

# 3 CLUSTER ANALYSIS OF MICROARRAY EXPERIMENTS

Each microarray contains a complete expression profile for a specific sample of mRNA – one scalar value for each of the (roughly) 10,000 genes on the array. The challenge, then, is to develop technologies and an interpretive framework to make sense of this large quantity of data. *Cluster analysis* is a form of multivariate analysis frequently used in the analysis of microarray data. In this section I describe cluster analysis, drawing particular attention to sources of variability and uncertainty in clustering. The inherent uncertainty of cluster analysis is used to motivate the consensus methodology described in section 5.

## 3.1 Introduction to Cluster Analysis

Machine learning can be broken into two paradigms: supervised learning and unsupervised learning. Cluster analysis is a form of unsupervised learning. In supervised learning certain features of the data are known a priori, and this knowledge is used to guide the analysis. Supervised situations often involve discriminant analysis, group classification and class prediction. In unsupervised learning, by contrast, all knowledge and structure must be ‘discovered’ in the data. Unsupervised situations typically involve class discovery and subtype identification. Though cluster analysis is often augmented with supervised forms of analysis (e.g. in data preprocessing and gene selection)[9, 40, 82], as a computational tool it is

inherently unsupervised. The methodology and techniques presented in this paper are designed for completely unsupervised situations.

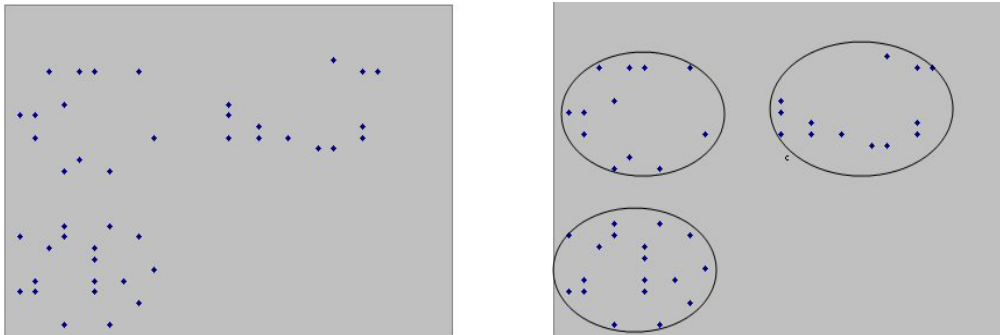
The purpose of cluster analysis is to assign objects to clusters such that objects within a cluster are highly similar, whereas objects in different clusters are highly dissimilar. The resulting clusters *partition* the original objects into non-overlapping sets, such that each of the original objects is a member of exactly one cluster and no cluster contains multiple instances of the same object. Cluster analysis is commonly applied to data mining in a wide variety of fields, including but not limited to information retrieval, signal processing, marketing, socioeconomic research, and classification of single malt whiskeys [97]. In the analysis of microarray data, cluster analysis is one of the most commonly-used analytic tools. In this field, clustering has become so prevalent that Goldstein recently remarked: “It is now commonplace for researchers to perform a hierarchical clustering of microarray data to identify patterns in the clustering. In many instances, cluster analysis is the primary technique of data analysis, regardless of the specific questions of interest.”[39]

In microarray experiments, clustering typically is used for one of the following purposes: (1) To identify samples with similar expression profiles ([4, 6, 40, 42] Schummer 1999), (2) To identify genes with similar expression across samples [10, 22, 31, 92] Chu 1998, Iyer 1999), or (3) Some combination of the two [6, 37, 85, 87]. For purposes of clarity, in this paper I deal primarily with (1), though all of the techniques in principle apply to the others as well.

When using cluster analysis to identify samples with similar expression levels, each *sample* (or equivalently, each microarray) is represented as a vector of

expression levels. Thus, given  $m$  samples with  $n$  expression levels, the data can be represented as  $m$  vectors in  $n$ -dimensional space. Computationally, input data is stored as an  $m \times n$  matrix  $X$  of expression data, where each row corresponds to a sample and each column represents a gene. The goal of clustering is to group similar samples into clusters based on expression levels across  $n$  dimensions. Note that  $m$  and  $n$  are general features of cluster analysis corresponding to points and dimensions - for the sake of clarity I will henceforth refer to points as samples and dimensions as genes.

Ideally, each cluster will contain samples that are similar to each other and dissimilar to samples in other clusters (Figure 3.1). In rough terms, good clusters will be crisp and compact, and geometrically separated from other clusters. *Cluster validity analysis*, briefly discussed in section 4.1, formalizes these ideas.



**Figure 3.1:** *Visualization of clustering process.* **a)** Input data set consists of 50 points (samples) in 2-dimensional space ( $m=50, n=2$ ). **b)** Cluster analysis reveals 3 natural clusters in the dataset.

The next section presents two clustering algorithms; the notation in Table 3.1 will be used there and throughout this thesis.

	Description
$m$	number of samples/patients/vectors
$n$	number of genes/dimensions
$x_i$	the $i^{\text{th}}$ sample, $1 \leq i \leq m$
$X$	the set of all samples $\{x_1, x_2, \dots, x_m\}$ ; equivalently, the $m \times n$ expression matrix
$x_{i,g}$	the expression level of the $g^{\text{th}}$ gene of $i^{\text{th}}$ sample, $1 \leq g \leq n$
$\bar{x}_i$	the average expression level of $x_i$ : $\bar{x}_i = \frac{1}{n} \sum_{g=1}^n x_{i,g}$
$d(a,b)$	The distance between (dissimilarity of) vectors a and b
$C_p$	the $p^{\text{th}}$ cluster of samples
$C_{p,i}$	the expression vector of the $i^{\text{th}}$ sample of $p^{\text{th}}$ cluster
$D(C_p, C_q)$	The distance between (dissimilarity of) $C_p$ and $C_q$
$\bar{C}_p$	the centroid of $C_p$ : $\bar{C}_p = \sum_{x \in C_p} \frac{x}{ C_p }$

**Table 3.1:** Notation used in this thesis

### 3.2 Clustering Algorithms

A *clustering algorithm* is an algorithm by which cluster analysis is accomplished. Though many different clustering algorithms exist, the forms most commonly applied to microarray data [78] are *hierarchical algorithms* and *partitioning algorithms*. There are subclasses of both of these types, but the most representative forms, *agglomerative hierarchical* and *k-means*, are presented below. For a discussion of other clustering algorithms, including *grid-based*, *density-based* and *model-based* algorithms, the reader is referred to [57], [12] and [99], or Appendix A for a short discussion.

### 3.2.1 Hierarchical Algorithms – Agglomerative Hierarchical Clustering

Hierarchical algorithms are the most common type of clustering algorithm used in microarray analysis. Seminal experiments using this type of clustering include [31], [26] and [76]. These algorithms generate a tree-like clustering hierarchy of samples. Unlike many forms of cluster analysis (including the partitioning algorithms discussed below), hierarchical algorithms are deterministic, though the solution is potentially non-unique[73]. Each sample on the tree (referred to as a *dendrogram*) is a leaf; the length of the branch connecting samples corresponds to the distance between the two (Figure 3.2). The basic algorithm requires  $O(m^2 \log m)$  time and  $O(m^2)$  space[63], and consists of three steps:

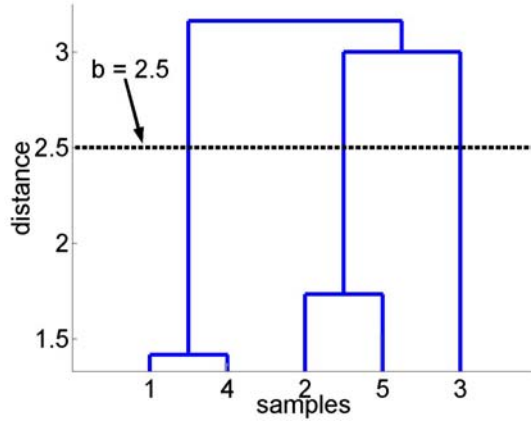
- (1) Calculate dissimilarity matrix based on pairwise dissimilarities

$$\mathbf{D}(i,j) = \{d(x_i, x_j) \mid 1 \leq i < j \leq m\} \quad (3.1)$$

There are thus  $\frac{m(m-1)}{2}$  values, as the major diagonal is filled with zeros, since  $d(x_i, x_i) = 0$  for each  $1 \leq i \leq m$ , and the values below the major diagonal are redundant. The sample-sample distance  $d$  is calculated based on a distance function (e.g. Euclidean distance, Minkowski metric) or other dissimilarity functions (e.g. Pearson correlation, cosine distance).

(2) *Iteratively merge most similar clusters.* Initially, there are  $m$  singleton clusters, such that  $C_p = \{x_i\}$  for each  $1 \leq i = p \leq m$ . The two most similar clusters  $C_p$  and  $C_q$ , satisfying  $\mathbf{D}(p,q)=\min(\mathbf{D})$ , are then merged to form a new cluster joined by a branch of length  $\mathbf{D}(p,q)$ . After this first join  $\mathbf{D}$  must be updated to account for the fact that two clusters have been merged (i.e. there are now fewer than  $m$  clusters, and at least one cluster is no longer a singleton cluster).  $\mathbf{D}$  is updated using a *linkage function*  $D$  that measures the distance between clusters (e.g. nearest neighbor distance or centroid-centroid distance). The entire process is iterated  $m-1$  times until all clusters are linked and a complete dendrogram is generated.

(3) *Divide dendrogram into distinct clusters.* The last phase of hierarchical clustering is to divide the dendrogram into distinct clusters. Typically, a breakpoint  $b$  is chosen that represents a maximum distance permitted to exist between clusters. For instance, in Figure 3.2, the breakpoint  $b = 2.5$  identifies the following three clusters  $\{1,4\}$ ,  $\{2,5\}$  and  $\{3\}$ , while the value  $b=1.7$  produces four clusters  $\{1,4\}$ ,  $\{2\}$ ,  $\{5\}$  and  $\{3\}$ . Note that  $b$  can be chosen to produce any user-defined number of clusters  $k$ , as long as  $2 \leq k \leq m$ . Choosing  $b = 2.5$  in Figure 3.2 is equivalent to choosing  $k=3$ .



**Figure 3.2:** Simple hierarchical clustering dendrogram. Samples 1 and 4 are closely related, as are samples 2 and 5. The centroid of cluster  $\{1,4\}$  is quite dissimilar to that of cluster  $\{2,5,3\}$ . The breakpoint  $b = 2.5$  divides the dendrogram into three clusters:  $C_1 = \{1,4\}$ ,  $C_2 = \{2,5\}$ ,  $C_3 = \{3\}$ .

### 3.2.2 Partitioning Algorithms - K-Means

$K$ -means clustering is used nearly as often as hierarchical clustering (see, for example, [86, 90, 100]). As in hierarchical clustering, partitioning algorithms divide data into groups; however, partitioning algorithms are more direct. Rather than producing a dendrogram that must later be cut at a breakpoint,  $k$ -means immediately divides the data into  $k$  subsets (clusters), and then updates the clusters until they are ‘good,’ as defined by equation (3.3) below.

More precisely, each cluster  $C_p$ ,  $1 \leq p \leq k$ , has a *centroid*  $\bar{C}_p$  that is the  $n$ -dimensional mean of all samples in  $C_p$ :

$$\bar{C}_p = \sum_{x \in C_p} \frac{x}{|C_p|} \quad (3.2)$$

The  $k$ -means algorithm attempts to minimize

$$d_{tot} = \sum_{x_i \in X} d(x_i, \mathbf{c}(x_i)) \quad (3.3)$$

where  $\mathbf{c}(x_i)$  is the centroid of the cluster containing the  $i^{\text{th}}$  sample and  $d(x_i, \mathbf{c}(x_i))$  is the distance between the  $i^{\text{th}}$  sample and the centroid. In other words,  $k$ -means searches for an assignment of samples to clusters that minimizes the total sum of sample-to-centroid distances, summed over all  $m$  samples (or equivalently, over all  $k$  clusters). The algorithm proceeds as follows:

- (1) *Choose  $k$  seed points.* These points are typically selected randomly from the entire set of samples, although other techniques exist[1].
- (2) *Assign samples to clusters.* Each sample  $x_i$  is assigned to the nearest cluster  $\mathbf{c}(x_i)$ , as defined by a distance metric  $d$  (typically one of the Minkowski metrics). Initially, the  $k$  seed points are used as the centroids.
- (3) *Recalculate all cluster centroids.* At each iteration, the cluster memberships will change, so it is necessary to recompute the centroids based on equation (3.2).
- (4) *Repeat steps (2) and (3) until convergence, or up to a maximum number of iterations.* Convergence is achieved when samples cease to be reassigned, or when  $d_{tot}$  ceases to decrease.

Eventually, the  $k$ -means algorithm will converge to a local minimum of  $d_{tot}$ , which, in general, is *not* a global minimum. The problem of computing the global minimum solution is NP-hard[60] and can only be achieved through exhaustive repetition of the algorithm. However, a single run of the algorithm can be optimized to run in time proportional to  $O(m)$ , though most implementations depend on  $k$  and other factors.[21]

### 3.3 Major Sources of Variability in Clustering

A more complete discussion of these and other sources of variability is contained in Appendix A. Here, I briefly summarize the most important factors.

#### 3.3.1 Choice of Algorithm

Whereas hierarchical algorithms create a clustering hierarchy in which samples are related in a tree-like structure, partitioning algorithms divide the data into absolute subsets that are completely unrelated. Hierarchical algorithms proceed deterministically, often yielding clusters of strange shapes and sizes;  $k$ -means bounces around between local minima, generally producing “spherical” clusters of similar sizes. Though both eventually partition the set of samples, the logic behind each partitioning is fundamentally different. The reader is referred to section A.1 for a more thorough comparison of these algorithms.

#### 3.3.2 Choice of Parameters

Given a clustering algorithm, there are still many decisions to make. For instance, in  $k$ -means clustering it is necessary to specify a distance function  $d$ . Figure 3.1 shows a picture of 2-dimensional Euclidean space, but  $k$ -means typically is run in many-dimensional spaces that aren't necessarily Euclidean. The choice of  $d$  determines the structure of the metric space for the original  $m$  samples (see Figure A.2). In hierarchical clustering,  $d$  is similarly used to generate the

dissimilarity matrix  $\mathbf{D}$ . Additionally, hierarchical clustering utilizes a linkage function  $D$  to measure distance between clusters. Just as  $d$  determines the metric space of samples,  $D$  determines the metric space of clusters. Other than the distance metric, major parameters include the value and placement of the  $k$  seeds ( $k$ -means), and the placement of the breakpoint  $b$  (hierarchical).<sup>1</sup> Various distance metrics,  $k$ -values, and other parameters are addressed in more detail in section A.2, and will not be discussed any further in the text, though section 4 assumes familiarity with the concepts.

### 3.3.3 Choice of Input Data

Getz [37] recently observed, “the main difficulty is that each biological process on which we wish to focus may involve a relatively small subset of the genes; the large majority of those present on the microarray constitute a noisy background that may mask the effect of the small subset.” Each sample’s expression profile contains tens of thousands of genes that could potentially be used as the basis for the expression vector in clustering; section A.3 gives strong biological, statistical, and computational reasons *not* to use the entire set of  $n$  genes. These reasons, which are echoed throughout the literature, motivate the use of  $n_{\text{used}}$  genes, where  $n_{\text{used}} < n$ . For this reason, a large number of experiments use filtering criteria to select the ‘most relevant’ genes. However, it is unclear how to choose these genes, and even how many to choose. To the author’s knowledge, no work has been done to systematically examine the effects of using a particular subset of genes as the basis for each of the

---

<sup>11</sup> Henceforth I will simply refer to the number of clusters as  $k$ , noting that  $b$  maps directly to  $k$ , and vice versa (though technically the former is many-one and the latter is one-many).

$n_{\text{used}}$ -dimensional vectors (cf. [47]; the dearth of literature on this subject has also been noted in [39]). As can be imagined, the definition of ‘most relevant’ varies from source to source. Some authors use high variances [84], others use high average expression across samples [15, 76] or genes that satisfy maximum/minimum expression criteria[6], or  $n$ -fold change from median [76], etc.

### 3.4 Summary

Cluster analysis is commonly used in microarray experiments, but there is a large amount of uncertainty in the clustering process. Though it has repeatedly been demonstrated that a large portion of clusters are preserved across clustering methods [65, 69, 70, 75], it is also widely acknowledged that different methods produce different clusters. As has been demonstrated here (in Appendix A) and in the literature, the major sources of variability are: (1) the choice of clustering algorithm [7, 39, 90], (2) the parameters to the algorithm [18, 19, 46]; and (3) the subset of input data [37, 39, 47]. Each unique combination of these three factors produces a potentially unique set of clusters. The ‘unique combination’ will henceforth be referred to as a *clustering configuration*.

**Definition 3.1:** *Clustering Configuration*

A unique combination of:

- (1) Clustering algorithm
- (2) Algorithm parameters
- (3) Collection of genes used

The set of clusters produced by a particular clustering configuration is a *partition* of the original  $m$  samples – each sample appears in exactly one cluster.

**Definition 3.2: Partition**

A set of clusters  $\Pi = \{C_1, \dots, C_k \mid C_p \subseteq X\}$  is a *partition* of  $X$  if and only if  $\bigcup_{p=1}^k C_p = X$  and  $C_p \cap C_q = \emptyset$  for  $p \neq q$ .