Fundamentals of Data Mining in Genomics and Proteomics.
Edited by Werner Dubitzky, Martin Granzow, and Daniel Berrar.
Springer, Pp 123-148

6

Clustering

Class Discovery in the Post-Genomic Era

Joaquín Dopazo

Department of Bioinformatics, Centro de Investigación Príncipe Felipe, E46013, Valencia, Spain
jdopazo@cipf.es

6.1 Introduction

From a historical perspective we can distinguish an initial period in the DNA microarray technology in which almost all publications were related to reproducibility and sensitivity issues. Thus, many classical microarray papers dating from the late nineties were simple proof-of-principle experiments (Eisen et al., 1998; Perou, et al., 1999), in which only cluster analysis was applied in order to check whether differences at gene expression level could reproduce macroscopic observations. Later, specificity became a main concern as a natural reaction against quite liberal interpretations of microarray experiments made by some researchers, such as the fold change criterion to select differentially expressed genes. It soon became obvious that genome-scale experiments need to be carefully analyzed, because many apparent associations happened merely by chance when large amounts of data were studied (Ge et al., 2003). In this context, different methods for the adjustment of p-values, which are considered standard today, started to be extensively used (Benjamini et al., 2001; Storey et al., 2003) . More recently, the use of microarrays for building predictive models of clinical outcomes (van't Veer et al., 2002), albeit not being free of criticisms (Simon, 2005), fuelled the use of the technology because of its practical implications. There are still some concerns with the cross-platform coherence of results, but it seems clear that intra-platform reproducibility is high (Moreau et al., 2003), and, although the overlap between the lists of genes differentially expressed among platforms was low, the enrichment in biologically relevant labels emerging from these lists was consistent (Bammler, et al., 2005). This fact clearly points to the importance of the interpretation of experiments in terms of their biological implications instead of restricting them to a mere comparison of lists of gene identifiers (Al-Shahrour et al., 2005b; Al-Shahrour, et al., 2005c).

Despite the fact that clustering is one of the most popular methodologies and the first one in being used in the field of microarray data analysis (Quackenbush, 2001; Slonim, 2002), it has often been improperly used (Simon

et al., 2003). The literature on DNA microarrays provides numerous examples for the inadequate use of clustering for tackling problems of class comparison. Although cluster analysis is appropriate for class discovery, it tends to be inefficient for class comparison or class prediction. An important caveat when analysing DNA microarray experiments is that, although these are not based on gene-specific mechanistic hypotheses, they must be designed with clear objectives. Three typical types of objectives are *class comparison*, *class prediction* and *class discovery* (Golub et al., 1999). *Clustering*, also known as unsupervised analysis, belongs to this last category because no previous information about the class structure of the data set is used in the study. Cluster analysis makes reference to an extensive set of methods for partitioning samples into groups on the basis of their respective differences, referred to as distances (D'Haeseleer, 2005). Usually, the distance measures are computed with regard to the complete set of genes represented on the array. Clustering can be done on the experiments (based on all the genes) or on the genes (across all the experiments). Although the methods used can be exactly the same, a note of caption must be introduced here because it is not uncommon that a given class of experiments (disease, molecular subtype, etc.) is distinguished by a relatively small number of genes, whose effect may end up being diluted by the irrelevant genes. To circumvent this problem, there exists a family of clustering methods, generically known as *biclustering*, in which the aim is to find groups of genes with coordinated expression only across a subset of experimental conditions (Cheng et al., 2000; Lazzeroni et al., 2002; Tanay et al., 2002; Sheng et al., 2003).

There are other types of data that deserve particular attention: Time series or dose-response data. In this case, clustering of experiments is meaningless because there are sequential data and one is typically interested in clustering genes across all the time (or dosage) points. Recently, time series are gaining importance because the experimental methods for synchronising cell cultures are becoming more accurate, constituting nowadays a 30% of the total number of DNA microarray experiments published (Simon et al., 2005). While typical microarray assays are designed to study static experimental conditions, in time series a temporal process is measured. Time series offer the possibility of identifying the dynamics of gene activation, which might allow to infer causal relationships. An important difference between these two types of experiments is that, while static data from a sample population (e.g., diseased cases, healthy controls, etc.) are assumed to be independent, time series data are characterized by displaying a strong autocorrelation between successive points (Bar-Joseph, 2004). Initially, time series were analyzed using methods originally developed for independent data points (Spellman et al., 1998; Zhu, et al., 2000). More recently, algorithms were developed to specifically address this type of data. Different clustering methods specially designed for time series data have been recently proposed. Among these, clustering based on the dynamics of the expression patterns (Ramoni et al., 2002), clustering using a

hidden Markov model (Schliep et al., 2003), and clustering specifically devised for short time series (Ernst et al., 2005) can be cited.

Once the clustering has been performed the following questions arise: Is the partition obtained relevant? Is there a "better" partition involving more or less clusters or a different distribution of the items within the clusters? Since most of the clustering algorithms do not include any type of measure of the reliability of the clusters obtained, these questions have to be addressed a posteriori. There are different criteria to estimate the quality of the clustering obtained (Kerr et al., 2001; Azuaje, 2002; Dudoit et al., 2003; Handl et al., 2005) and some programs (e.g., the CAAT in GEPAS (Montaner et al., 2006)) offer the possibility of obtaining cluster quality indexes. Given that some methods require that the number of clusters is predefined, (e.g., k-means or self-organizing maps), the exact determination of the number of clusters in the context of microarray data is a major concern, which has been specifically addressed by different authors (Horimoto et al., 2001; Dudoit, et al., 2002; Bolshakova et al., 2006).

But, why should we expect to find groups of co-expressed genes or a class structure in our experiments? Genes do not operate alone in the cell, but in a sophisticated network of interactions that we only recently start to decipher (Rual et al., 2005; Stelzl et al., 2005; Hallikas et al., 2006). It has been a long recognised fact that co-expressed genes tend to play some common roles in the cell (Stuart et al., 2003; Lee et al., 2004). Ultimately, it is this common functionality that we aim to understand when we face a clustering problem. Thus, an important and non-negligible last step of any clustering analysis (and, in general, of any DNA microarray experiment) is the *functional interpretation* (Al-Shahrour and Dopazo, 2005b). There are a number of tools specially designed to search for significant enrichment of biological terms – usually gene ontology terms (Ashburner et al., 2000), but others can be used – in sets of genes (Khatri et al., 2005). Typically, one set of genes is tested against the rest of genes in the array. This set of genes can be, more precisely, a cluster of co-expressed genes (Al-Shahrour, et al., 2004), and the result produced accounts for the functional roles played by the genes in the cluster. There are different tools that allow to easily link results of clustering methods to algorithms for functional annotation, such as the GEPAS (Herrero et al., 2003; Herrero et al., 2004; Vaquerizas et al., 2005; Montaner et al., 2006).

Recently, biological annotations (e.g., GO, KEGG pathways, etc.) have been used for cluster validation (Bolshakova et al., 2005) and, even more importantly, biological information (Huang et al., 2006; Pan, 2006) or phenotypic information (Jia et al., 2005) have been used as a constitutive part of clustering algorithms.

Clusters can be obtained in numerous different ways. There are many distinct algorithms for measuring distances among genes and many procedures for partitioning the data. In addition, most of the clustering methods do not provide any measurement of the reliability of the results obtained. This apparent diversity of ways for approaching the same problem, together with the

lack of information on the reliability of the results obtained has attracted over the clustering an undeserved reputation of subjective analysis strategy. Understanding the basis of the distance metrics and the partitioning procedures and being aware of their limitations will provide the fundaments for a proper and reasonable class discovery analysis.

6.2 Basic Concepts

Despite the large number of clustering methods and the new methods proposed in the field of DNA microarray data analysis (Heyer, et al., 1999; Hastie et al., 2000; Yeung et al., 2001a; De Smet, et al., 2002), only a subset of them have been used with some regularity in this context. Among other merits, the reason for the popularity of many methods of microarray data analysis, and clustering is not an exception, resides in its availability in standard software packages. Among the most commonly used methods we can cite hierarchical clustering (Eisen et al., 1998), k-means (McQueen, 1967), *self-organizing maps* (SOMs) (Kohonen, 1997) or *self-orgainzing tree algorithm* (SOTA) (Herrero et al., 2001). Implicitly or explicitly, clustering methods depend on distances between objects. Different ways of computing distances account for different biological properties of the data. In this section I will review different distance metrics, distinct clustering algorithms, different ways of estimating cluster quality and algorithms for the functional annotation of clustering results.

6.2.1 Distance Metrics

In a widely accepted standard representation, microarray experiments are two-dimensional matrices of gene expression values in which columns correspond to genes and rows to experiments. Thus, the identification of genes with coordinated expression across the experiments or, alternatively, the identification of groups of experiments with similar expression values for all the genes is achieved through the comparison of the column or row vectors, respectively, by means of a distance function. The choice of such distance function depends on the biological property that the researcher considers. There are two types of distances extensively used in the comparison of expression profiles: *Euclidean distance* and *Pearson coefficient of correlation*.

Euclidean distance is obtained as the square root of the summation of the squares of the differences between all pairs of corresponding gene expression values (rows or columns). Euclidean distance computes the geometric distance between two points in an n-dimensional space (n being the size of the vectors – row or column – involved in the comparison). Thus, pairs of genes (or experiments) whose components display similar magnitude of expression are considered similar by this distance.

Although this property may be useful in some cases, it seems more relevant, from a biological point of view, to search for genes (or experiments),

whose expression profiles display a similar overall trend, irrespective of their absolute values. The Pearson correlation coefficient (r) measures this property. It provides values between -1 (negative correlation) and 1 (positive correlation). The more the two expression profiles display the same trend, the closer to 1 is the r-value. This measure of similarity in the shapes of two profiles, while not taking the magnitude of the profiles into account, suits well the biological intuition of coexpression (Eisen et al., 1998). Euclidean distance can be used for obtaining correlations if the data are properly transformed (standardized, that is, subtracting the mean and dividing by the variance). Then the Euclidean distance between two points x and y relates to correlation as $(x - y)^2 = 2(1 - |r|)$ (Alon et al., 1999).

Most of the distances found in the microarray-related literature are derived from the Euclidean distance or from the correlation coefficient. Also some non-parametrical distances have been applied, such as the *Spearman rank correlation* (Kotlyar et al., 2002) or *jackknifed correlation coefficient* (Heyer et al., 1999). (More distance metrics can be found in Chapter 7, Table 7.2).

However, there are other different scenarios beyond the simple coexpression whose exploration is of much interest from a biological point of view. A very interesting property of the correlation coefficient is that it can be used to detect negatively correlated expression profiles. The study of such negative correlations can be very useful for identifying control processes that antagonistically regulate downstream pathways.

6.2.2 Clustering Methods

According to the final representation of the results, data can be clustered in two different ways: In a hierarchical or in a non-hierarchical manner. Hierarchical clustering allows detecting higher-order relationships between clusters of profiles whereas most of the non-hierarchical classification techniques allocate profiles into a predefined number of clusters, without any assumption on the inter-cluster relationships (see Figure 6.1a). Many authors prefer hierarchical clustering because it allows to explore the entire hierarchy of relationships at different levels. There are distinct clustering methods based on different ways of aggregating data, which use (implicitly or explicitly) different distance functions. Without the aim of producing an exhaustive enumeration of them, here I will briefly review some of the most commonly used and most relevant clustering methods. In a quick review of 1157 papers found in Pubmed using "cluster and microarray" as keywords, I have found that 74% used hierarchical clustering, 15% used k-means, 6% used SOM, 2% used SOTA, another 2% used model-based clustering, and in the remaining cases other alternative methods were used. Although these figures can change depending on the keywords used for finding the papers, they give an approximate idea on the relative actual usage of each procedure.

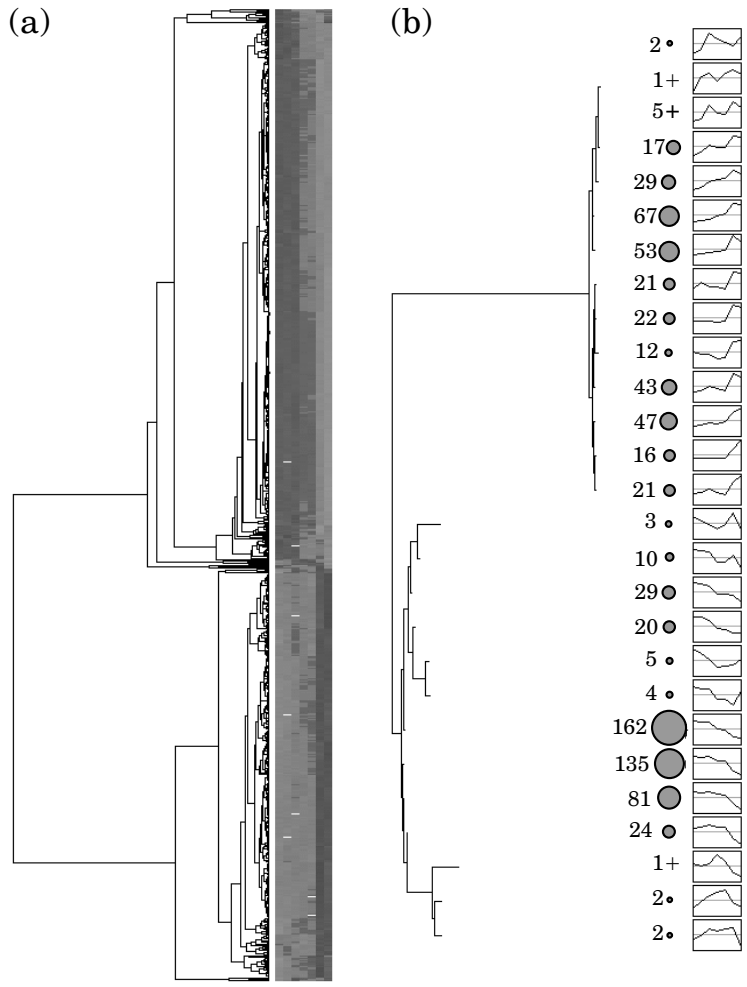

Fig. 6.1. Different clustering methods applied to cluster genes. a) Aggregative hierarchical clustering, and b) SOTA with default parameters.

6.2.2.1 Aggregative Hierarchical Clustering

Aggregative hierarchical clustering (Eisen et al., 1998) is one of the preferred choices for the analysis of patterns of gene expression (Quackenbush, 2001; D'Haeseleer, 2005). Standard aggregative hierarchical clustering produces a representation of the data with the topology of a binary tree, in which the most similar patterns are clustered in a hierarchy of nested subsets (Sneath, et al., 1973). Figure 6.1a shows a typical output of produced by the method. In aggregative hierarchical clustering, each vector (gene or experiment) is ini-

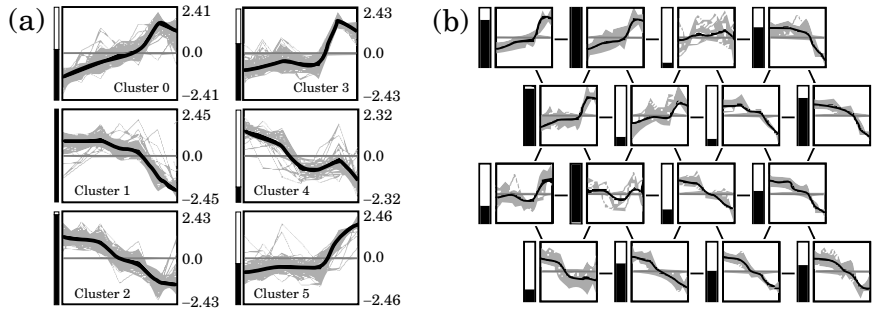

Fig. 6.2. Different clustering methods applied to cluster genes. a) k-means with $k = 6$, and (b) SOM with a 4×4 output map with hexagonal neighbourhood.

tially assigned to a single cluster; at each step, the distance between every pair of clusters is calculated and the pair of clusters with the smallest distance is merged; the procedure is iteratively carried on until all the data are grouped into a single cluster. Depending on the way in which vectors are merged into a cluster and the distance of the new cluster to the rest of items (also known as linkage distance) is calculated, different variants of the method can be distinguished. This linkage distance can be calculated as the shortest distance of any of the two joined members (*single linkage*), the largest distance (*complete linkage*) or either weighted or unweighted averages (*average linkage*).

After the full tree is obtained, the determination of the final partition is achieved by "cutting" the tree at a certain level or height, which is equivalent to putting a threshold on the pairwise distance between clusters. Note that the decision of the final partition is thus rather arbitrary.

6.2.2.2 *k*-Means

The k-means algorithm (McQueen, 1967) requires the specification of the number of clusters, k, into which the objects are going to be partitioned. Then the mean vector for each of the k clusters, the *seed*, is initialized either by direct assignment (e.g., from the input) or by random generation (random initial seeds). Then, the algorithm proceeds through an iterative procedure, consisting of the following two steps: (1) Using the given mean vectors, the algorithm assigns each gene (or experiment) to the cluster with the closest mean vector. (2) The algorithm recalculates the mean vectors (which are the sample means) for all the clusters. The iterative procedure ends when all the mean vectors of the clusters remain constant or do not change significantly. Figure 6.2a shows the output of the algorithm in a data set using $k = 6$.

6.2.2.3 Self-Organizing Maps

Self-organizing maps (SOM) (Kohonen, 1997) are a technique to visualize the high-dimensional input data (in our case, the gene expression data) onto an (usually two-dimensional) output map of prototype vectors (also called neurons) by a process known as *self-organization*. Similarly to k-means, the dimension of the output map needs to be specified by the user. After initializing the prototype vectors, the algorithm iteratively performs the following steps. (1) Every input vector is associated with the closest prototype vector of the output map, (2) the components of the prototype vector (and with less intensity the prototype vectors in the neighbourhood) are updated according to a weighted sum of all the input vectors that are assigned to it. This process is repeated until the prototype vectors of the output node converged to a constant value. During the clustering process the prototype vectors are pulled towards the regions of the space that are more densely populated by the input vectors. Figure 6.2b show a typical output of SOM using an output map of 4×4 with an hexagonal neighbourhood.

6.2.2.4 Self-Organizing Tree Algorithm

The textitself-organizing tree algorithm (SOTA) (Dopazo et al., 1997; Herrero et al., 2001) is a different type of self-organizing neural network based on the SOM, but implementing a binary tree topology, instead of the classical two-dimensional grid, and a different strategy of training. The iterative procedure, with the application of the self-organization principle to the prototype vectors, is similar to the case of SOM. The differences reside in the fact that the unique prototype vectors directly updated are the leaves of the tree structure. The neighbourhood is defined through the tree topology. After convergence of the network, the prototype vector containing the most variable population of expression profiles (variation is defined here by the maximal distance between two profiles that are associated with the same prototype vector) is split into two sister vectors (causing the binary tree to grow), hereafter the entire process is restarted. The algorithm stops (i.e., the tree stops growing) when a threshold of variability is reached for each prototype vector. Hence, the number of clusters does not need to be specified in advance. The determination of the threshold of variability involves the actual construction of a randomised data set. In contrast to hierarchical clustering, which is an aggregative method, the SOTA is divisive. Figure 6.1b shows an example of clustering of genes obtained with SOTA.

6.2.2.5 Model-Based Clustering

Although *model-based clustering* has already been used in the past in other fields, its application to microarray data is relatively recent (Yeung et al., 2001a; Ghosh et al., 2002; McLachlan et al., 2002). In contrast to the clustering

methods described so far, model-based methods provide a consistent statistical framework for obtaining data partitions. The basic assumption in model-based clustering is that the data are generated by a mixture of a finite number of underlying probability distributions, where each distribution represents one cluster. Model-based clustering methods face the problem of associating every gene (or experiment) with the best underlying distribution in the mixture, and at the same time, finding out the parameters for each of these distributions. Different approximations can be used to infer these parameters. Gaussian mixture models have been applied with success to microarray data clustering (Yeung et al., 2001a). On the other hand, problems such as the estimation of the number of clusters can be solved in a more efficient way using a Bayesian framework (Vogl et al., 2005).

6.2.3 Biclustering

As previously mentioned, biclustering methods search for genes with coordinated expression across a subset of experiments. While in the beginning clustering algorithms were applied to both genes and experiments of the microarray matrix to reorganize data and thus visualize patterns common to genes and experiments (Alon et al., 1999; Getz et al., 2000), soon algorithms specifically designed for biclustering, such as the *Samba*, methods based on graph theory (Tanay et al., 2002), the *iterative signature algorithm* (ISA) (Ihmels et al., 2002) or mixtures of normal distributions (Lazzeroni and Owen, 2002) were proposed. Recently, model-based algorithms providing a more rigorous statistical framework have been proposed (Barash et al., 2002; Sheng et al., 2003).

6.2.4 Validation Methods

As previously mentioned, validation of the relevance of the cluster results is of paramount importance given that most clustering methods do not provide any clue on reliability. Validation can be based on either external or internal criteria. In the first case some gold standard is chosen and its agreement with the partition obtained by the clustering method is taken as a support for such a partition. Usually, biological information (gene ontology, pathways, etc.) is used for this purpose (Tavazoie et al., 1999; Toronen, 2004; Al-Shahrour and Dopazo, 2005b). Internal criteria for statistical cluster validation imply the assessment of cluster coherence using different measures that compare inter- to intra-cluster variability, such as *silhouette coefficient* (Rousseeuw, 1987), *Dunn-like indices* (Azuaje, 2002), *connectedness* or *separation measures* (Handl et al., 2005). Another internal criterion consists of testing the stability or the robustness of a cluster result when noise is deliberately added to the data (Kerr and Churchill, 2001; Dudoit and Fridlyand, 2002).

6.2.5 Functional Annotation

Clustering of microarray data produces a collection of objects (genes or experiments) based on the comparison of their expression profiles but gives no information on the functional basis for this grouping. While not much effort has been developed on the way of understanding the molecular functional basis of clustering of experiments, there are however numerous papers dealing with the issue in the case of clustering of genes. Ending up with a mere list of genes of interest is only half-way to the result of a microarray experiment. Apart from the utility that functional annotation can have as an external criterion for cluster quality, it constitutes itself an unavoidable final step of any microarray analysis. The proper interpretation of cluster analysis of microarray experiments is usually performed in two steps: In a first step, clusters of genes of interest are selected, and then the enrichment of any type of biologically relevant annotation for these genes is compared to the corresponding distribution of this annotation in the background (typically, the rest of genes). It is important to note that this comparison to the background is essential because sometimes apparent high enrichment in a given annotation is nothing but a reflect of a high proportion of this particular term in the whole genome and, consequently, has nothing to do with the set of genes of interest. There are different available tools, such as FatiGO (Al-Shahrour, et al., 2004) and others (Khatri and Draghici, 2005), that estimate significant enrichment in different functionally relevant annotation terms such as GO (Ashburner et al., 2000), KEGG pathways (Kanehisa, et al., 2004), etc.

6.3 Advantages and Disadvantages

The methods and algorithms previously described have been developed for situations and under assumptions that are not always fulfilled by DNA microarray data. In this section I will comment some of the positive and negative features of the methods in the light of some of the most common problems in clustering.

- **Finding the proper number of clusters**. In general, clustering methods do not define the proper number of clusters by themselves. k-means and SOM need the pre-specification of the number of clusters. Different strategies are used to circumvent this problem, but commonly different runs of the program with different values of k (in k-means) need to be evaluated with a quality cluster index to decide about the optimal number of clusters. Nevertheless, this strategy is finally computationally expensive. A similar problem affects some model-based procedures. In this case the algorithm has to compare multiple log maximum likelihood values to optimize the complexity of the model (Yeung et al., 2001a), or resampling the data set (Yeung et al., 2001b). Both strategies are very time-consuming.

On the other hand, model-based methods based on a Bayesian approach can estimate the partition with the proper number of clusters (although also at the expense of high run times). Besides, the SOTA method (Herrero et al., 2001) implements a quick permutation-based strategy that produces the partition at which the clusters contain elements with significant intra-cluster distances (that is, distances that cannot be found in random clusters).

- **Reliability of the clusters obtained**. As mentioned above, the reliability of clustering methods can be checked in different ways, based on external information or on internal properties of the partition obtained. There are several benchmarking studies that compare the relative efficiencies of different clustering methods in defining partitions in both artificial data sets and in well-known real data sets (Gibbons et al., 2002; Datta et al., 2003; D'Haeseleer, 2005; Handl et al., 2005). As general conclusion, hierarchical clustering with single linkage would not be a good choice, because of its poor performance (Gibbons and Roth, 2002; D'Haeseleer, 2005). Depending on the study, hierarchical clustering with complete or average linkage results in different performances: Sometimes one of the linkage strategies seems to work better than the alternative and sometimes not. In general, k-means, SOM and SOTA seem to exhibit a better performance than hierarchical clustering (Gibbons and Roth, 2002; D'Haeseleer, 2005; Handl et al., 2005) according to different indexes such as silhouette, Dunn, etc. It is important to note here that the performances reported for k-means and SOM refer to an unrealistic situation in which the number of clusters is provided to the method. This information is currently unknown in real scenarios. Unfortunately, there are no benchmarking studies that include model-based clustering methods to date, and only a few performance comparisons are available. Thus, for example, model-based Bayesian methods seem to perform better than k-means, even in situations in which the number of clusters (k) was provided to the method (Vogl et al., 2005).
- **Reliability of biclustering**. An interesting, although not exhaustive, comparative study has recently been published (Prelic et al., 2006). Here, the Samba (Tanay et al., 2002) and ISA (Ihmels et al., 2002) methods seem to work reasonably well in the absence of noise, and a method proposed by the authors, *Bimax*, seems to outperform them in noisy situations.
- **Run times**. Despite the advantages of model-based clustering methods (they can estimate the reliability of the partition and some versions can estimate the number of clusters and impute missing values), their extremely long run times (hours to days) and usually its requirement of powerful computers for running, represent a limitation to its application to real situations. As a general rule, methods that use pair-wise distance matrices (e.g., hierarchical clustering or k-means) have run times that are, at least, quadratic on the number of items, while methods based on the distances of the items to a number of clusters (e.g., SOM or SOTA) have almost linear run times. Nevertheless, a data set with a number of features ranging from

20,000 to 40,000 and a number of experiments ranging from 20 to 100 can be a matter of seconds for SOM and SOTA, a few minutes for hierarchical clustering and no more than 15 minutes for k-means.

- **Interpretation of the results**. As mentioned above, the functional annotation of the partition obtained can be used as an external criterion to check the quality of the clustering obtained, but, at the same time it is crucial for obtaining a proper annotation of the results obtained. Functional annotation of clusters implies searching for enrichment of some functional terms (typically GO, KEGG pathways, etc.) in them. One important consideration in this step is the correction for multiple testing. For example, there are around 14,000 GO terms; the possibility of finding apparent enrichments in a few GO terms just by chance is high. To avoid obtaining a considerable number of false positive enrichments different methods for multiple testing adjustments can be used. Beyond the classical Bonferroni or Holm's corrections, which are extremely conservative, one of the most popular choices are the *false discovery rate* (FDR), which in addition accounts for dependencies between the data (Benjamini et al., 1995; Benjamini and Yekutieli, 2001). One of the first programs to incorporate this correction was *FatiGO* (Al-Shahrour et al., 2004), although now it is included in a number of systems (Onto-Express, GOStat, GOToolBox, Gosurfer, etc.) Despite the importance of applying such corrections, there are still programs, such as *GoMiner*, *DAVID*, *eGOn*, *GOTM* or *CLENCH* that do not include it yet (Khatri and Draghici, 2005).

6.4 Caveats and Pitfalls

It is worth noting that many clustering methods produce partitions even with random data. This is commonly known as the "garbage-in- garbage-out" effect in programming and points to the necessity of having some criteria in the application of these methods. There are two potential weak points in any clustering analysis: The distance function used and the algorithm for producing the partition. The combined effect of both choices (sometimes restricted by the clustering method) and the properties of the particular data set at hand will make one of the methods more efficient compared to the alternatives. Benchmarking studies, albeit not perfect, give an idea of the relative performance of the different methods under different conditions, especially when some of the conclusions are consistent across different, independent studies. In the previous section some considerations have been made on the different methods and a common conclusion was the poor performance of hierarchical clustering when single linkage was used. While hierarchical clustering with average or complete linkage seems to work well, SOM, SOTA and k-means seem to be superior according to internal indexes (Silhouette, Dunn, and other) or external criteria (enrichment of functional terms). Model-based methods (in particular Bayesian approaches) seem to show a superior performance, al-

though runtimes are still excessive as to be considered feasible alternatives on many computers. Beyond the advantages and disadvantages commented in the previous section some considerations follow that deserve to be made.

6.4.1 On Distances

Usually, the distance metrics are computed with regard to the complete set of genes represented on the array. Clustering can be done on the experiments (based on all the genes) or on the genes (across all the experiments). It is not uncommon that a given class of experiments (diseases, molecular subtypes, etc.) is distinguished by a relatively few number of genes, whose effect may end up diluted among the contributions of the rest of genes. This can lead to the construction of groups based on irrelevant features unrelated to the aim of the study. And this effect represents an even greater problem when working with systems that cannot be under a strict experimental control, i.e., patients or samples directly collected from nature. In a classical paper, only two types of diffuse large B-cell lymphoma could clearly be defined while some subtypes were merged together in clusters not reflecting the clinical subtype composition of the disease (Alizadeh et al., 2000). The only way described so far to overcome this problem is via a biclustering approach. Similarly, typical distances used in microarray assume that all the vector components used in the computation are independent and this assumption clearly does not hold in the case of time series, where all the experiments are autocorrelated. In this case clustering methods specifically designed for time series should be used (Ramoni et al., 2002; Schliep et al., 2003). Moreover, microarray time series are short in comparison with typical time series in other disciplines (about 80% of microarray time series experiments involve only three to eight time points (Ernst et al., 2006)), so clustering methods specifically developed for this purpose should be used (Ernst et al., 2005).

6.4.2 On Clustering Methods

A significant problem associated with k-means or SOM algorithms is the arbitrary choice of the number of clusters, since this information is commonly not available in a real class-discovery problem. In practice, this makes it necessary to use a trial-and-error approach where a comparison and validation of several runs of the algorithm with different parameter settings are necessary. A similar problem affects some versions of model-based methods, such as Gaussian mixture models (Yeung et al., 2001a), but the strategies for finding the number of clusters here (Yeung, et al., 2001a; Yeung et al., 2001b) are enormously time-consuming.

Another parameter that will influence the result of k-means clustering is the choice of the seeds. The algorithm is hampered by the problem of local minima. This means that with different seeds, the algorithm can yield different result. This problem also applies to SOM although to a lesser extent.

Another inherent problem in SOM is that the training of the network (and, consequently, the definition of clusters) depends on the number of items assigned to each cluster. If irrelevant data (e.g., invariant, "flat" profiles) or some particular type of profile is over-represented in the data, SOM will produce an output in which this type of data will populate the vast majority of clusters. As a consequence, the most interesting profiles may appear in a few clusters and the resolution obtained for them is poorer.

In contrast to SOM, the number of nodes does not need to be initialized in SOTA. The partition obtained with SOTA is proportional to the heterogeneity of the data, but not to the number of items in each cluster. Thus, SOTA is quite insensitive to perturbing effects of big clusters on the global cluster structure and can simultaneously resolve small and big clusters. Since SOTA is a divisive method, a test can easily be coupled to the growing tree process to decide at which point the growing of the tree should be stopped because all the significant clusters have been found (Herrero et al., 2001).

6.5 Alternatives

Clustering has been extensively used over many years for different purposes and consequently many clustering methods are available (Sneath and Sokal, 1973), so an exhaustive description of alternatives falls beyond the scope of this chapter. In this chapter the clustering methods most commonly used in the field of microarray data analysis have been described. Nevertheless, other proposals have been made that, despite their potential, have not been extensively used yet.

Early from an historical perspective in microarray data analysis, the QT-Clust method was introduced (Heyer et al., 1999). This method considers each expression profile in the data and determines how many of them are within the distance specified as quality guarantee. The candidate cluster with the largest number of expression profiles is selected as the output of the algorithm. Then, the expression profiles of the selected cluster are removed, and the whole procedure starts again to find the next cluster. The algorithm stops when the number of profiles in the largest remaining cluster falls below a pre-specified threshold.

Adaptive quality-based clustering (De Smet et al., 2002) uses a heuristic two-step approach to find one cluster at a time. In the first step, a quality-based approach is performed to locate a cluster centre in the area where the density (i.e., the number) of gene expression profiles has a local maximum. In the second step, the algorithm re-estimates the quality (i.e., the radius) of the cluster so that the genes belonging to the cluster are, in a statistical sense, significantly co-expressed. The cluster found is subsequently removed from the data and the whole procedure is restarted. Only clusters whose size exceeds a predefined number are reported in the output.

Contrarily to aggregative hierarchical clustering, the divisive version of this method provides a picture of the tree from lower to higher resolution, as the construction of the tree proceeds. Apart from SOTA (Herrero et al., 2001), other divisive hierarchical methods, e.g., based on the maximum entropy principle (Alon et al., 1999), have been proposed. The algorithm tries to find the most likely partition of data into sets and subsets, creating in this way a binary tree structure.

Fuzzy versions of some clustering methods have also been applied to microarray data analysis (Dembele et al., 2003). The rationale behind the proposal of the use of fuzzy methods is the difficulty of defining cluster boundaries (Spellman et al., 1998). Fuzzy membership of genes should then be considered more an operative procedure than a reality. Difficulties in the placement of a gene in a cluster are due to noise and multifunctionality and can be best addressed through biclustering methods.

Furthermore, other types of distances can be mentioned. There are distances that can deal with datasets containing large numbers of measures that have a high degree of internal correlations. Correlations between experiments or genes tend to produce elliptical clusters, which cause problems to methods whose optimal performance occurs with compact, spherical clusters, such as k-means. Distances that take into account covariance between experiments, like the Mahalanobis distance (Mahalanobis, 1936), may be useful for datasets with high internal correlation. The problems that originate from the complex joint distribution of gene expression values, particularly their structure of internal correlations and non-normality, have been addressed by other researchers (Hunter et al., 2001), who argue that simple similarity metrics such as Euclidean distance or correlation similarity are suboptimal in microarray datasets and propose the use of Bayesian approaches.

6.6 Case Study

Understanding the molecular roles played by potentially relevant genes in a given experiment is still one of the most interesting objectives in many microarray experiments. One of the most popular hypotheses in microarray data analysis is that coexpression of genes across a series of experiments is most probably explained through some common functional role (Eisen et al., 1998). Actually, this causal relationship has been used to predict gene function from patterns of co-expression (Stuart et al., 2003; Lee et al., 2004).

In this case study I use data from a genome-wide study to search for the factors responsible for the transcription of the cluster of co-ordinately expressed ribosomal proteins of Saccharomyces cerevisiae (Rudra et al., 2005). This dataset is publicly available at `http://gepas.bioinfo.cipf.es/cgi-bin/datasets`. There is a step that must be taken prior to any sort of microarray data analysis: *Normalization* of the data. This step is beyond the scope of this

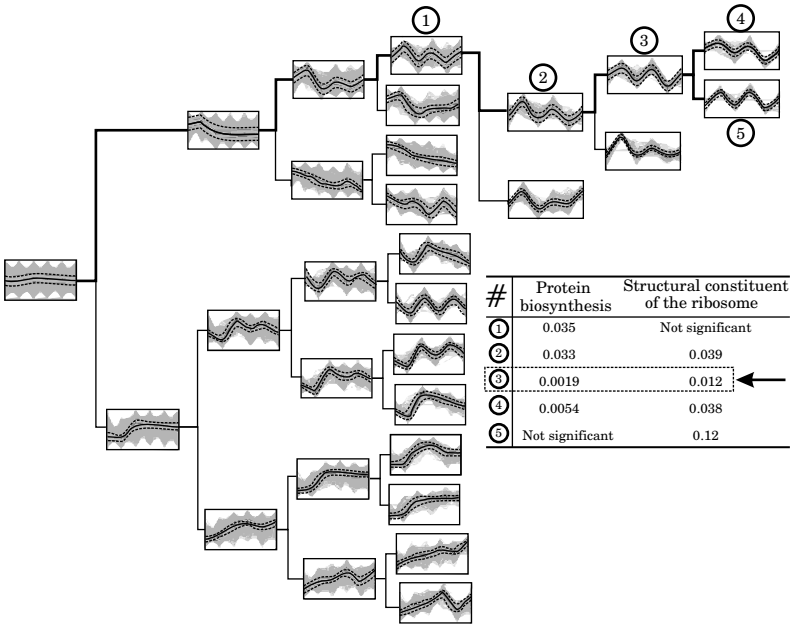

Fig. 6.3. Clustering of gene expression profiles obtained with the SOTA method (setting the variability threshold to 80%) and represented using the CAAT tool (Montaner et al., 2006). The summarized description of the tree is obtained with the CAAT tool, with the representation of the gene expression profiles (individual gene profiles in grey and average profile in black) assigned to clusters and sub-clusters. The upper branch is developed until no more partitions are produced by the SOTA algorithm. Note how the confidence intervals for the average gene expression profile become narrower as we move towards the terminal nodes. The arrow marks the level at which the enrichment in the biological terms studied has a maximum significance.

chapter and aims to remove all the variability due to experimental manipulation and unrelated to the actual experiment. (See Chapter 3 for details on normalizing microarray data.) It can be carried out by using standard programs such as the DNMAD (Vaquerizas et al., 2004), SNOMAD (Colantuoni et al., 2002) or other programs. With the goal of finding groups of genes that co-express across the experiments, gene expression patterns were clustered using the SOTA algorithm (Herrero et al., 2001) as implemented in the GEPAS (`http://www.gepas.org`) suite of web tools (Herrero et al., 2003; Herrero et al., 2004; Vaquerizas et al., 2004; Montaner et al., 2006). Figure 6.3 shows a general view of the SOTA hierarchical tree obtained, where the top branch is shown in detail, and the terminal nodes (clusters as defined by SOTA) are at the right end. The CAAT tool allows selecting clusters and automatically submitting them to the *FatiGOplus* tool (Al-Shahrour et al., 2004; Al-Shahrour, et al., 2006) for functional analysis. When the upper terminal node is chosen,

we found the GO terms "Protein biosynthesis" (FDR-adjusted $p = 0.0054$) and "Structural constituent of the ribosome" (FDR-adjusted $p = 0.0038$) significantly over-represented in the group of co-expressing genes contained in the cluster, when compared to the rest of the tree. This operation can be repeated for all the nodes of the tree and most of them will display significant over-representation of GO terms. And, what is even more interesting, we can examine internal nodes. If the internal nodes are sequentially analyzed along the branch of a tree for the enrichment in biologically relevant terms it is possible to find a level in the tree in which this enrichment is maximum (and significant). In Figure 6.3, this level in the tree that maximizes the proportion of genes annotated as "protein biosynthesis" and "structural constituent of the ribosome" is marked by an arrow. Actually, it is the parent of the level at which SOTA decides to stop growing. As we move from higher levels to lower levels of the hierarchy we find clusters with tighter co-expression, which are more likely involved in a common function. At this point, clustering based on the distance measure has a natural, functional meaning. Beyond this point, new partitions will not reflect a functional (biologically relevant) co-expression (see the two last clusters in which the p-value increases or, in some cases is non significant). Functional annotation can be considered an external cluster quality measure.

6.7 Lessons Learned

The first and most important lesson is that clustering is for class discovery (unsupervised analysis), but not for class discrimination or class prediction (supervised analysis). Although this may sound obvious, there is still an extensive misuse of these techniques (Simon et al., 2003). Clustering of genes and experiments can be carried out using exactly the same methods (applied to columns or to rows, respectively). The final partition obtained is based on equal contributions of each experiment (when clustering genes) or each gene (when clustering experiments). It is worth remembering that many genes will only introduce noise (because they represent physiological conditions or any particularity of the sample, unrelated to the biological trait we have in mind) and consequently, the partition obtained could be irrelevant from a biological point of view. Not all the experiments are equivalent in terms of their analysis. In addition to the inherent noise there are situations, such as time series or dose-response experiments, in which the data display a high internal correlation. These cases should be clustered with methods specifically designed for them (Bar-Joseph, 2004). Finding a partition requires the correct estimation of the number of clusters and its reliability. Only a few methods include these features . Among them I can cite SOTA (Herrero et al., 2001) or recent versions of model-based clustering (Vogl et al., 2005), although the latter is too demanding in computational resources to constitute an alternative. Most clustering methods require external strategies to find the optimal number of

clusters and their reliability. This is nothing that should prevent one from using a particular clustering method but must be taken into account. Just trying with several k values for k-means and choosing the partition which "looks nicer" can be a interesting exploratory exercise, but is definitively not a proper way of obtaining a partition. Contrarily, irrespective of the final decision on the clustering method, SOTA, given its reliability (according internal indexes and external criteria, see (Handl et al., 2005)) and speed, constitutes a good choice for a first exploration of the data. With respect to the performance of the different methods, recent comparative studies (Gibbons and Roth, 2002; Datta and Datta, 2003; D'Haeseleer, 2005; Handl et al., 2005) suggest that hierarchical clustering (with complete or average linkage), SOM and k-means (if the number of clusters is known) and SOTA tend to produce accurate partitions according to several cluster quality indexes. And last but not least, clusters of co-expressing genes represent biological processed cooperatively carried out by the genes. A proper understanding of these processes require of the application of methods that examine the biological roles jointly carried out by the genes, that is, the functional annotation of the experiment (Al-Shahrour and Dopazo, 2005b; Khatri and Draghici, 2005).

6.8 List of Tools and Resources

There are different tools available and several repositories containing tools for the analysis of microarray data. The list below does not intend to be an exhaustive catalogue of these resources but contains some of the most complete and stable ones.

6.8.1 General Resources

- `http://www.nslij-genetics.org/microarray/soft.html`.
- `http://ihome.cuhk.edu.hk/`.
- `http://bioinformatics.ubc.ca/resources/`.

6.8.1.1 Multiple Purpose Tools (Including Clustering)

- GEPAS: A web-based resource for microarray gene expression data analysis (Herrero et al., 2003; Herrero et al., 2004; Vaquerizas et al., 2005; Montaner et al., 2006) which beyond clustering offers many more tools (normalisation, gene selection, predictors, functional annotation, Array-CGH, etc.) `http://www.gepas.org`.
- INCLUSive: A web portal for clustering and regulatory sequence analysis (Coessens et al., 2003) `http://www.esat.kuleuven.ac.be/inclusive`.
- Expression Profiler is a web based platform for microarray data analysis developed at the EBI. (Kapushesky et al., 2004). `http://www.ebi.ac.uk/expressionprofiler`.

6.8.2 Clustering Tools

- `http://homes.esat.kuleuven.be/~thijs/Work/Clustering.html`: Adaptive Quality-Based Clustering (De Smet et al., 2002).
- `http://www.ii.uib.no/~bjarted/jexpress/index.html`: J-EXPRESS: University of Bergen, Norway.
- `http://rana.lbl.gov/EisenSoftware.htm`: CLUSTER, TREEVIEW: Eisen's lab at Lawrence Berkeley National Laboratory.
- `http://www.genome.wi.mit.edu/MPR/software.html`: GENE-CLUSTER: Whitehead Institute.
- `http://gepas.bioinfo.cipf.es/cgi-bin/sotarray`: SOTA (Herrero et al., 2001): CIPF, Spain. Also included in GEPAS.

6.8.3 Biclustering Tools

There are not many biclustering tools available yet. The coupled two-way analysis (Getz et al., 2000) is a simple method available at `http://ctwc.weizmann.ac.il/`. Also, GEMS (Wu et al., 2005) is a nice example of a web-based tool for biclustering (available at `http://genomics10.bu.edu/terrence/gems/`).

6.8.4 Time Series

Time series (and dose-response) experiments are characterized by displaying a strong autocorrelation between successive points (Bar-Joseph, 2004) and must, consequently, be analysed with algorithms that specifically take into account this fact. The algorithm STEM has been, in addition, designed for short time series and can be found at `http://www.cs.cmu.edu/~jernst/stem`.

6.8.5 Public-Domain Statistical Packages and Other Tools

Probably, the most popular resource for microarray data analysis is *bioconductor* (Gentleman et al., 2004). It is written in the popular R statistical programming language and offers many modules for the analysis of microarray data. It is available at `http://www.bioconductor.org`. The BRB tools, developed by the Richard Simon and Amy Peng Lam group, offer a variety of useful algorithms. Available at: `http://linus.nci.nih.gov/BRB-ArrayTools.html`. Additionally, there are packages in Java, which are very popular, as is the case of MEV (`http://www.tigr.org/software/microarray.shtml`) (Saeed et al., 2003). Java packages provide an interactive and convenient interface and can run on multiple platforms, constituting an interesting alternative to web-based tools, which cannot offer the same degree of interactivity. The only limitation comes from the characteristics of the local computer in which the program is installed (which can be an obstacle in a non-negligible number of cases).

6.8.6 Functional Analysis Tools

- *Babelomics* (Al-Shahrour et al., 2005c; Al-Shahrour et al., 2006) is a suite of web tools for the functional annotation and analysis of groups of genes in high throughput experiments. Tools include: *FatiGO* (Al-Shahrour et al., 2004), *FatiGOplus*, *Fatiscan* (Al-Shahrour et al., 2005a), *Gene Set Enrichment Analysis* (GSEA) (Subramanian et al., 2005), *Marmite*, and the *Tissues Mining Tool* (TMT). `http://www.babelomics.org`.
- *goCluster* simultaneously implements annotation information, clustering algorithms and visualization tools for microarray data analysis (Wrobel et al., 2005). Available at: `http://www.bioconductor.org`; `http://www.bioz.unibas.ch/gocluster`.

6.9 Conclusions

Clustering is essential for finding either (functionally related) co-expressed genes or subtypes of experiments based on their gene expression profiles. Although clustering of genes and experiments can be carried out using exactly the same methods, the final result obtained is based on equal contributions of each data component. Thus, it is worth noting that in the case of clustering of experiments many genes will only introduce noise and consequently the resulting partition can be meaningless from a biological point of view. In addition to noise, some experiments are conceptually different. Time series or dose-response experiments, for example, are characterized by the existence of a high internal correlation between consecutive experiments. These experiments must be clustered with methods specifically designed for them (Bar-Joseph, 2004). Regarding the comparative performances of the methods, hierarchical clustering (except in the case of single linkage), SOM and k-means (provided the number of clusters is known) and SOTA seem to produce reliable partitions (Gibbons and Roth, 2002; Datta and Datta, 2003; D'Haeseleer, 2005; Handl et al., 2005). Finally, methods that examine the enrichment in biologically relevant terms (Al-Shahrour and Dopazo, 2005b; Khatri and Draghici, 2005) are necessary for a proper understanding of the biological processes cooperatively carried out by the genes present in co-expression clusters .

Acknowledgements

This work is supported by grants from MEC BIO2005-01078, NRC Canada-SEPOCT Spain and Fundación Genoma España.