Cluster Analysis in DNA Microarray Experiments

Sandrine Dudoit and Robert Gentleman

Bioconductor Short Course

Winter 2002

Outline

- Overview of cluster analysis.
- Clustering gene expression data.
- Clustering methods
  - Partitioning methods.
  - Hierarchical methods.
- Estimating the number of clusters.
- Other topics
  - Inference.
  - Outliers.
  - Hybrid methods.
  - Bagged clustering.

Supervised vs. unsupervised learning

Task. Assign objects to classes on the basis of measurements made on these objects.

Unsupervised learning. The classes are **unknown** a priori and need to be "discovered" from the data.

a.k.a. cluster analysis; class discovery; unsupervised pattern recognition.

Supervised learning. The classes are **predefined** and the task is to understand the basis for the classification from a set of labeled objects. This information is then used to classify future observations.

a.k.a. classification; discriminant analysis; class prediction; supervised pattern recognition.

Cluster analysis

Associated with each object is a set of G measurements which form the **feature vector**, $\mathbf{X} = (X_1, \ldots, X_G)$. The feature vector $\mathbf{X}$ belongs to a feature space $\mathcal{X}$ (e.g., $\Re^G$).

The task is to identify groups, or **clusters**, of *similar* objects on the basis of a set of feature vectors, $\mathbf{X}_1 = \mathbf{x}_1, \ldots, \mathbf{X}_n = \mathbf{x}_n$.

Cluster analysis

Clustering is in some sense a more difficult problem than classification. In general, all the issues that must be addressed for classification must also be addressed for clustering. In addition, with clustering,

- there is no learning set of labeled observations;
- the number of groups is usually unknown;
- implicitly, one must have already selected both the relevant features and distance measure;
- the goals can be quite vague: "*Find some interesting and important clusters in my data*";
- most of the algorithms that are appealing are computationally too complex to have exact solutions; approximate solutions are used instead and reproducibility becomes an issue.

Cluster analysis

Clustering involves several distinct steps.

First, a suitable distance between objects must be defined, based on relevant features.

Then, a clustering algorithm must be selected and applied.

The results of a clustering procedure can include both the number of clusters K (if not prespecified) and a set of n cluster labels $\in \{1, \ldots, K\}$ for the n objects to be clustered.

Appropriate choices will depend on the questions being asked and available data.

Cluster analysis

Clustering procedures fall into two broad categories.

- **Hierarchical methods**, either **divisive** or **agglomerative**. These methods provide a hierarchy of clusters, from the smallest, where all objects are in one cluster, through to the largest set, where each observation is in its own cluster.
- **Partitioning methods**. These usually require the specification of the number of clusters. Then, a mechanism for apportioning objects to clusters must be determined.

Most methods used in practice are agglomerative hierarchical methods. In large part, this is due to the availability of efficient exact algorithms.

Distance

The feature data are often transformed to an $n \times n$ **distance** or **similarity matrix**, $\mathbf{D} = (d_{ij})$, for the n objects to be clustered.

Once a distance measure between individual observations has been chosen, one must often also define a distance measure *between clusters*, or groups of observations (e.g., average, single, and complete linkage agglomeration).

Different choices here can greatly affect the outcome.

More details in the lecture *Distances and Expression Measures*.

R clustering software

- `class` package: Self Organizing Maps (`SOM`).
- `cluster` package:
  - AGglomerative NESting (`agnes`),
  - Clustering LARe Applications (`clara`),
  - DIvisive ANAlysis (`diana`),
  - Fuzzy Analysis (`fanny`),
  - MONothetic Analysis (`mona`),
  - Partitioning Around Medoids (`pam`).
- `e1071` package:
  - Fuzzy C-means clustering (`cmeans`),
  - Bagged clustering (`bclust`).
- `mva` package:
  - Hierarchical clustering (`hclust`, `cophenetic`),
  - k-means (`kmeans`).

Specialized summary, plot, and print methods for clustering results.

Gene expression data

Gene expression data on G genes (features) for n mRNA samples (observations)

$$X_{G\times n} = \begin{array}{c} \text{mRNA samples} \\ \left[\begin{array}{cccc} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{G1} & x_{G2} & \dots & x_{Gn} \end{array}\right] \end{array} \quad \text{Genes}$$

x_{gi} = expression measure for gene g in mRNA sample i.

An array of conormalized arrays.

Gene expression data

Features correspond to expression levels of different genes; possible classes include tumor types (e.g., ALL, AML), clinical outcomes (survival, non-survival), and are labeled by $\{1, 2, \ldots, K\}$.

Gene expression data on G genes (features) for n mRNA samples (observations)

$$\begin{aligned} \mathbf{x}_i &= \left(x_{i1}, x_{i2}, \ldots, x_{iG}\right) \\ &\quad \text{– gene expression profile / feature vector for sample } i \\ y_i &= \text{class for sample } i, \qquad i = 1, \ldots, n. \end{aligned}$$

Other covariates such as age, sex may also be important and could be included in the analysis.

Gene expression data

Most efforts to date have involved clustering only the expression measures collected on a number of different genes and samples.

However, there is likely to be a need for incorporating other data into the analysis, such as sample level covariates and biological metadata.

For example, a common task is to determine whether or not gene expression data can reliably identify or classify different types of a disease. However, one might ask as well whether such data improve our ability to classify over already available sample level covariate data.

Clustering gene expression data

- One can cluster genes and/or samples (arrays).
- Clustering leads to readily interpretable figures.
- Clustering strengthens the signal when averages are taken within clusters of genes (Eisen et al., 1998).
- Clustering can be helpful for identifying gene expression patterns in time or space.
- Clustering is useful, perhaps essential, when seeking new subclasses of cell samples (tumors, etc).
- Clustering can be used for quality control: compare array/gene clustering results to experimental variables such as array batch, mRNA amplification method, lab, experimenter, etc.

Clustering gene expression data

Cluster genes (rows)

- to identify groups of co-regulated genes, e.g., using a large number of yeast experiments;
- to identify spatial or temporal expression patterns;
- to reduce redundancy (cf. feature selection) in prediction;
- to detect experimental artifacts;
- for display purposes.

Transformations of the expression data matrix using linear modeling may be useful in this context:

$$\text{genes} \times \text{arrays} \Longrightarrow \text{genes} \times \text{estimated effects.}$$

Cf. *Microarray Experimental Design and Analysis*, Summer 2002.

Clustering gene expression data

Cluster samples or arrays (columns)

- to identify new classes of biological samples, e.g., new tumor classes, new cell types;
- to detect experimental artifacts;
- for display purposes.

Cluster both rows and columns at once.

Clustering gene expression data

Clustering can be employed for quality control purposes. The clusters that obtain from clustering arrays/genes should be compared with different experimental conditions such as:

- batch or production order of the arrays;
- batch of reagents;
- mRNA amplification procedure;
- technician;
- plate origin of clones, etc.

Any relationships observed here should be considered as a potentially serious source of bias.

Tumor classification using gene expression data

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.

Also, it is likely that the existing classes are heterogeneous and comprise diseases which are molecularly distinct and follow different clinical courses.

Tumor classification using gene expression data

DNA microarrays may be used to characterize the molecular variations among tumors by monitoring gene expression profiles on a genomic scale.

This may lead to a finer and more reliable classification of tumors, and to the identification of marker genes that distinguish among these classes.

Eventual clinical implications include an improved ability to understand and predict cancer survival.

Tumor classification using gene expression data

There are three main types of statistical problems associated with tumor classification:

1. the identification of new tumor classes using gene expression profiles – **unsupervised learning**;
2. the classification of malignancies into known classes – **supervised learning**;
3. the identification of marker genes that characterize the different tumor classes – **feature selection**.

Example: Row and column clustering

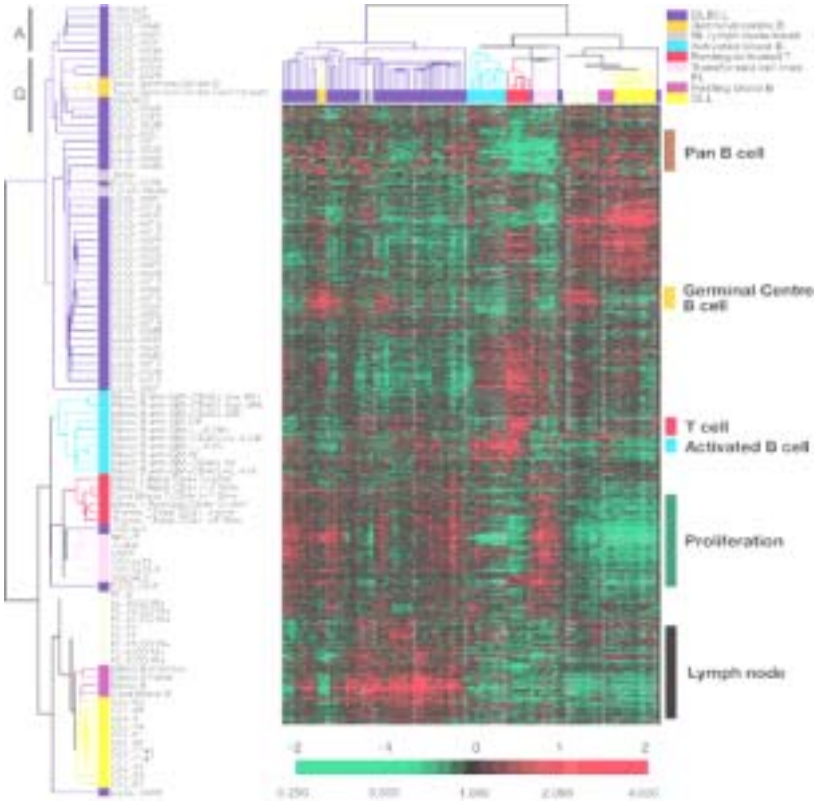

Figure 1: Alizadeh et al. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*.

Clustering gene expression data

Preliminary questions

- Which genes / arrays to use?
- Which transformation/standardization?
- Which distance function?
- Which clustering algorithm?

Answers will depend on the biological problem.

Clustering gene expression data

Important questions (which are generic)

- How many clusters?
- How reliable are the clustering results?
  - Statistical inference: distributional properties of clustering results.
  - Assessing the strength/confidence of cluster assignments for individual observations;
  - Assessing cluster homogeneity.

Partitioning methods

- Partition the data into a **prespecified** number K of mutually exclusive and exhaustive groups.
- Iteratively reallocate the observations to clusters until some criterion is met, e.g., minimize within-cluster sums-of-squares.
- Examples:
  - k-means; extension to fuzzy k-means;
  - Partitioning Around Medoids – PAM (Kaufman & Rousseeuw, 1990);
  - Self-Organizing Maps – SOM (Kohonen, 2001);
  - model-based clustering,
    e.g., Gaussian mixtures in Fraley & Raftery (1998, 2000) and McLachlan et al. (2001).

Partitioning around medoids

Partitioning around medoids or **PAM** of Kaufman and Rousseeuw (1990) is a partitioning method which operates on a distance matrix, e.g., Euclidean distance matrix.

For a prespecified number of clusters K, the PAM procedure 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.

Partitioning around medoids

The goal is to find K medoids, $\mathbf{M} = (\mathbf{m}_1, \ldots, \mathbf{m}_K)$, which minimize the sum of the distances of the observations to their closest medoid, that is,

$$\mathbf{M}^* = \text{argmin}_{\mathbf{M}} \sum_i \min_k d(\mathbf{x}_i, \mathbf{m}_k).$$

PAM can be applied to general data types and tends to be more robust than k-means.

Silhouette plots

Rousseeuw (1987) suggested a graphical display, the **silhouette plot**, which can be used to: (i) select the number of clusters and (ii) assess how well individual observations are clustered.

The **silhouette width** of observation i is defined as

$$sil_i = (b_i - a_i)/\max(a_i, b_i),$$

where a_i denotes the average distance between i and all other observations in the cluster to which i belongs, and b_i denotes the minimum average distance of i to objects in other clusters.

Intuitively, objects with large silhouette width sil_i are well-clustered, those with small sil_i tend to lie between clusters.

Silhouette plots

For a given number of clusters K, the overall **average silhouette width** for the clustering is simply the average of sil_i over all observations i, $\bar{sil} = \sum_i sil_i/n$.

Kaufman & Rousseeuw suggest estimating the number of clusters K by that which gives the largest average silhouette width, $\bar{sil}$.

Note that silhouette widths may be computed for the results of any partitioning clustering algorithm.

Partitioning around medoids

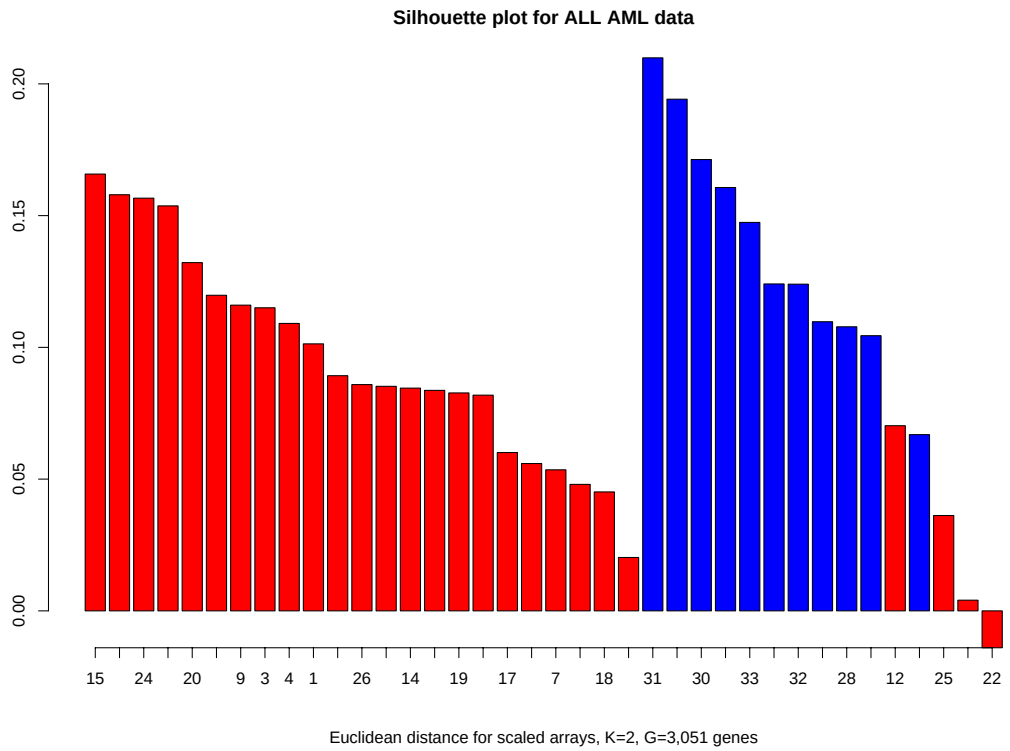

Figure 2: Golub et al. (1999) ALL AML data. Silhouette plot for PAM, red=ALL, blue=AML.

PAMSIL

PAMSIL. van der Laan, Pollard, & Bryan (2001).

Replace PAM criteria function with average silhouette.

	PAM	PAMSIL
Criteria	$-\sum_i \min_k d(\mathbf{x}_i, \mathbf{m}_k)$	$\sum_i sil_i$
Algorithm	Steepest ascent	Steepest ascent
Starting values	Build	PAM, random
K	Given or data-adaptive	Given or data-adaptive
Overall performance	"Robust"	"Efficient"
Splitting large clusters	Yes	No
Outliers	Ignore	Identify

Hierarchical methods

- Hierarchical clustering methods produce a **tree** or **dendrogram**.
- They avoid specifying how many clusters are appropriate by providing a partition for each K. The partitions are obtained from cutting the tree at different levels.
- The tree can be built in two distinct ways
  - bottom-up: **agglomerative** clustering;
  - top-down: **divisive** clustering.

Hierarchical methods

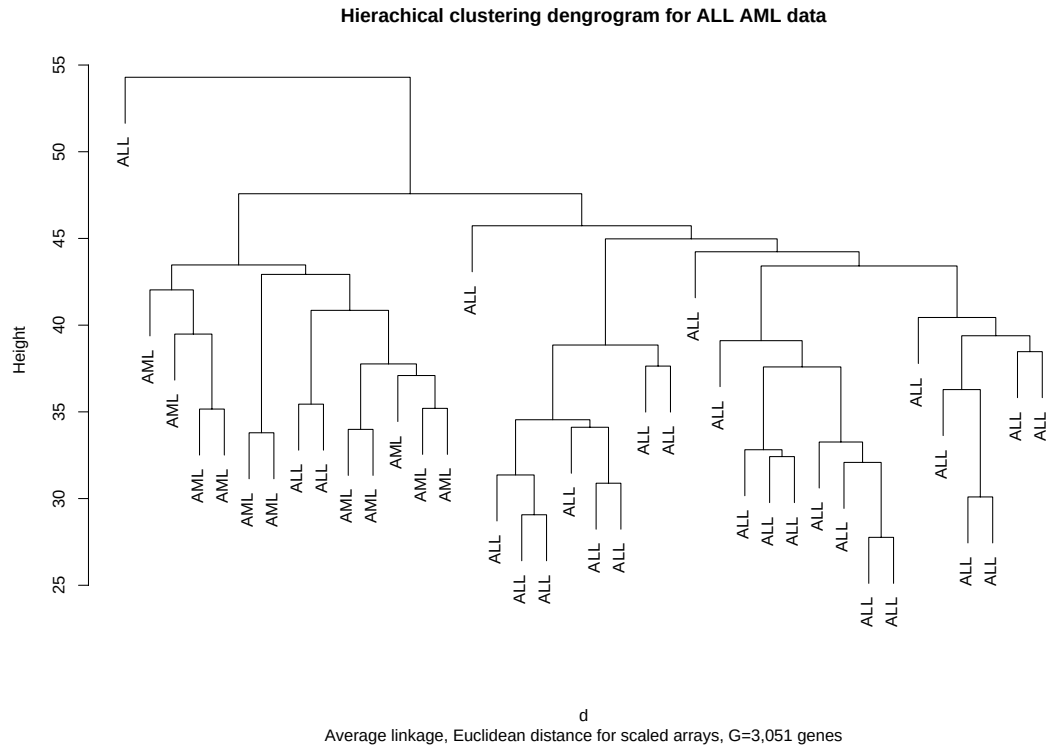

Figure 3: Golub et al. (1999) ALL AML data. Dendrogram for agglomerative hierarchical clustering.

Agglomerative methods

- Start with n mRNA sample (or G gene) clusters.
- At each step, merge the two closest clusters using a measure of between-cluster distance which reflects the shape of the clusters.
- Between-cluster distance measures:
  - **Average linkage**: average of pairwise distances;
  - **Single linkage**: minimum of pairwise distances;
  - **Complete linkage**: maximum of pairwise distances.

More details are given in the lecture *Distances and Expression Measures*.

Divisive methods

- Start with only one cluster.
- At each step, split clusters into two parts.
- Advantages: Obtain the main structure of the data, i.e., focus on upper levels of dendrogram.
- Disadvantages: Computational difficulties when considering all possible divisions into two groups.
- Examples
  - Self-Organizing Tree Algorithm – SOTA (Dopazo & Carazo, 1997);
  - DIvisive ANAlysis – DIANA (Kaufman & Rousseeuw, 1990).

Dendrograms

Dendrograms are often used to visualize the nested sequence of clusters resulting from hierarchical clustering.

While dendrograms are quite appealing because of their apparent ease of interpretation, they can be misleading.

First, the dendrogram corresponding to a given hierarchical clustering is *not unique*, since for each merge one needs to specify which subtree should go on the left and which on the right – there are $2^{(n-1)}$ different dendrograms.

The default in the R function `hclust` (`cluster` package) is to order the subtrees so that the tighter cluster is on the left.

Dendrograms

A second, and perhaps less recognized shortcoming of dendrograms, is that they *impose* structure on the data, instead of *revealing* structure in these data.

Such a representation will be valid only to the extent that the pairwise distances possess the hierarchical structure imposed by the clustering algorithm.

Dendrograms

The **cophenetic correlation coefficient** can be used to measure how well the hierarchical structure from the dendrogram represents the actual distances.

This measure is defined as the correlation between the $n(n-1)/2$ pairwise dissimilarities between observations and their **cophenetic distances** from the dendrogram, i.e., the between cluster dissimilarities at which two observations are first joined together.

The cophenetic distances have a great deal of structure, e.g., there are many ties.

Function `cophenetic` in `mva` package.

Partitioning vs. hierarchical

- **Partitioning**
  - Advantages: Provides clusters that satisfy an optimality criterion (approximately).
  - Disadvantages: Need initial K, long computation time.
- **Hierarchical**
  - Advantages: Fast computation (for agglomerative clustering).
  - Disadvantages: Rigid, cannot correct later for erroneous decisions made earlier.

Estimating the number of clusters

- **Internal indices.** Statistics based on within- and between-clusters matrices of sums-of-squares and cross-products (30 methods reviewed in Milligan & Cooper (1985)). Estimate is the number of clusters K which minimizes or maximizes one of these indices.
- **Average silhouette width.** (Kaufman & Rousseeuw, 1990).
- **Model-based methods.** EM algorithm for Gaussian mixtures, Fraley & Raftery (1998, 2000) and McLachlan et al. (2001).
- **Gap statistic.** (Tibshirani et al., 2001). Resampling method, for each K compare an observed internal index to its expected value under a reference distribution and look for K which maximizes the difference.

MSS

Mean Silhouette Split – MSS. (Pollard & van der Laan, 2002).

Given K clusters, consider each cluster $k = 1, \ldots, K$ separately

- Apply the clustering algorithm to the elements of cluster k.
- Choose the number of child clusters that maximizes the average silhouette width. Call this maximum the **split silhouette**, SS_k.

Define the **mean split silhouette** as a measure of average cluster heterogeneity.

$$MSS(K) = \frac{1}{K} \sum_{k=1}^{K} SS_k.$$

Choose the number of clusters K which minimizes $MSS(K)$.

MSS

- Identifies finer structure in gene expression data. When clustering genes, existing criteria tend to identify global structure only.
- Provides a measure of cluster heterogeneity.
- Computationally easy.

Clest

Clest. (Dudoit & Fridlyand, 2002). Resampling method which estimates the number of clusters based on prediction accuracy.

- For each number of clusters k, repeatedly randomly divide the original dataset into two non-overlapping sets, a learning set $\mathcal{L}^b$ and a test set $\mathcal{T}^b$, $b = 1, \ldots, B$.
  - Apply the clustering algorithm to observations in the learning set $\mathcal{L}^b$.
  - Build a classifier using the class labels from the clustering.
  - Apply the classifier to the test set $\mathcal{T}^b$.
  - Compute a similarity score $s_{k,b}$ comparing the test set class labels from prediction and clustering.

Clest

- The similarity score for k clusters is the median of the B similarity scores: $t_k = \text{median}(s_{k,1}, \cdots, s_{k,B})$.
- The number of clusters K is estimated by comparing the observed similarity score t_k for each k to its expected value under a suitable reference distribution with $K = 1$.

Applies to any partitioning algorithm and any classifier.

Better suited for clustering samples than clustering genes.

Inference

van der Laan & Bryan (2001).

General framework for statistical inference in cluster analysis.

View clustering as a deterministic rule that can be applied to parameters (or estimates thereof) of the distribution of gene expression measures.

Parameters of interest include covariances between the expression measures of different genes.

The parametric bootstrap can be used to study distributional properties (bias, variance) of the clustering results.

Outliers

In classification it has often been found useful to define a class of *outliers*.

This does not seem to have been extended to clustering. However, outlier detection is an important issue since outliers can greatly affect the between-cluster distances.

Simple tests for outliers, such as identifying observations that are responsible for a disproportionate amount of the within-cluster sum-of-squares seems prudent.

Hybrid method – HOPACH

Hierarchical Ordered Partitioning And Collapsing Hybrid – HOPACH (van der Laan & Pollard, 2001)

- Apply a partitioning algorithm iteratively to produce a hierarchical tree of clusters.
- At each node, a cluster is partitioned into two or more smaller clusters. Splits are not restricted to be binary. E.g., choose K based on average silhouette.

Hybrid method – HOPACH

- **Hierarchical.** Can look at clusters at increasing levels of detail.
- **Ordered.** Ordering of the clusters and elements within clusters is data-adaptive and unique, performing better than other hierarchical algorithms. Clustering and ordering are based on the same distance function. The ordering of elements in any level can be used to reorder the data or distance matrices, and visualize the clustering structure.
- **Partitioning.** At each node, a cluster is split into two or more smaller clusters.
- **Collapsing.** Clusters can be collapsed at any level of the tree to join similar clusters and correct for errors made in the partitioning steps.
- **Hybrid.** Combines the strengths of both partitioning and hierarchical clustering methods.

Bagged clustering

Leisch (1999). Hybrid method combining partitioning and hierarchical methods. A partitioning method is applied to bootstrap learning sets and the resulting partitions are combined by performing hierarchical clustering of the cluster centers.

Dudoit & Fridlyand (2002). Apply a partitioning clustering method to bootstrap samples of the learning set. Combine the resulting partitions by (i) voting or (ii) the creation of a new distance matrix. Assess confidence in the clustering results using cluster votes.

Acknowledgments

- **Brown Lab**, Biochemistry, Stanford.
- **Sabina Chiaretti**, Dana Farber Cancer Institute.
- **Jane Fridlyand**, UCSF Cancer Center.
- **Mark van der Laan**, Biostatistics, UC Berkeley.
- **Katie Pollard**, Biostatistics,UC Berkeley.
- **Yee Hwa (Jean) Yang**, Statistics, UC Berkeley.