Consensus Clustering

A resampling-based method for class discovery and visualization of gene expression microarray data

Stefano Monti*, Pablo Tamayo, Jill Mesirov and Todd Golub
{smonti,tamayo,mesirov,golub}@broad.mit.edu
Broad Institute/MIT
Center for Genome Research
320 Charles St., Cambridge, MA – 02141

Abstract. In this paper we present a new methodology of class discovery and clustering validation tailored to the task of analyzing gene expression data. The method can best be thought of as an analysis approach, to guide and assist in the use of any of a wide range of available clustering algorithms. We call the new methodology *consensus clustering*, and in conjunction with resampling techniques, it provides for a method to represent the consensus across multiple runs of a clustering algorithm and to assess the stability of the discovered clusters. The method can also be used to represent the consensus over multiple runs of a clustering algorithm with random restart (such as K-means, model-based Bayesian clustering, SOM, etc.), so as to account for its sensitivity to the initial conditions. Finally, it provides for a visualization tool to inspect cluster number, membership, and boundaries. We present the results of our experiments on both simulated data and real gene expression data aimed at evaluating the effectiveness of the methodology in discovering biologically meaningful clusters.

1. Introduction

The problem of discovering new taxonomies (classifications of objects according to some natural relationships) from data has received considerable attention in the statistics and machine learning community. In this paper, we are concerned with a particular type of taxonomy discovery, namely, cluster analysis, the discovery of distinct and non-overlapping sub-populations within a larger population, the member items of each sub-population sharing some common features or properties deemed relevant in the problem domain of study [17]. This type of unsupervised analysis is of particular significance in the emerging field of functional genomics and gene expression data analysis, where the need for the molecular-based refinement of broadly defined biological classes is an active field of study, with potentially high payoffs in cancer diagnosis, prognosis, and treatment, among others.

Fundamental issues to be addressed when clustering data include: i) how to determine the number of clusters; and ii) how to assign

[*] To whom correspondence should be addressed.

confidence to the selected number of clusters, as well as to the induced cluster assigments. The latter issue is particularly important in gene expression data analysis, where the problem of a relatively small sample size is compounded by the very high dimensionality of the data available, making the clustering results especially sensitive to noise and susceptible to over-fitting.

Recent proposals exist for the use of resampling and cross validation techniques to simulate perturbations of the original data set, so as to assess the stability of the clustering results with respect to sampling variability [3, 4, 10, 18, 22, 31]. In particular, in [4] the use of bootstrapping to assess clustering stability and to validate the results output by hierarchical clustering was introduced. In this paper we build upon some of those ideas, and develop a general, model-independent resampling-based methodology of class discovery and clustering validation and visualization tailored to the task of analyzing gene expression data. One of the important features of the proposed methodology is that all of the information provided by the analysis of the resampled data can be graphically visualized, and incorporated in the decisions about clusters' number and cluster membership. As we will show, the inspection of the visualized data can often help gain additional insight into the recommendations returned by the algorithm.

We call the new methodology *consensus clustering*, as it provides for a method to represent the consensus across multiple runs of a clustering algorithm, to determine the number of clusters in the data, and to assess the stability of the discovered clusters. The method can also be used to represent the consensus over multiple runs of a clustering algorithm with random restart (such as K-means, model-based Bayesian clustering, SOM, etc.), so as to account for its sensitivity to the initial conditions. Finally, it provides for a visualization tool to inspect cluster number, membership, and boundaries.

The remainder of this manuscript is organized as follows. In Section 2, we briefly discuss some related work and some outstanding issues in cluster analysis. Section 3 describes the proposed methodology in detail. In particular, we formally define consensus and methods for its measurement and visualization. In Section 4, we present the results of our experiments on both simulated data and real gene expression data aimed at evaluating the effectiveness of the methodology in discovering biologically meaningful clusters. We conclude the manuscript with a discussion of the results and of possible directions for further research.

2. Background

In this section, we briefly survey some of the more commonly used clustering algorithms. This is not meant to be a comprehensive review. Rather, it is aimed at emphasizing some of the outstanding issues in cluster analysis. The interested reader can find a more comprehensive survey of the topic in several recent books and papers (e.g., [10, 15, 17]).

In functional genomics, agglomerative hierarchical clustering (HC) has been widely adopted as the unsupervised analysis tool of choice, mainly because of its intuitive appeal and its visualization properties [12]. By not committing to a specific number of clusters, HC provides for a multi-resolution view of the data that can be extremely useful in exploratory data analysis. On the other hand, the method is exposed to the risk of incorporating the biases and preconceptions of the analyst because it does not provide for an "objective" criterion to establish the number of clusters and the clusters' boundaries. Furthermore, the resulting trees can lock in accidental features reflecting idiosyncrasies of the agglomeration rule. This is due to the deterministic nature of the agglomeration rule and the bottom-up direction of the agglomeration. An interesting recent proposal tries to circumvent some of HC's limitations, by providing for a principled model-based agglomeration rule that allows for the automatic determination of the number of clusters [27]. However, the proposed rule, which is based on a Bayesian score, was devised for the clustering of genes only.

Iterative descent clustering methods, such as the self-organizing map (SOM) [20, 21, 30] and K-means clustering [15], circumvent some of the shortcomings of HC by providing for univocally defined clusters and cluster boundaries. However, they lack the intuitive and visual appeal of HC, and the number of clusters must be chosen a priori. Methods of model-based probabilistic clustering [1, 6, 33, 38] automatically select the number of clusters. These methods are often based on the Bayesian paradigm, thus allowing for the seamless combination of prior knowledge and observational data. A possible difficulty in their use is due to the distributional assumptions on which they are based. In particular, most of these methods are based on asymptotic approximations of the marginal likelihood, whose accuracy tends to decrease as the sample size decreases [7, 19]. This can clearly be a problem in the "large N, small p" paradigm (i.e., high dimension and small sample size) typical of gene expression data [35, 36]. An additional shortcoming common to all the iterative, greedy search-based clustering methods is their sensitivity to the search's starting point. This makes the clustering results harder to trust, since it may be difficult to reconcile inconsistent results returned by multiple runs of the algorithm.

In addition to model-based clustering, several other strategies have been proposed for automatically selecting the number of clusters [10, 23, 32, 39]. Most of these strategies attempt to minimize some measure of cluster *compactness*, that is, of the within-cluster *vs* between-cluster variability. Since this measure is bound to decrease even in the absence of a multi-cluster signature, some form of penalty for "model complexity" is required. Some of these penalties represent analytically derived measures of the minimum improvement in cluster compactness that can be expected as the number of clusters is increased under a null hypothesis encoding for the absence of clusters [14, 39]. In most cases, an appropriate encoding of the null hypothesis does not allow for the analytical derivation of the corresponding penalty. An empirical estimate of the penalty can then be computed based on some form of permutation test, so as to try to capture as much as possible of the idiosyncrasies of the data distribution of interest [10, 32].

An extremely important issue in cluster analysis is the validation of the clustering results, that is, how to gain confidence about the significance of the putative clusters, both in terms of cluster numbers and in terms of cluster assignments. Lacking an external objective criterion — the equivalent of a known class label in supervised learning — this validation becomes somewhat elusive. While some principled statistical procedures exist for testing the significance of a clustering result [5], these have only been derived for low-dimensional data, and it is not clear how well they apply to high-dimensional gene expression data. Similarly, Bayesian approaches to model-based clustering provide for measures such as the Bayes factor [19] to assign confidence to the attained results. However, as already pointed out, the computation of the Bayes factor is often based on asymptotic approximations whose accuracy deteriorates as the sample size decreases [7, 19]. This clearly does not detract from these methods, since virtually every clustering method is based — whether implicitly or explicitly — on several assumptions about the data-generating process. However, it points to the fact that the last word on their usefulness must come from an empirical evaluation of their performance when applied to gene expression data, such as in [38].

An alternative approach to cluster validation is based on resampling [3, 4, 10, 18, 22, 31]. Methods adopting this approach use different resampling schemes to simulate perturbations of the original data set, so as to assess the stability of the clustering results with respect to sampling variability. The underlying assumption is that the more stable the results are with respect to the simulated perturbations, the more these results are to be trusted. One of the appeals of some of these methods is that, by using the data itself to simulate the perturbations, the resulting perturbed data can incorporate the relevant dependencies among the observed features. Clearly, whether or not these dependencies are then modeled depends on the clustering algorithm used. A possible drawback of these methods is in the fact that by not explicitly modeling the assumptions underlying the data-generating process, it is sometime difficult to rigorously evaluate the significance of the results produced. The empirical evaluation of these methods on real data thus becomes all the more important.

The clustering methodology we propose in this paper falls squarely in the category of resampling-based methods. The extensive experimental evaluation reported in Section 4 tries to address the latter concern.

3. Methodology

The main motivation for the proposed methodology is the need to assess the "stability" of the discovered clusters, that is, the robustness of the putative clusters to sampling variability. The basic assumption of this method is intuitively simple: if the data represent a sample of items drawn from distinct sub-populations, and if we were to observe a different sample drawn from the same sub-populations, the induced cluster composition and number should not be radically different. Therefore, the more the attained clusters are robust to sampling variability, the more we can be confident that these clusters represent real structure.

To this end, perturbations of the original data can be simulated by resampling techniques. The clustering algorithm of choice can then be applied to each of the perturbed data sets, and the agreement, or *consensus*, among the multiple runs can be assessed. Consensus clustering simply formalizes this procedure, and it is summarized in pseudo-code format in Figure 1. In the remainder of this section we illustrate in detail each of the procedure's steps.

3.1. Notation

Given a data set of interest $D = \{e_1, e_2, \ldots, e_N\}$, the goal of clustering is to partition the observed data into a set of exhaustive and non-overlapping clusters. Formally, a K-cluster partition $\boldsymbol{P}$ of D can be defined as $\boldsymbol{P} \equiv \{P_1, P_2, \ldots, P_K\}$, such that $\bigcup_{k=1}^{K} P_k = D$, and $P_i \cap P_j = \emptyset$, $\forall_{i,j} : i \neq j$.

While the proposed methodology is quite general, and can be both applied to the clustering of genes and samples/experiments, we will mainly focus on the latter task. However, in order to emphasize the

```
Procedure Consensus Clustering

input: a set of items $D = \{e_1, e_2, \ldots, e_N\}$
       a clustering algorithm Cluster
       a resampling scheme Resample
       number of resampling iterations $H$
       set of cluster numbers to try, $\mathcal{K} = \{K_1, \ldots, K_{\max}\}$
for $K \in \mathcal{K}$ do
  $\boldsymbol{M} \leftarrow \emptyset$ {set of connectivity matrices, initially empty}
  for $h = 1, 2, \ldots, H$ do
    $D^{(h)} \leftarrow$ Resample($D$) {generate perturbed version of $D$}
    $M^{(h)} \leftarrow$ Cluster($D^{(h)}$,$K$) {cluster $D^{(h)}$ into $K$ clusters}
    $\boldsymbol{M} \leftarrow \boldsymbol{M} \cup M^{(h)}$
  end {for $h$}
  $\mathcal{M}^{(K)} \leftarrow$ compute consensus matrix from $\boldsymbol{M} = \{M^{(1)}, \ldots, M^{(H)}\}$
end {for $K$}
$\hat{K}$ $\leftarrow$ best $K \in \mathcal{K}$ based on consensus distribution of $\mathcal{M}^{(K)}$'s {§ 3.3.1}
$\boldsymbol{P} \leftarrow$ Partition $D$ into $\hat{K}$ clusters based on $\mathcal{M}^{(\hat{K})}$
return $\boldsymbol{P}$ and $\{\mathcal{M}^{(K)} : K \in \mathcal{K}\}$
```

Figure 1. High level pseudo-code for the consensus clustering procedure.

generality of the method, we will mostly refer to the elements being clustered (the $e_i \in D$) as items, and the "coordinates" of the data

Table I. Summary of the notation used.

Symbol	**Description**
$D = \{e_1, \ldots, e_N\}$	Generic dataset (e_i's are the items to be clustered)
N	Number of items in a dataset
$\boldsymbol{P} = \{P_1, \ldots, P_K\}$	Partition of D into K clusters
$K, K_{\max}$	Number of clusters, max number of clusters
N_k	Number of items in cluster k
H	Number of resampling iterations
$D^{(h)}$	Dataset obtained by resampling D (h-th iteration)
M, $M^{(h)}$	Connectivity matrix, corresponding to h-th iteration
$\mathcal{M}$, $\mathcal{M}^{(K)}$	Consensus matrix, corresponding to K clusters

space as features. When clustering genes, these will be the items, and the features will be the observations of a gene across many experiments. When clustering experiments, these will be the items, and the features are the genes whose expression is measured for each experiment. Table I presents a summary of the symbols used throughout the paper.

3.2. Measuring consensus

Assuming a resampling scheme and a clustering algorithm have been selected, we need to devise a method for representing and quantifying the agreement among the clustering runs over the perturbed datasets. To this end, we define a *consensus matrix.* A consensus matrix is an $(N \times N)$ matrix that stores, for each pair of items, the proportion of clustering runs in which two items are clustered together. The consensus matrix is obtained by taking the average over the *connectivity matrices* of every perturbed dataset. More specifically, let $D^{(1)}, D^{(2)}, \ldots, D^{(H)}$ be the list of H perturbed datasets obtained by resampling the original dataset D. Also, let $M^{(h)}$ denote the $(N \times N)$ connectivity matrix corresponding to dataset $D^{(h)}$ (more precisely, corresponding to the result of applying the clustering algorithm of choice to dataset $D^{(h)}$). The entries of this matrix are defined as follows:

$$M^{(h)}(i,j) = \begin{cases} 1 & \text{if items } i \text{ and } j \text{ belong to the same cluster}\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{1}$$

Finally, let $I^{(h)}$ be the $(N \times N)$ indicator matrix such that its (i,j)-th entry is equal to 1 if both items i and j are present in the dataset $D^{(h)}$, and 0 otherwise. The need for the indicator matrix is due to the use of resampling. Most resampling schemes — such as bootstrapping, or subsampling — yield datasets that do not include all items from the original dataset. We thus need to keep track of the number of iterations in which two items are both included in the resampled dataset.

The consensus matrix $\mathcal{M}$ can then be defined as a properly normalized sum of the connectivity matrices of all the perturbed datasets $\{D^{(h)} : h = 1, 2, \ldots, H\}$:

$$\mathcal{M}(i,j) = \frac{\sum_h M^{(h)}(i,j)}{\sum_h I^{(h)}(i,j)}\,. \tag{2}$$

That is, the entry (i,j) in the consensus matrix records the number of times items i and j are assigned to the same cluster divided by the total number of times both items are selected. It should be clear that the consensus matrix is symmetric, in that $\mathcal{M}(i,j) = \mathcal{M}(j,i)$, for all i and j. We will refer to the entry (i,j) in the consensus matrix as the *consensus index* for the corresponding item pair (irrespective of the items' order).

As defined, each entry in $\mathcal{M}$ is a real number between 0 and 1, and perfect consensus corresponds to a consensus matrix $\mathcal{M}$ with all the entries equal to either 0 or 1. Furthermore, if the items in the matrix were arranged so that items belonging to the same cluster are adjacent to each other, perfect consensus would translate into a block-diagonal matrix with non-overlapping blocks of 1's along the diagonal — each block corresponding to a different cluster — surrounded by 0's.

Another important property of the consensus matrix is that it provides for a similarity measure that can be used in conjunction with an agglomerative hierarchical tree construction algorithm to yield a dendogram of item adjacencies. That is, $\mathbf{1} - \mathcal{M}$ defines a new distance matrix that can be used in place of the usual measures, such as Euclidean distance, Pearson correlation, or Kullback-Leibler divergence, among others. This is a point to which we will return.

3.2.1. *Consensus matrix reordering and visualization*

The consensus matrix lends itself naturally to be used as a visualization tool to help assess the clusters' composition and number. In particular, if we associate a color gradient to the 0-1 range of real numbers, so that white corresponds to 0, and dark red corresponds to 1, and if we assume the matrix is arranged so that items belonging to the same cluster are adjacent to each other (with the same item order used to index both the rows and the columns of the matrix), a matrix corresponding to perfect consensus will be displayed as a color-coded *heat map* characterized by red blocks along the diagonal, on a white background. Figure 2 shows the color-coded heat maps obtained by applying consensus clustering to two simulated datasets, `Uniform1` and `Gaussian3`. Dataset `Uniform1` is generated from a uniform 600-dimensional hypercube. Dataset `Gaussian3` represents the union of three Gaussian distributions in a 600-dimensional space. The heat maps shown in Figure 2 represent the consensus over 500 iterations for $K = 3$ (details about the data and the consensus clustering settings needed to produce the matrices of Figure 2 are given in Section 4). It is evident from Figure 2 that the heat map corresponding to `Gaussian3` displays a well defined 3-block structure, while the heat map corresponding to `Uniform1` shows no such structure.

In the example of Figure 2, since the datasets were artificially created, the items are already sorted by their known cluster membership (so that items belonging to the same cluster are displayed next to each other) thus yielding the block-diagonal structure shown. In general, however, the cluster membership is not known in advance, and an

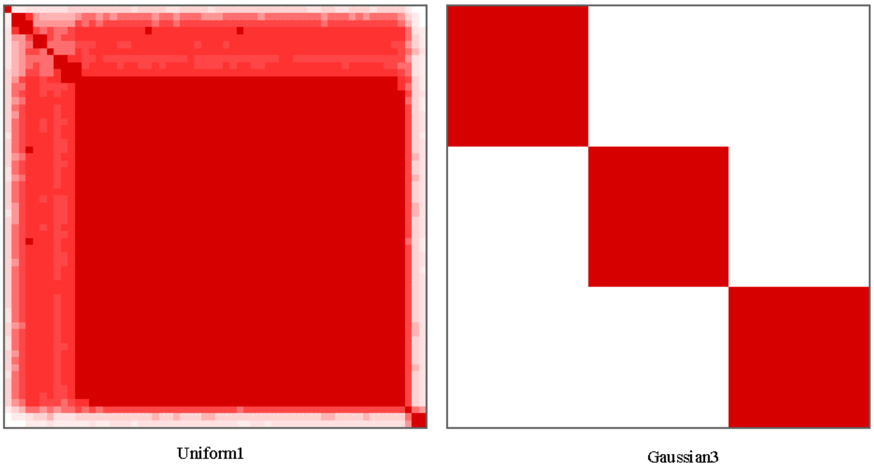

Figure 2. Color-coded heat maps corresponding to the consensus matrices $\mathcal{M}^{(3)}$ for `Uniform1` and `Gaussian3`.

item order needs to be chosen. We can use the consensus matrix itself to determine the optimal item order. In particular, if we carry out hierarchical clustering with the consensus matrix as similarity matrix, the induced dendogram will have its leaves arranged so as to have items with highest consensus index adjacent to each other, thus maximizing the block-diagonal nature of the heat map ordered accordingly (it is important to emphasize again that the same item order is used to index both the rows and the columns of the matrix). For the leaf-ordering task, we use the optimal leaf-ordering algorithm described in [2].

As we will show, the visualization of a consensus matrix provides for a very powerful tool to assess the stability of the putative clusters, as well as their optimal number.

3.2.2. *Consensus' summary statistics*

Based on the consensus matrix, we can define summary statistics accounting for the stability of a given cluster as well as of a cluster's members. These statistics can be used to establish a ranking of the clusters in terms of their stability, as well as to identify the more representative items within each clusters. For each cluster $k \in K$, we define the *cluster consensus* $m(k)$ and, for each item $e_i \in D$ and each cluster k, we define the *item consensus* $m_k(i)$.

Let us first define I_k as the set of indices of items belonging to cluster k, that is, $I_k = \{j : e_j \in k\}$. We can then define a cluster's consensus as follows:

$$m(k) = \frac{1}{N_k(N_k-1)/2} \sum_{\substack{i,j \in I_k \\ i<j}} \mathcal{M}(i,j)\,, \tag{3}$$

that is, as the average consensus index between all pairs of items belonging to the same cluster. Furthermore, for each item e_i, and each cluster k, the corresponding item consensus can be defined as

$$m_i(k) = \frac{1}{N_k - 1\{e_i \in I_k\}} \sum_{\substack{j \in I_k \\ j \neq i}} \mathcal{M}(i,j)\,, \tag{4}$$

where $1\{\texttt{cond}\}$ is the indicator function that is equal to 1 when `cond` is true, and 0 otherwise. The item consensus $m_i(k)$ measures the average consensus index between item e_i and all the (other) items in cluster k. For example, in case of perfect consensus (i.e., of a consensus matrix containing 0's and 1's only), the cluster consensus $m(k)$ would be 1 for all k's. Similarly, for a given k the item consensus $m_i(k)$ would be 1 for all items $e_i \in k$, and 0 for the others.

These measures can be used to quantify the stability of each cluster, and to rank items within clusters in terms of how representative of a given cluster they are (e.g., when clustering experiments, how the expression pattern of the observed genes for a given experiment is prototypical of the experiments within that cluster).

3.3. Determining the number of clusters

The properties of the consensus matrix illustrated in the previous section also suggest a method for finding the number of clusters that best fits the data. In particular, given that perfect consensus translates into a consensus matrix with all the entries set to either 1 or 0, we should interpret deviation from this optimal scenario as an indication of lack of stability of the putative clusters. At the most general level, the idea is to construct a consensus matrix $\mathcal{M}^{(K)}$ for each of a series of cluster numbers ($K = 2, 3, \ldots, K_{\max}$), to compare the resulting consensus matrices, and to select the cluster number corresponding to the "cleanest" matrix (i.e., a matrix containing 0's and 1's only). In this section, we introduce and discuss a measure of consensus based on the matrix $\mathcal{M}$. We refer to this measure as *consensus distribution*, as it is based on an assessment of how the entries of the consensus matrix are distributed within the 0-1 range. The extent to which this distribution is skewed toward 0 and 1 is taken as an indication of good clustering. The measure we propose is closely related to the concept of *concentration* of a distribution [8]. As we will show in the experimental evaluation of Section 4, this measure works remarkably well when tested on both simulated and gene-expression datasets.

3.3.1. *Consensus distribution*

If we were to plot a histogram of a consensus matrix entries (i.e., a histogram of the $N(N-1)/2$ entries $\mathcal{M}(i,j)$'s for $i < j$), perfect consensus would translate into two bins centered at 0 and 1. The histogram corresponding to a noise cloud devoid of any signal would translate in the limit into a single bin centered at some fractional value between 0 and 1 (the exact fractional value would depend on the number K of clusters requested, and it would decrease as K increases, reducing to 0 when K equals the number of items in the dataset). This is a consequence of the fact that any two items would have an equal probability of being clustered together.

With these considerations in mind, we return to the two simulated datasets, `Uniform1` and `Gaussian3`, introduced in the previous section, and whose consensus matrices are shown Figure 2. Figure 3.a shows the corresponding histograms of consensus indices. Notice the bimodal nature of the histogram for `Gaussian3` (with modes around 0 and 1), and the largely unimodal nature of the histogram for `Uniform1`. Few words of explanation regarding the histogram for `Uniform1` are in order. The nature of hierarchical clustering is such that, even in the absence of a multi-cluster signature, the procedure will still establish a ranking of items according to their distance from one another. Furthermore, the relative distance between item pairs remains the same under the resampling scheme adopted (subsampling, see Section 3.4). Therefore, when no detectable multi-cluster structure is present in the data, the procedure will tend to group most of the items in a single cluster (corresponding to the large mode around 1 in the histogram), with few singleton items to account for the other clusters (corresponding to the trailing long tail toward 0 in the histogram).

For a given histogram, we can define and plot the corresponding *empirical cumulative distribution* (CDF) defined over the range $[0,1]$ as follows:

$$\mathrm{CDF}(c) = \frac{\sum_{i<j} 1\{\mathcal{M}(i,j) \leq c\}}{N(N-1)/2}\,, \tag{5}$$

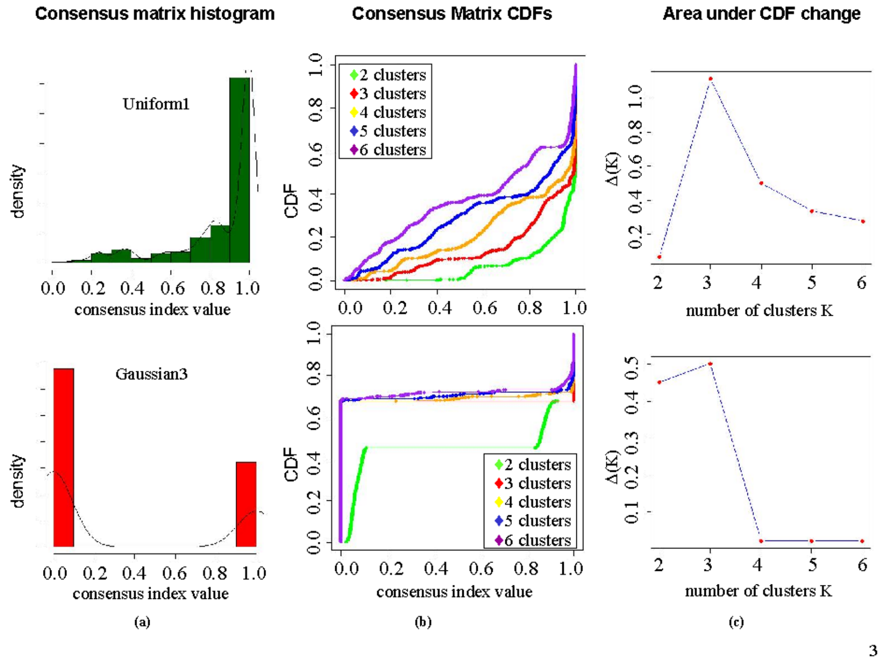

Figure 3. Measuring consensus for different K's on the simulated datasets of Figure 2: a) histograms of the entries of the consensus matrices $\mathcal{M}^{(3)}$; b) empirical CDFs corresponding to the entries of consensus matrices $\mathcal{M}^{(K)}$ for $K = 2, 3, \ldots, 6$; c) proportion increase $\Delta(K)$ in the area under the CDF.

where $1\{\ldots\}$ denotes the indicator function, $\mathcal{M}(i, j)$ denotes entry (i, j) of the consensus matrix $\mathcal{M}$, and N is the number of rows (and columns) of $\mathcal{M}$.

Figure 3.b shows plots of the CDFs for the histograms of Figure 3.a. The plots also include the CDFs corresponding to consensus matrices obtained for K's other than 3 (namely, $K = 2, 3, \ldots, 6$). For `Gaussian3`, it is clear how the predominance of 0's and 1's affects the shape of the corresponding CDFs, with a step around 0 (the magnitude of which is equivalent to the proportion of 0's in the matrix), a flat line reaching across the 0-1 range, and a second step around 1. Notice how for $K = 3$ and higher, the shape of the curve approaches the ideal step function, and how this shape hardly changes as we increase K past 3. The CDF for `Uniform1`, on the other hand, displays quite a different shape, with a gradual climb of values between 0 and 1, reflecting the lack of stability in cluster membership. The difference between the two CDFs can be partially summarized by measuring the area under the two curves. The area under the CDF corresponding to $\mathcal{M}^{(K)}$ is computed based on the following formula:

$$A(K) = \sum_{i=2}^{m} [x_i - x_{i-1}] \mathrm{CDF}(x_i) , \tag{6}$$

where the set $\{x_1, x_2, \ldots, x_m\}$ is the sorted set of entries of the consensus matrix $\mathcal{M}^{(K)}$ (with $m = N(N-1)/2$).

Notice that the area for `Gaussian3` is considerably greater than the area for `Uniform1`. However, the important comparison is not between CDFs corresponding to different datasets, but between CDFs for different K's computed over the same dataset. The plot of the CDFs for `Gaussian3`, shown in Figure 3.b, illustrates the typical bimodal shape and the progression of curves to be observed when the analyzed dataset contains actual clusters. As K is increased, the area under the CDF markedly increases as long as K is less than or equal to K_{true} (as a consequence of the increase in the number of 0's in the consensus matrix). However, when K_{true} is reached, any further increase in the number of clusters does not lead to a corresponding marked increase in the CDF area. This is because as we start introducing spurious clusters, these are inherently unstable, thus leading to an increase in the number of fractional entries in the consensus matrix (hence, away from 0). Conversely, for `Uniform1` the bimodal shape of the CDFs is clearly absent, and the area under the CDF keeps increasing in a stable manner.

This behavior can be summarized by plotting the proportion increase in the CDF area as K increases, computed as follows:

$$\Delta(K) = \begin{cases} A(K) & \text{if } K = 2 \\ \frac{A(K+1) - A(K)}{A(K)} & \text{if } K > 2 , \end{cases} \tag{7}$$

where $A(K)$ denotes the area under the CDF for the consensus matrix corresponding to K clusters. The special treatment of $K = 2$ is due to the fact that $A(1) = 0$ (the consensus matrix for $K = 1$ contains only 1's), thus not allowing for the standard computation of relative increase in the area for $K = 2$. This "impropriety" of $\Delta(K)$ also points to the fact that it cannot be sensibly used to choose between 1 and 2 clusters, and that inspection of the corresponding CDF curves will be needed for this purpose.

Figure 3.c plots the value of $\Delta(K)$ for different numbers of clusters K's. As shown, for `Gaussian3` the $\Delta(K)$ is significantly larger than 0 only up to $K = 3$, at which point it levels off (with values close to 0%). Conversely, for `Uniform1`, as K increases, the corresponding $\Delta(K)$'s remain large (well above a 30% increase, with a maximum of >100% at $K = 3$).

It should be pointed out that while the area under the CDF is guaranteed to increase when consensus clustering uses hierarchical clustering in the inner loop, this is not true in general. When using hierarchical clustering, an increase from K to $K+1$ clusters is obtained by splitting one of the available K clusters. The clusters' boundaries are thus in agreement, and the consensus matrix $\mathcal{M}^{(K+1)}$ is guaranteed to have a number of 0's greater than or equal to the number of 0's in $\mathcal{M}^{(K)}$. Consequently, the area under the CDF is bound not to decrease as K increases. This is not true of other clustering algorithms. For example, if we use SOM in the inner-loop, when searching for $K+1$ clusters, these do not necessarily follow the boundaries of the clusters obtained when searching for K clusters. Consequently, the number of 0's in $\mathcal{M}^{(K)}$ is not a monotonic function of K. In this case, we redefine $\Delta(K)$ to measure the relative area increase with respect to the largest area observed for any $K' < K$. That is, we replace $A(K)$ with $\hat{A}(K)$ in Equation (7), where $\hat{A}(K) = \max_{K' \in \{2,\ldots,K\}} A(K')$. In so doing, we assess the relative improvement with respect to the best result (the best K) obtained thus far.

To summarize, the CDFs and the corresponding $\Delta(K)$'s try to quantify the *concentration* [8] of the consensus distribution. The goal is to find the K that maximizes this concentration. The selection of the appropriate number of clusters proceeds by inspection of the CDFs' shape and progression as K increases. The inspection of a CDF shape is to assess its bimodality, which suggests the presence of clusters. The inspection of the CDF progression is to select the largest K that induces a large enough increase in the area under the corresponding CDF.

3.4. Resampling schemes

In this section, we briefly discuss some of the possible resampling schemes that can be used within the consensus clustering procedure illustrated in Figure 1.

A well established resampling scheme is *bootstrapping* [11], whereby items are sampled with replacement from the original dataset. One of the desirable features of this resampling scheme is in its producing perturbed datasets that have the same size (i.e., the same number of items) as the original one. This is particularly relevant when trying to determine the number of clusters present in the data, an assessment that can be highly dependent on the sample size. However, bootstrapping produces datasets with identical replicate items, thus artificially inflating the compactness of the resulting dataset (e.g., if a given item is selected n times in a bootstrap dataset, the clustering algorithm might deem those n items – which are actually n replicates of the same item – worth a cluster by themselves). In light of this shortcoming, we will mainly focus on *subsampling* techniques, whereby a subset of items is sampled without replacement from the original dataset. While subsampling produces datasets smaller than the original one, our experimental evaluation shows that this does not have an adverse effect on the successful estimate of the "correct" number of clusters.

Given the very high dimensionality of the gene-expression data, when clustering samples it is possible to adopt *gene resampling* schemes, where the perturbations of the original dataset are obtained by selecting different subsets of genes, with or without replacement, at each iteration. That is, at each iteration clustering is performed based on the projection of the original dataset onto the subspace of selected genes.

Variations of this basic scheme can be obtained by associating non-uniform weights to the candidate genes, so that different genes will be sampled with different frequency. With this approach, we can incorporate prior information in the clustering process by assigning higher weight to genes considered more informative by some external criterion. For example, we might want to favor genes associated to biological pathways deemed particularly relevant to the phenotype of interest.

Feature selection for clustering purposes is a particularly difficult task, since a class label to guide this selection is not available. A welcome benefit of using gene resampling is that it allows us to evaluate how sensitive the clustering results are to the particular choice of genes included in the dataset used for analysis.

4. Experimental evaluation

We tested our clustering methodology on several simulated and real datasets. In this section, we first summarize the evaluation methodology and the evaluation metrics used. We then briefly describe the simulated and real datasets used for the evaluation. Finally, we report and discuss the results of the evaluation.

4.1. Evaluation methodology

The evaluation we carry out is aimed at assessing how good the proposed clustering method is at recovering known clusters from both simulated and gene-expression microarray data. To this end, we consider several datasets for which a multi-class distinction (a phenotype) is available. Each of this data sets constitutes the *gold standard* against which we evaluate the clustering results. We refer to the gold standard partition as *classes*, while we reserve the word *clusters* for the partition returned by the clustering algorithm.

4.1.1. *Evaluation metrics*

The cluster composition can be evaluated by measuring the agreement of the cluster partition with the known phenotype. If the number of clusters were always correct (i.e., equal to the known number of classes), we could use the classification error rate as the measure of agreement. That is, once we establish a mapping between cluster labels and known class labels, we can interpret cluster assignment as a conventional classification task, and measure its error rate. However, the error rate is harder to interpret when the number of clusters is incorrect. To this end, we use the adjusted Rand index [16], a measure of agreement between alternative data partitions that can be used even when considering partitions with different numbers of clusters [24]. The adjusted Rand index ranges between 0 and 1, with 1 corresponding to perfect agreement, and 0 corresponding to the expected value of the index for two random partitions under the assumption of a hypergeometric distribution for the model of randomness (see Appendix A for details about its computation).

4.1.2. *Experimental design*

The details of the experimental design are as follows:

- We apply consensus clustering to the dataset of choice. The output will include an estimate of the number of clusters K and, for a given K, a cluster assignment, both of which can be evaluated against the gold standard. When the estimated number of clusters K is different from the known number K_{true}, we report the value of the Rand index for both K and K_{true}. The latter provides additional information about how well the clustering procedure respects the known class boundaries.

- As a term of comparison, we also apply the Gap statistic proposed in [32]. Following [32], let W_K denote the within-cluster sum of squares, and let W^*_{Kb} denote the same quantity obtained after randomly permuting the genes/features within each item/experiment. Then the Gap statistic for K clusters, computed based on B permutation iterations, yields:

$$\text{Gap}(K) = \frac{1}{B}\left[\sum_{b=1}^{B}\log(W^*_{Kb})\right] - \log(W_K)\,. \tag{8}$$

The quantity $\mathrm{s}_K = \mathrm{sd}(K)\sqrt{1+1/B}$ denotes the corresponding standard deviation corrected for the simulation error. Given a set of cluster numbers $\mathcal{K} = \{1, 2, \ldots, K_{\max}\}$, the selection criterion proposed in [32] returns the first $K \in \mathcal{K}$ such that $\mathrm{Gap}(K) \geq \mathrm{Gap}(K+1) - \mathrm{s}_{K+1}$. This is also the selection criterion we will use.

– We compare the cluster assignment produced based on the application of HC to the consensus matrix, with the cluster assignment produced by application of HC to the raw data. This comparison is to evaluate whether we can obtain more accurate cluster assignments if we use the consensus matrix as a similarity measure in place of the usual Euclidean distance.

– Since different datasets are inherently more or less difficult to cluster, in order to evaluate the partition induced by the clustering algorithm, we compare it to the baseline Rand index produced by a naïve-Bayes (NB) classifier [9] trained on the same dataset by leave-one-out cross-validation. It is important to emphasize that the training of the NB is based on the supervised samples (i.e., the samples including the class labels), thus making the learning task considerably simpler. The NB's error rate gives an indication of the clustering accuracy we may hope to achieve under much more favorable conditions.

As noted, we have included in the evaluation the results of the application of the Gap statistics for the estimation of the number of clusters. We use the Gap statistic because, similar to our method, it is model-independent, thus allowing for its adoption in conjunction with different clustering algorithms. However, we want to emphasize that the comparison of our method with another model-independent method is somewhat beside the point. In fact, our proposed methodology is meant to go beyond the estimation of the number of clusters, by also providing for a resampling-based method of cluster assignment and visualization. In other words, the goal of our evaluation, is not to carry out an exhaustive comparison of methods for estimating the number of clusters, nor to prove the superiority of our method in this regard. The main purpose of the inclusion of the Gap statistic is as a term of reference, so as to be able to relate our results to those obtained based on a well accepted metric that has been shown to perform reasonably well in several studies [3, 10, 32].

4.1.3. *Consensus clustering settings*

We apply consensus clustering to a given data set as outlined in Figure 1. Consensus clustering requires the specification of a clustering algorithm (the algorithm `Cluster` in Figure 1). For this purpose, we explore the use of two algorithms: i) hierarchical clustering with average linkage; and ii) the self organizing map (SOM). For each K we run $H = 500$ resampling iterations ($H = 200$ with SOM, since it takes much longer to run). At each iteration, the perturbed data set is obtained by sampling, without replacement, 80% of the items from the original data. The results produced by consensus clustering include: i) a set of ordered consensus matrices, one for each of the K's considered; ii) an estimate of the number of clusters; and iii) the corresponding cluster assignment. Cluster number and assignment can be evaluated against the gold standard.

To determine the number of clusters based on consensus clustering, we use the consensus distribution, and the proportion change in the area under the consensus CDF (see Section 3.3.1). The selected $\hat{K}$ will correspond to the number of clusters where the CDF levels off and the corresponding $\Delta(\hat{K})$ gets close to zero. Often, this selection criterion identifies a range of K's, rather than a single one. However, in combination with the inspection of the consensus matrix progression, it is usually possible to unambiguously select a single best $\hat{K}$. When ambiguities remain, these will be duly reported in the text and the corresponding tables.

In most cases we tried up to $K_{\max} = 9$ clusters. In some of the larger datasets (Novartis, St. Jude, and Normal tissues) we set $K_{\max} = 15$. Once the optimal number of clusters $\hat{K}$ is chosen, we establish the clusters' boundaries by using the corresponding consensus matrix $\mathcal{M}^{(\hat{K})}$ as a similarity measure to feed to a hierarchical clustering algorithm with average linkage, and by stopping the agglomeration procedure when $\hat{K}$ branches are left. The resulting subtrees determine the cluster members.

The data used were row- and column-normalized (so that both rows and columns sum to 0 and have a standard deviation of 1). This is necessary when using consensus clustering with HC, because it yields well-balanced hierarchical trees, which can in turn be split into non-trivial (i.e., non-singleton) clusters. Although this data-normalization is not necessary when using SOM, for comparative purposes we use the normalized data in all cases.

4.2. Datasets

In this section, we only describe the datasets used, which are listed in Table II, together with some of their relevant characteristics, such as number of classes, number of features/genes, and number of items/samples. The first six datasets represent simulated data, while the last six represent gene-expression microarray data.

4.2.1. *Simulated data*

Uniform1 and Gaussian1 are two datasets generated in order to evaluate the behavior of the clustering methodology when applied to data known not to contain distinct sub-populations. We considered both the uniform and the Gaussian distributions, as they represent rather different generating processes, and we were interested in examining how the corresponding consensus matrices and CDFs would look.

Gaussian3 is the 3-cluster, 60-sample dataset introduced in the example of Section 3, Figure 2. It is generated by having 200 distinct features out of the 600 assigned to each cluster. The data simulates a pattern whereby a distinct set of 200 genes is up-regulated in one of three clusters, and down-regulated in the remaining two clusters.

Gaussian4 and Gaussian 5 represent standard *mixtures of Gaussians* models. In particular, Gaussian4 represents the union of observations from four bivariate Gaussians, with the same diagonal covariance matrix $\mathbf{\Sigma} = 0.25\boldsymbol{I}$, and centered at the four corners of a square with side length $\lambda = 2$. A total of 200 samples, 50 per class, were generated. Similarly, Gaussian 5 represents the union of observations from 5 bivariate Gaussians, 4 of which are centered at the corners of the square of side length λ, with the 5th Gaussian centered at $(\lambda/2, \lambda/2)$. A total of 250 samples, 50 per class, were generated. We used two values of λ, namely, $\lambda = 2$ and $\lambda = 3$, to investigate different levels of overlapping between clusters.

Finally, we considered a dataset with unequal-size clusters and "gene" markers of different strength. Simulated6 consists of a 600-gene by 60-sample dataset. It can be partitioned into 6 classes with 8, 12, 10, 15, 5, and 10 samples respectively, each marked by 50 distinct genes uniquely up-regulated for that class. Additionally, 300 noise genes (i.e., genes having the same distribution within all clusters) are included. The genes for the different clusters are of varying "sharpness". That is, the 50 genes marking the first class are the sharpest — whith highest differential expression and lowest variation — followed by the 50 genes for the second cluster, etc. Figure 5.a depicts the expression profile of the 600 genes within each cluster. Simulated4 is simply a subset of Simulated6, obtained by removing the two sharpest clusters (clusters 1 and 2).

4.2.2. *Gene-expression microarray data*

The gene-expression datasets used are listed in Table II, and a very short description of their content is given in Table III. Further biological details about these data sets can be found in the referenced papers. Most data were processed on the Human Genome U95 Affymetrix©

Table II. Description of the simulated and real dataset used in the experimental evaluation.

Dataset	**# of classes**	**# of samples**	**# of features**	**Chip type**
Uniform1	1	60	600	
Gaussian1	1	60	600	
Gaussian3	3	60	600	
Gaussian4	4	400	2	
Gaussian5	5	500	2	
Simulated6	6	60	600	
Leukemia [13]	3	38	999	HU6800
Novartis multi-tissue [29]	4	103	1000	U95
St. Jude Leukemia [37]	6	248	985	U95
Lung cancer [4]	4+	197	1000	U95
CNS tumors [25]	5	48	1000	U95
Normal tissues [26]	13	99	1277	U95

microarrays. The leukemia dataset is from the previous-generation Human Genome HU6800 Affymetrix© microarray.

To make sure that the known phenotype for a given data set is the dominant signature in the data, we project the dataset on the space of gene markers for that phenotype. This is necessary, since we are using the given phenotype's information (i.e., its number of classes and its label assignments) as the gold standard against which to test the clustering method.[1] We use a simple signal-to-noise ratio (SNR) to rank genes [28], and the final gene pool is obtained by selecting the most up-regulated genes for each class, where the exact number depends on the dataset. In particular for a K-class dataset, we select the top n up-regulated genes for each class by considering the K one-vs-all binary

[1] This necessity is best explained by considering a simple, although admittedly extreme, example. Assume that the dataset of interest consists of breast, prostate, and lung cancer tissues for which long term survival information is available. Assume also that this tissue-type phenotype were unknown to us, and that we were to use, e.g., the 5-year survival as the phenotype against which to test our clustering algorithm. It is clear that the survival signature, if it exists at all, would be overwhelmed by, among others, the much stronger (but unknown) tissue-type signature. Therefore, using the 2-class survival distinction as our gold standard would be totally inappropriate since clearly it is not the dominant signature in the data.

Table III. Description of the class types included in the gene-expression data sets.

Dataset	Description
Leukemia [13]	Bone marrow samples obtained from acute leukemia patients at the time of diagnosis: 11 acute myeloid leukemia (AML) samples; 8 T-lineage acute lymphoblastic leukemia (ALL) samples; and 19 B-lineage ALL samples.
Novartis multi-tissue [29]	Tissue samples from four distinct cancer types: 26 breast, 26 prostate, 28 lung, and 23 colon samples.
St. Jude Leukemia [37]	Diagnostic bone marrow samples from pediatric acute leukemia patients corresponding to 6 prognostically important leukemia subtypes: 43 T-lineage ALL; 27 E2A-PBX1, 15 BCR-ABL, 79 TEL-AML1, and 20 MLL rearrangements; and 64 "hyperdiploid>50" chromosomes.
Lung cancer [4]	Includes 4 known classes: 139 adenocarcinomas (AD), 21 squamous cell carcinomas (SQ), 20 carcinoids (COID), and 17 normal lung (NL). The AD class is highly heterogeneous, and substructure is known to exist, although not well understood [4].
CNS tumors [25]	Embryonal tumors of the central nervous system (CNS): 10 medulloblastomas (MD); 8 primitive neuroectodermal tumors (PNET); 10 atypical teratoid/rhabdoid tumors (Rhab); 10 malignant gliomas (Glio); and 4 normal cerebellum (Ncer).
Normal tissues [26]	Includes 13 distinct tissue types: breast (5), prostate (9), lung (7), colon (11), germinal center cells (6), bladder (7), uterus (6), peripheral blood monocytes (5), kidney (12), pancreas (10), ovary (4), whole brain (5), cerebellum (3).

distinctions. The number n of gene markers depends on how many genes are differentially expressed with a sufficiently high significance level (0.05) as determined by permutation test [28].

4.3. Results

Tables IV and V summarize the results of the evaluation on simulated data. Tables VI and VII summarize the results on real gene-expression data.

4.3.1. *Simulated data*

We used several simple simulated datasets in order to better understand the proposed methodology, and to answer some basic questions. What would the clustering methodology find in a dataset with no clusters (the datasets `Uniform1` and `Gaussian1`)? How would the proposed methodology behave in the optimal scenario where all features are unam-

Table IV. Estimated number of clusters by consensus clustering (CC) and by the Gap statistic, in combination with hierarchical clustering (HC) and self-organizing map (SOM). Application to simulated data. The numbers between parentheses represent local maxima of the Gap statistic (see text).

Dataset	K_{true}	$\mathbf{CC}_{\text{HC}}$	$\mathbf{CC}_{\text{SOM}}$	$\mathbf{Gap}_{\text{HC}}$	$\mathbf{Gap}_{\text{SOM}}$
`Uniform1`	1	1	1	1	1
`Gaussian1`	1	1	1	3	1
`Gaussian3`	3	3	3	3	3
`Gaussian4`	4	4	4	1	1 (4)
`Gaussian5` ($\lambda = 3$)	5	5	5	1 (5)	1 (5)
`Gaussian5` ($\lambda = 2$)	5	4–5	4	1	1
`Simulated6`	6–7	7	6	7	3
`Simulated4`	4	4	4	4	2

Table V. Adjusted Rand index for naïve-Bayes (NB), hierarchical clustering (HC), consensus clustering with hierarchical clustering (CC_{HC}), and consensus clustering with SOM (CC_{SOM}).

Dataset	**NB**	**HC**	$\mathbf{CC}_{\text{HC}}$	$\mathbf{CC}_{\text{SOM}}$
`Uniform1`	—	—	—	—
`Gaussian1`	—	—	—	—
`Gaussian3`	1.000	1.000	1.000	1.000
`Gaussian4`	0.896	0.768	0.915	0.908
`Gaussian5` ($\lambda = 3$)	0.951	0.932	0.932	0.941
`Gaussian5` ($\lambda = 2$)	0.667	0.522	0.589	0.592
`Simulated6`	0.906	0.986	0.986	0.986
`Simulated4`	1.000	1.000	1.000	1.000

biguously informative about the given cluster distinction (the datasets `Gaussian4` and `Gaussian5`)? Would the presence of non-discriminant features — features whose distribution of values does not change among clusters — worsen the performance of the method, and would the presence of clusters with feature markers of different strength make some cluster harder to find than others (datasets `Simulated4` and `Simulated6`)?

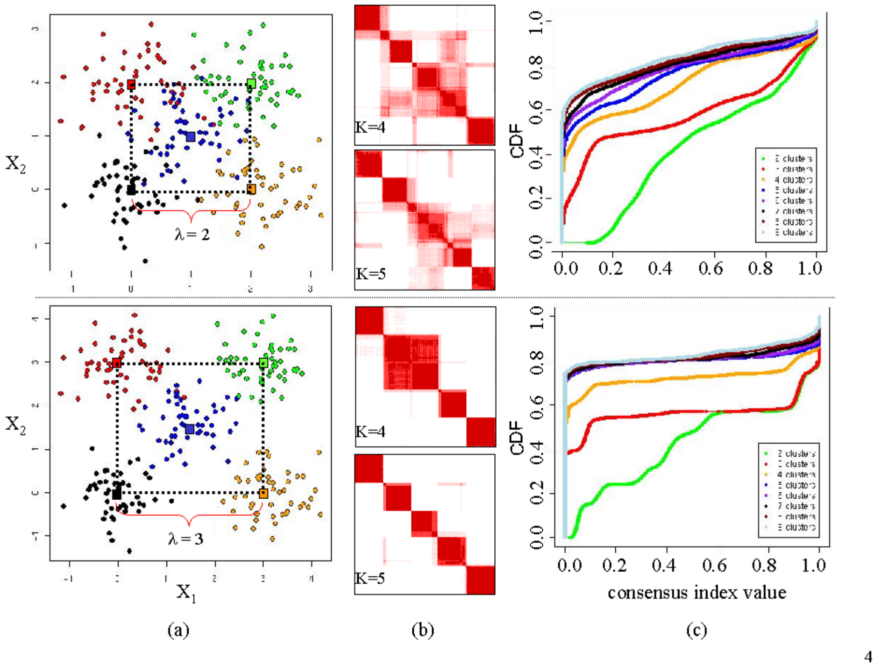

Figure 4. Results of consensus clustering applied to the simulated data `Gaussian5` with distance λ between Gaussians' centers $\lambda = 2$ (top) and $\lambda = 3$ (bottom): a) raw data; b) consensus matrices $\mathcal{M}^{(K)}$ for $K = 4, 5$; c) CDF plots corresponding to the consensus matrices in the range $K = 2, 3, \ldots, 9$.

The general answer is that the simulated data we used do not present a real challenge for the proposed methodology. In most cases we are able to recover the correct number of clusters and to correctly classify most items (those items wrongly classified are items falling in the overlapping region between clusters). A summary of the clustering results is given in Tables IV and V. A few comments on some of the datasets follow.

As expected, for the dataset `Gaussian5` with $\lambda = 3$ (where λ is the distance between Gaussian centers, see the dataset description in Section 4.2.1), the clustering procedure is able to easily recover the correct cluster structure, and to make very few errors when drawing the cluster boundaries. On the other hand, for $\lambda = 2$, the identification of the 5 clusters becomes harder. In this case, 33% of the items generated from the $(\lambda/2, \lambda/2)$ cluster lie closer to the center of some other cluster, thus making the 5-cluster structure harder to detect. Figure 4 shows the plots of the data for $\lambda = 2$ and $\lambda = 3$, the corresponding consensus matrices for $K = 4$ and $K = 5$, and the corresponding CDF plots for the consensus matrices in the range $K = 2, \ldots, 9$. The increased

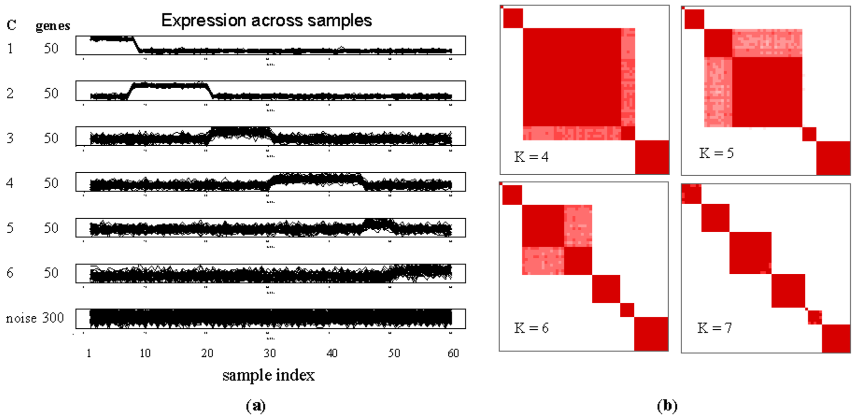

Figure 5. Application of consensus clustering to the simulated data `Simulated6`: a) expression profiles for each gene within each cluster; b) consensus matrices $\mathcal{M}^{(K)}$ for $K = 4, 5, 6, 7$. Notice the presence of a singleton cluster, corresponding to sample 8, in all four matrices (see text).

difficulty of the clustering task, resulting from the increased overlapping among clusters, is reflected in the corresponding Rand indices, shown in Table V. They correspond to an error rate of about 2% for $\lambda = 3$, increasing to about 20% for $\lambda = 2$.

Application of the Gap statistic to the datasets `Gaussian4` and `Gaussian 5` yields puzzling results. In all cases, if we accept the proposed selection criterion, we should select 1 as the correct number of clusters. If we look at the plot of the Gap statistic values for different values of K (not shown), a local maximum at the correct K is present (with the exception of the Gap statistic for HC applied to `Gaussian4`). However, the proposed selection procedure would not allow us to reach that maximum.

Finally, we come to the dataset `Simulated6`. When we created this dataset, we unintentionally introduced an additional complication: the 8th sample is claimed by both the first and the second cluster (that is, in sample 8 both the genes associated with cluster 1 and cluster 2 are up-regulated). As a result, this sample cannot be exclusively claimed by either clusters. This is manifested in the consensus matrices shown in Figure 5.b; sample 8 claims a cluster for itself. As a consequence, the cleanest consensus matrix is obtained for $K = 7$ rather than 6, which is confirmed by looking at the plot of the corresponding CDFs, and the corresponding $\Delta(K)$'s, which level off at $K = 7$ (not shown). With the exception of sample 8, all other samples are correctly classified, thus yielding a very high value of the Rand index. We described this case in some detail because a very similar pattern of behavior is observed in the analysis of the St. Jude leukemia data described in the next section.

4.3.2. *Gene-expression data*

In general, when applied to gene-expression data, consensus clustering with HC outperformed consensus clustering with SOM, and both methods outperformed the Gap statistic as a method to estimate the number of clusters. In most datasets, consensus clustering was able to select the correct number of clusters, and to establish the cluster membership with a high level of accuracy (as measured by the adjusted Rand index). Predictably, the estimation of the number of clusters and cluster assignment was most difficult in the CNS tumors and the Normal tissues datasets, given the relatively small sample size for the given number of classes. A few comments on some of the datasets follow.

When applied to the leukemia dataset [13], consensus clustering with HC selects 5 clusters as its optimal number, with two of the clusters corresponding exactly to AML and T-lineage ALL, and with the three remaining clusters further partitioning the B-lineage ALL (with 4, 4, and 11 items respectively). It is interesting to note that the Gap statistic also selects 5 as the optimal number of clusters. Consensus clustering with SOM selects 4 clusters, with two clusters splitting the B-lineage type. It is widely accepted that the class of acute lymphoblastic leukemia can be farther partitioned into biologically meaningful sub-classes (see, for example, the St. Jude leukemia data, discussed later in this section), although the composition and nature of these sub-classes is not as well accepted. Therefore, it is possible that the subclass structure we discover within the B-lineage ALLs reflects a biologically meaningful distinction.

When applying consensus clustering with HC to the St. Jude leukemia dataset [37], if we only take into account the consensus distribution (and the proportion increase in the area under the CDF), the suggested number of clusters is 5. This is also the number suggested by the Gap statistic. However, if we look at the consensus matrices $\mathcal{M}^{(K)}$ for K between 5 and 9, shown in Figure 6, a 6-cluster structure clearly emerges, and with cleaner boundaries than for $K = 5$. The best separation is obtained at $K = 7$, where 6 of the clusters correspond almost perfectly

Table VI. Estimated number of clusters by consensus clustering (CC) and by the Gap statistic, in combination with hierarchical clustering (HC) and self-organizing map (SOM). Application to gene-expression data. In parentheses is the estimated number of clusters based on visual inspection of the consensus matrices (when this differ from the one based on the consensus distribution).

Dataset	K_{true}	$\mathbf{CC}_{\text{HC}}$	$\mathbf{CC}_{\text{SOM}}$	$\mathbf{Gap}_{\text{HC}}$	$\mathbf{Gap}_{\text{SOM}}$
Leukemia	3	5	4	5	4
Novartis	4	4	4	4	4
St. Jude	6	5 (6)	5/7 (6)	5	11
Lung cancer	4+	5	5 (7)	5	7
CNS tumors	5	5	5/6	6	4
Normal tissues	13	7	4/5	12	7

Table VII. RAND index for naïve-Bayes (NB), hierarchical clustering (HC), consensus clustering with hierarchical clustering (CC_{HC}), and consensus clustering with SOM (CC_{SOM}). In parentheses is the Rand index corresponding to the partition into K_{true} classes (when this differ from the estimated K).

Dataset	**NB**	**HC**	$\mathbf{CC}_{\text{HC}}$	$\mathbf{CC}_{\text{SOM}}$
Leukemia	1.00	0.648 (0.46)	0.648 (1.0)	0.721 (0.6)
Novartis-tissue	0.946	0.83	0.921	0.897
St. Jude	0.971	0.949	0.948	0.825
Lung cancer	0.904	0.307 (0.28)	0.310 (0.28)	0.233 (0.22)
CNS tumors	0.632	0.628	0.549	0.429
Normal tissues	0.655	0.457 (0.572)	0.457 (0.572)	0.214 (0.487)

to the 6 known sub-types, and the remaining cluster contains a single sample (a “hyperdiploid>50” sample, see Table III). This one-sample cluster is also the reason why the consensus matrix for $K = 6$ fails to perfectly separate two of the six known subtypes, as one of the six clusters is “sacrificed” to this single outlyer sample. This pattern of behavior is very similar to the one encountered when analyzing the simulated dataset `Simulated6`. These two datasets allow us to illustrate the advantage of being able to visually inspect the cluster structure and stability by looking at the consensus matrices, which in turn allows us to correct the cluster number estimates based exclusively on the consensus distribution.

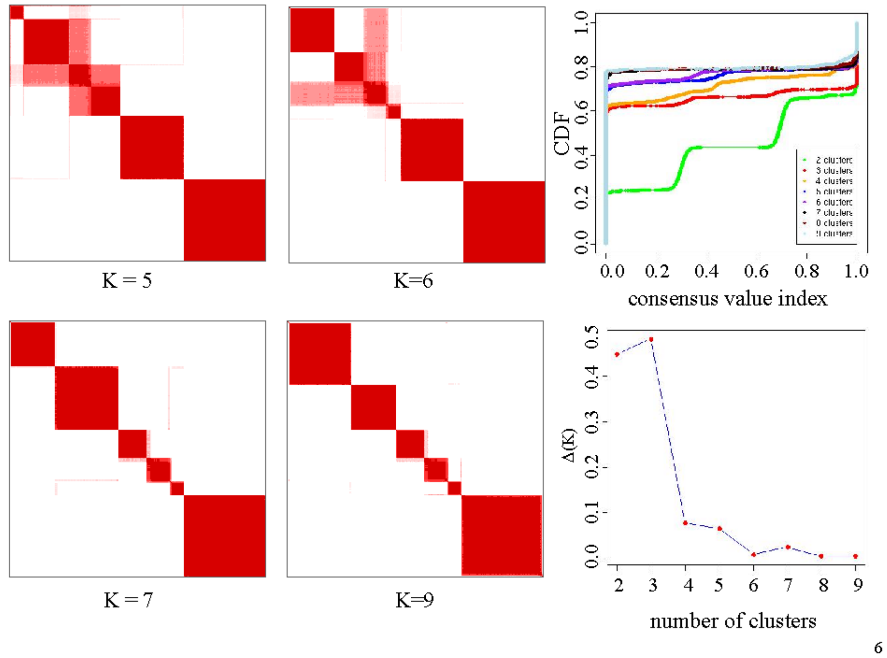

Figure 6. Consensus clustering applied to the St. Jude leukemia data: consensus matrices $\mathcal{M}^{(K)}$ for $K = 5, 6, 7, 9$.

Application of consensus clustering with HC to the lung cancer tissues dataset [4] yields an estimated number of clusters of 5. Three of the five clusters correspond rather cleanly to the non-AD types, namely, COID, SQ, and NL, while the additional two classes correspond to ADs. All the errors in cluster assignment (29/197) involve AD samples.[2] Application of consensus clustering with SOM also yields an estimated number of clusters of 5 based on the consensus distribution. However, if we inspect the consensus matrices (not shown), we clearly see that the matrices between 5 and 7 all have a very clean profile, with cleanly demarcated diagonal red blocks on a perfectly white background. The 7-cluster partition is also in better agreement with the known 4 types. In fact, this partition identifies the three non-AD types rather cleanly, while allocating the remaining 4 clusters to the AD samples.

Application of consensus clustering to the normal tissues dataset [26] did not return the correct number of clusters. However, it should be clear that given the large number of classes (13), and the small number

[2] The class of adenocarcinomas is highly heterogeneous, and it is widely accepted that clinically relevant AD subtypes exist, although their molecular and clinical profile is not well established or understood [4].

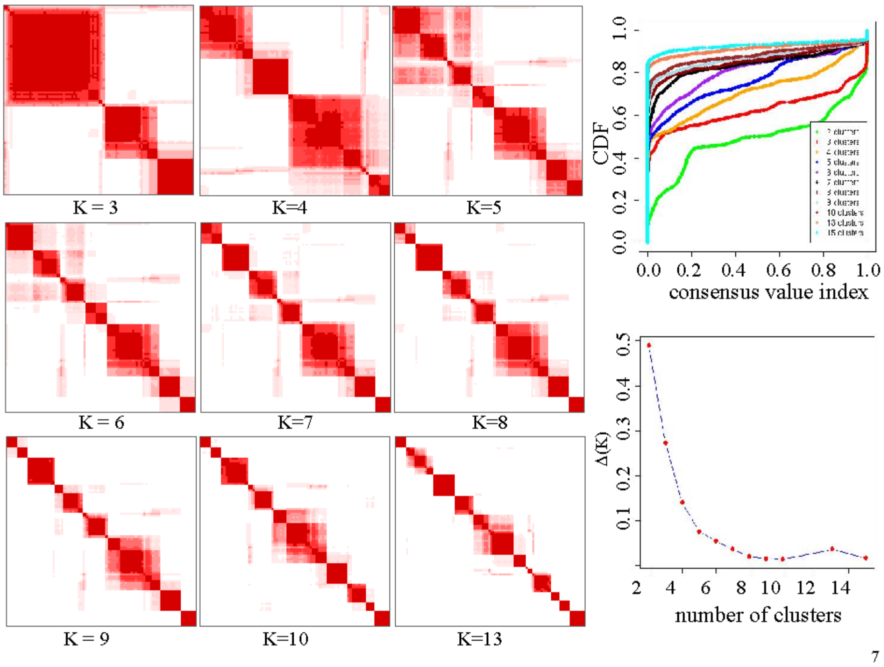

Figure 7. Consensus clustering applied to the normal tissues data.

of samples per class (for a total of 90 samples), the hope of recovering a 13-class distinction was rather slim. By looking at the consensus distribution (by both considering the proportion increase of the area under the CDF, and by visual inspection of the CDFs and the heat maps), $K = 7$ is the largest number of clusters that we can reasonably consider. If we follow the algorithm recommendation, and stop the partitioning of the data at 7 clusters, it is worth pointing out that most of the discovered clusters correspond to fairly clean unions of known types. Application of consensus clustering with SOM estimates an even lower number of clusters. Inspection of the consensus distributions and of the consensus matrices suggests not going above 5 clusters. The low Rand index confirms the fact that most clusters are a mixture of several tissue types, with several tissue types distributed across more than one cluster.

4.4. Discussion

The results of the application of consensus clustering to both simulated and real data are encouraging. In most cases the methodology is capable of recovering the correct number of clusters, and assigning most items to the correct clusters. Furthermore, when partitioning the data using the consensus matrix as the similarity measure rather than the usual Euclidean distance, the cluster assignments tend to be more accurate. This can be seen by comparing the Rand indices for the two methods (the columns HC and $\mathrm{CC}_{\mathrm{HC}}$ in Tables V and VII).

As expected, the data set where the method performed worse was the normal tissue data (given the small sample size relative to the number of classes). We believe the main lesson to be learned from the analysis of this data set is that cluster analysis can only go as far, and that an effort should be made to collect fairly homogeneous data sets where we can expect the number of clusters to be reasonably small and commensurate to the sample size available. In other words, this is an example of what a poorly designed experimental design for cluster analysis might look like, rather than evidence for or against the proposed clustering methodology.

It is important to notice that the results are dependent on the inner-loop clustering of choice (HC and SOM in the experiments), with consensus clustering based on HC producing slightly better results than consensus clustering based on SOM. This points to the fact that every clustering method has its own idiosyncrasies, related to the (implicit or explicit) measure of similarity it uses to compare and group data items.

A related issue is data normalization, and how the choice of normalization can affect the clustering results. As previously pointed out, the data used with HC were of necessity row- and column-normalized, so as to produce fairly balanced hierarchical trees. The reported SOM results were also based on the same normalized data. Although not reported, we also applied SOM to non-normalized data, and the results did not always agree with those based on normalized data. Therefore, the selection of the inner-loop clustering algorithm, as well as the choice of the data normalization methodology to be used, are both sensitive issues that need to be taken into account when performing clustering analysis.

Finally, the experimental evaluation shows that the consensus distribution described in Section 3.3.1 is a useful criterion to select the number of clusters, but we do not advocate its use in isolation. Often, the visual inspection of the ordered consensus matrices can be as informative, and help disambiguate the information provided by the consensus distribution. Ultimately, in the experiments we carried out, what seemed to work best was a model selection process based on a combination of the information coming from the consensus distribution and from the visual inspection of the ordered consensus matrices.

5. Conclusions

In this paper we introduced a clustering methodology based on resampling that allows for the estimation of the number of clusters in a dataset, the assessment of the stability of the putative clusters found, and the visualization of the clustering results. The method, by capturing the consensus among several clustering runs, attempts to produce data partitions that are more robust than the ones we may expect to obtain by application of a single clustering algorithm to the observed data.

A natural extension of the method is to use it to represent the "meta-consensus" across clustering algorithms. For example, with regard to the experiments described in the previous section, we could combine the consensus matrices corresponding to HC and SOM.

Another extension, partially explored in [4], is to use consensus clustering with probabilistic model-based clustering in the inner-loop (e.g., AutoClass [6]). Since methods of model-based clustering usually provide their own estimate of the number of clusters, the consensus matrix in this case would represent the consensus across multiple runs with each run returning a partition into a possibly different number of clusters. In other words, rather than having several consensus matrices for different K's, we would have a single consensus matrix. This would preclude us from using the consensus distribution to estimate the number of clusters. However, the visualization of the sorted consensus matrix could still be used to validate the recommended number of clusters, as well as to determine the cluster assignments.

On a related subject, it should be noted that the consensus matrix can easily accommodate the fractional cluster assignments usually output by probabilistic clustering algorithms. The entries of the connectivity matrix of Equation 1 will need to be properly modified so as to reflect the uncertainty in cluster membership. In particular, let $P(i \in C_k \mid D)$ denote the probability output by the clustering algorithm that item i belongs to cluster C_k, and let K be the number of clusters. Then, the connectivity matrix entries will be computed as follows:

$$M^{(h)}(i,j) = \sum_{k=1}^{K} P(i \in C_k \mid D) P(j \in C_k \mid D) \,. \tag{9}$$

with Equation (9) reducing to Equation (1) when all probabilities output by the clustering algorithms are either 0 or 1.

Finally, in this paper we have described the consensus distribution as summarized in the CDFs and the $\Delta(K)$'s as the measure of consensus we use to estimate the number of clusters. However, once we abstract the problem into one of finding the number of clusters that yield the "cleanest" consensus matrix, it may be possible to devise other measures that try to quantify this cleanliness. We are in the process of evaluating some of these alternative measures.

The methodology for consensus clustering described here, with HC and SOM as choices, was implemented in Java, and will be made part of the next release of GeneCluster [34].

Acknowledgments

We thank all the members of the Cancer Genomics group for useful discussions and assistance in the preparation and use of the data analyzed.

Appendix

A. The adjusted Rand index

The adjusted Rand index is a measure of agreement between alternative data partitions that can be used even when considering partitions with different numbers of clusters [16, 24]. In this appendix, we only give the formula for its computation. For examples and a more detailed explanation of its derivation, see, e.g., [38].

Let $\boldsymbol{P}_a = \{P_{a1}, P_{a2}, \ldots, P_{aK_a}\}$ and $\boldsymbol{P}_b = \{P_{b1}, P_{b2}, \ldots, P_{bK_b}\}$ be two partitions of the dataset D, with K_a and K_b not necessarily equal. The adjusted Rand index assumes the generalized hypergeometric distribution as the model of randomness, i.e., the $\boldsymbol{P}_a$ and $\boldsymbol{P}_b$ partitions are picked at random but with the constraint that the number of objects in the classes and the numbers of clusters are fixed.

Let N_{ij} be the number of items of D that are both members of cluster P_{ai} and of cluster P_{bj}. These N_{ij} entries basically define a confusion matrix (with rows indexed by $\boldsymbol{P}_a$, and columns indexed by $\boldsymbol{P}_b$) relating the cluster assignments in $\boldsymbol{P}_a$ with the cluster assignments in $\boldsymbol{P}_b$. Accordingly, let $N_{i.}$ denote column sums (i.e., the number of items members of cluster P_{ai} irrespective of their membership in $\boldsymbol{P}_b$), and let $N_{.j}$ denote row sums (i.e., the number of items members of cluster P_{bj} irrespective of their membership in $\boldsymbol{P}_a$).

Then the adjusted Rand index r is computed as follows:

$$r = \frac{\sum_{ij}\binom{N_{ij}}{2} - \left[\sum_i \binom{N_{i.}}{2} \sum_j \binom{N_{.j}}{2}\right] / \binom{N}{2}}{\frac{1}{2}\left[\sum_i \binom{N_{i.}}{2} + \sum_j \binom{N_{.j}}{2}\right] - \left[\sum_i \binom{N_{i.}}{2} \sum_j \binom{N_{.j}}{2}\right] / \binom{N}{2}} \, . \tag{10}$$

The measure thus defined ranges between 0 and 1, with 1 corresponding to perfect agreement between the two partitions. It can be shown that the adjusted Rand index has an expected value of 0 for two random partitions.

References

1. Banfield, J. and A. E. Raftery: 1993, 'Model-based Gaussian and non-Gaussian clustering'. *Biometrics* **49**(803–821).
2. Bar-Joseph, Z., E. D. Demaine, D. K. Gifford, A. M. Hamel, T. S. Jaakkola, and N. Srebro: 2002, 'K-ary Clustering with Optimal Leaf Ordering for Gene Expression Data'. *Bioinformatics*. To appear.
3. Ben-Hur, A., A. Elisseeff, and I. Guyon: 2002, 'A Stability Based Method for Discovering Structure in Clustered Data'. In: *Pacific Symposium on Biocomputing 2002*, Vol. 7. Lihue, Hawaii, pp. 6–17.
4. Bhattacharjee, A., W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker, and M. Meyerson: 2001, 'Classification of Human Lung Carcinomas by mRNA Expression Profiling Reveals Distinct Adenocarcinomas Sub-classes'. *Proceedings of the National Academy of Sciences* **98**(24), 13790–13795.
5. Bock, H.: 1985, 'On Some Significance Tests in Cluster Analysis'. *Journal of Classification*.
6. Cheeseman, P. and J. Stutz: 1996, 'Bayesian Classification (AutoClass): Theory and Results'. In: U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurasamy (eds.): *Advances in Knowledge Discovery and Data Mining*. MIT Press, pp. 153–180.
7. Chickering, D. M. and D. Heckerman: 1997, 'Efficient Approximation for the Marginal Likelihood of Bayesian Networks with hidden variables'. *Machine Learning* **29**, 181–212.
8. Cowell, F. A.: 1995, *Measuring inequality*. New York: Prentice Hall.
9. Duda, R. O. and P. E. Hart: 1973, *Pattern Classification and Scene Analysis*. John Wiley & Sons.
10. Dudoit, S. and J. Fridlyand: 2002, 'A Prediction-based Resampling Method for Estimating the Number of Clusters in a Dataset'. *Genome Biology* **3**(7), 1–21.
11. Efron, B. and R. J. Tibshirani: 1994, *An Introduction to the Bootstrap*, No. 57 in Monographs on Statistics and Applied Probability. CRC Press.
12. Eisen, M. B., P. T. Spellman, P. O. Brown, and D. Botstein: 1998, 'Cluster Analysis and Display of Genome-Wide Expression Patterns'. *Proceedings of the National Academy of Sciences* **95**, 14863–14868.
13. Golub, T. R., D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, , and E. Lander: 1999, 'Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression'. *Science* **286**(5439), 531–537.
14. Hartigan, J. A.: 1978, 'Asymptotic Distributions for Clustering Criteria'. *Annals of Statistics* **6**(1), 117–131.
15. Hastie, T., R. Tibshirani, and J. Friedman: 2001, *The Elements of Statistical Learning*, Statistics. New York: Springer.

16. Hubert, L. and P. Arabie: 1985, 'Comparing Partitions'. *Journal of Classification* **2**, 193–218.
17. Jain, A. K. and R. C. Dubes: 1988, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice Hall.
18. Jain, A. K. and J. Moreau: 1988, 'Bootstrap techniques in cluster analysis'. *Pattern Recognition* **20**, 547–568.
19. Kass, R. E. and A. E. Raftery: 1995, 'Bayes Factors'. *Journal of the American Statistical Association* **90**, 773–795.
20. Kohonen, T.: 1990, 'The Self-Organizing Map'. *Proceedings of the IEEE* **78**(9), 1464–1480.
21. Kohonen, T.: 1997, *Self-Organizing Maps*, Information Sciences. Springer.
22. Levine, E. and E. Domany: 2001, 'Resampling Method for Unsupervised Estimation of Cluster Validity'. *Neural Computation* **13**(11), 2573–2593.
23. Milligan, G. and M. Cooper: 1985, 'An examination of procedures for determining the number of clusters in a data set'. *Psyochometrika* **50**, 159–179.
24. Milligan, G. and M. Cooper: 1986, 'A study of the comparability of external criteria for hierarchical cluster analysis'. *Multivariate Behavioral Research* **21**, 441–458.
25. Pomeroy, S., P. Tamayo, M. Gaasenbeek, L. M. S. M. Angelo, M. E. McLaughlin, J. Y. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub: 2002, 'Gene Expression-Based Classification and Outcome Prediction of Central Nervous System Embryonal Tumors'. *Nature* **415**(6870), 436–442.
26. Ramaswamy, S., P. Tamayo, R. Rifkin, S. Mukherjee, C.-H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub: 2001, 'Multi-Class Cancer Diagnosis Using Tumor Gene Expression Signatures'. *Proceedings of the National Academy of Sciences* **98**(26), 15149–15154.
27. Ramoni, M., P. Sebastiani, and I. S. Kohane: 2002, 'Cluster analysis of gene expression dynamics'. *Proceedings of the National Academy of Sciences* **99**(14), 9121–9126.
28. Slonim, D. K., P. Tamayo, J. P. Mesirov, T. R. Golub, and E. S. Lander: 2000, 'Class Prediction and Discovery Using Gene Expression Data'. In: *RECOMB 2000: The Fourth Annual International Conference on Research in Computational Molecular Biology*. Tokyo, Japan, pp. 263–272.
29. Su, A. I., M. P. Cooke, K. A. Ching, Y. Hakak, J. R. Walker, T. Wiltshire, A. P. Orth, R. G. Vega, L. M. Sapinoso, A. Moqrich, A. Patapoutian, G. M. Hampton, P. G. Schultz, and J. B. Hogenesch: 2002, 'Large-scale Analysis of the Human and Mouse Transcriptomes'. *Proceedings of the National Academy of Sciences* **99**(7), 4465–447.
30. Tamayo, P., D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Godlub: 1999, 'Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation'. *Proceedings of the National Academy of Sciences* **96**, 2907–2912.
31. Tibshirani, R., G. Walther, D. Botstein, and P. Brown: 2001a, 'Cluster Validation by Prediction Strength'. Unpublished manuscript (`http://www-stat.stanford.edu/~tibs/ftp/predstr.pdf`).

32. Tibshirani, R., G. Walther, and T. Hastie: 2001b, 'Estimating the Number of Clusters in a Dataset via the Gap Statistic'. *Journal of the Royal Statistical Society B* **63**(2), 411–423.
33. Titterington, D., A. Smith, and U. Makov: 1985, *Statistical Analysis of Finite Mixture Distributions*. New York: Wiley.
34. Todd Golub et. al: 2002, 'GeneCluster 2.0'. `http://www-genome.wi.mit.edu/cancer/software/genecluster2/gc2.html`.
35. West, M.: 2002, 'Bayesian Factor Regression Models in the "Large p, Small n" Paradigm'. *Bayesian Statistics* **7**. To appear.
36. West, M., C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. A. Olson Jr., J. R. Marks, and J. R. Nevins: 2001, 'Predicting the Clinical Status of Human Breast Cancer by Using Gene Expression Profiles'. *Proceedings of the National Academy of Sciences* **98**(20), 11462–11467.
37. Yeoh, E.-J., M. E. Ross, S. A. Shurtleff, W. K. Williams, D. Patel, R. Mahfouz, F. G. Behm, S. C. Raimondi, M. V. Relling, A. Patel, C. Cheng, D. Campana, D. Wilkins, X. Zhou, J. Li, H. Liu, C.-H. Pui, W. E. Evans, C. Naeve, L. Wong, and J. R. Downing: 2002, 'Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling'. *Cancer Cell* **1**(2).
38. Yeung, K. Y., C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo: 2001a, 'Model-based clustering and data transformations for gene expression data'. *Bioinformatics* **17**(10), 977–987.
39. Yeung, K. Y., D. R. Haynor, and W. L. Ruzzo: 2001b, 'Validating clustering for gene expression data'. *Bioinformatics* **17**(4).