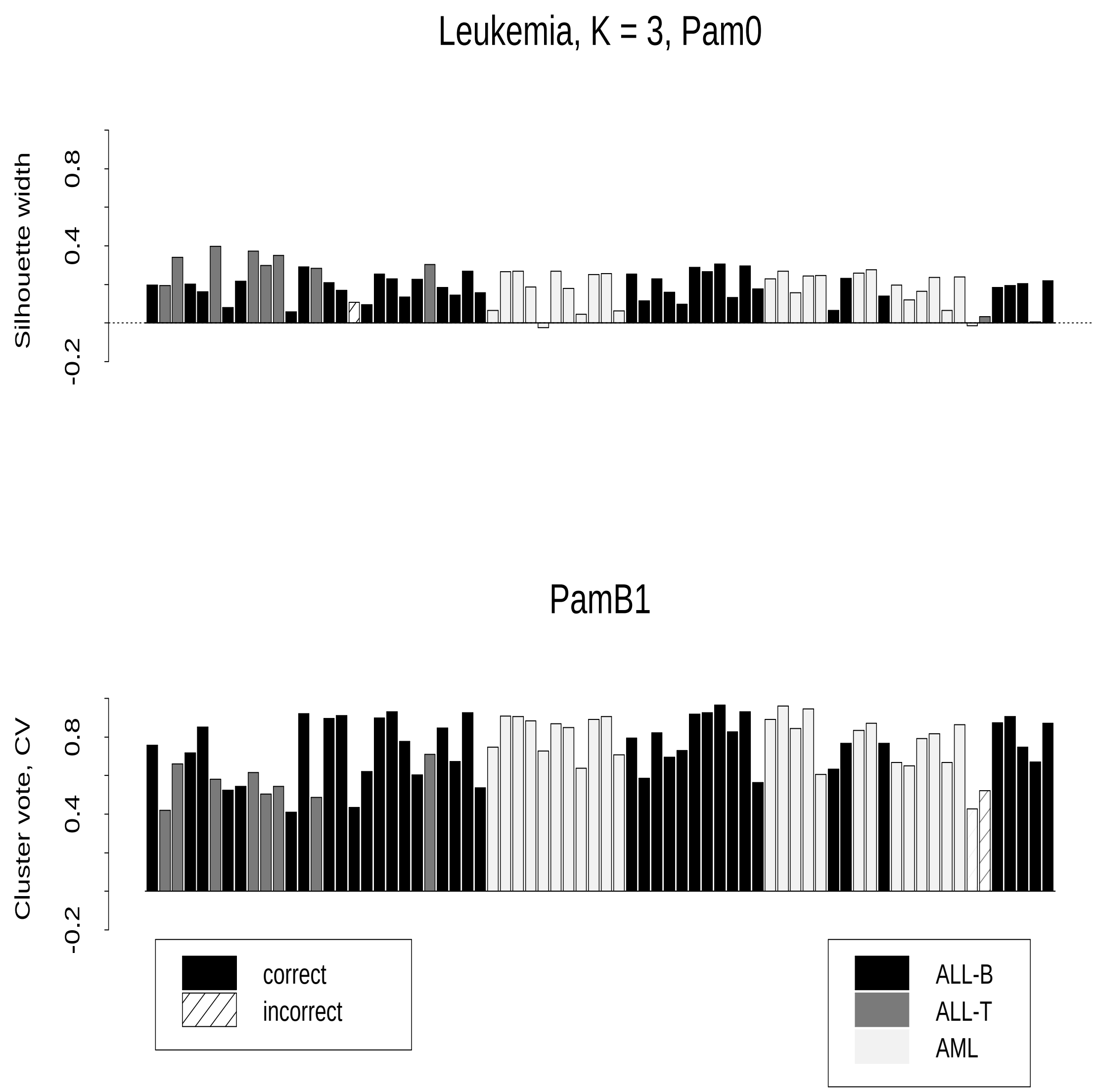


Figure 15: *Cluster votes and silhouette widths, leukemia dataset,* $K = 3$ – Silhouette plot for a single application of *PAM* (Pam0) and plot of cluster votes for *Bag1* (PamB1). Observations are ordered as in Golub *et al.* and not as in Figure 10.

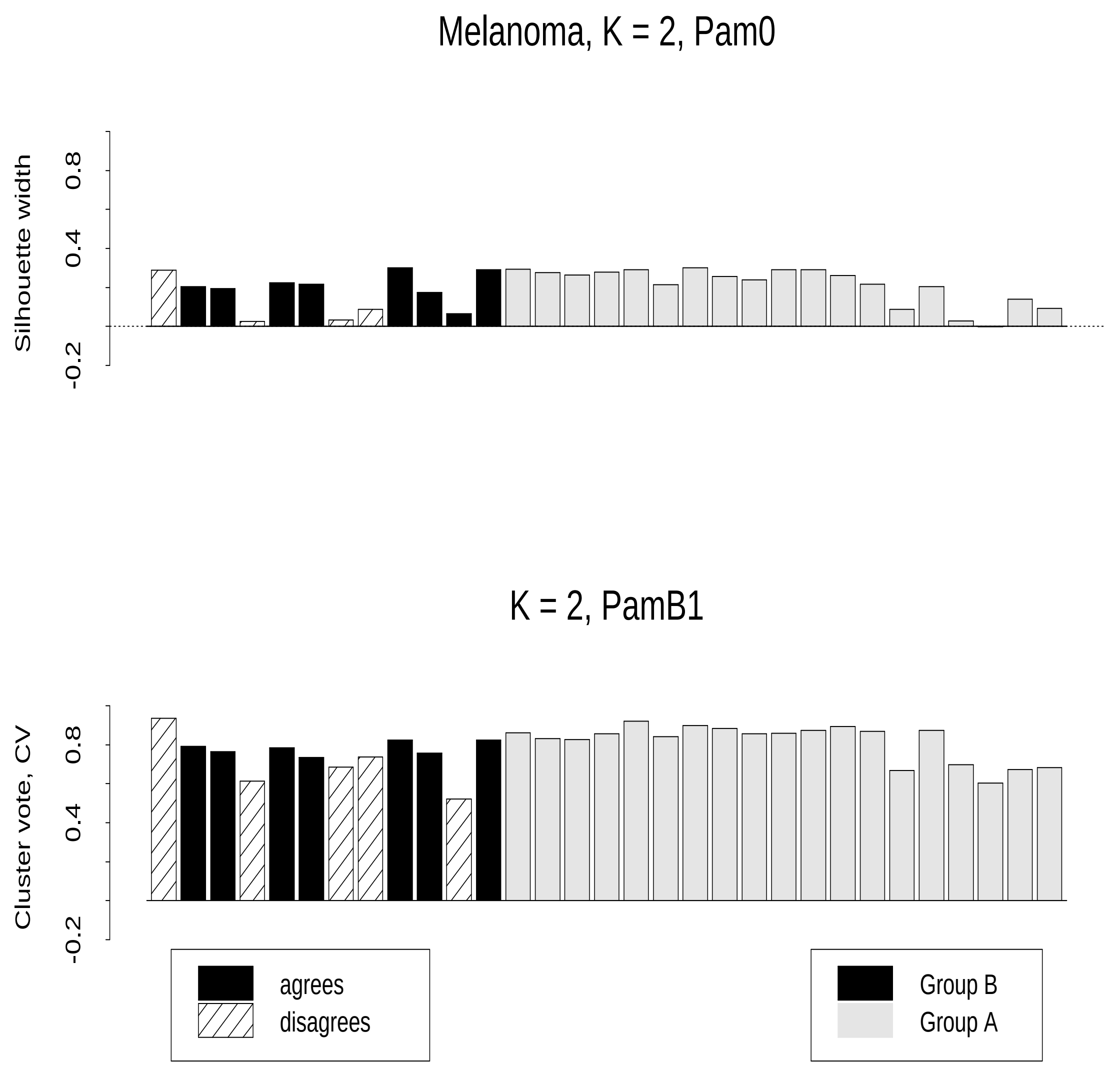


Figure 16: *Cluster votes and silhouette widths, melanoma dataset, $K = 2$* – Silhouette plot for a single application of *PAM* (Pam0) and plot of cluster votes for *Bag1* (PamB1). Observations are ordered as in Figure 12.