10

Advances in Cluster Analysis of Microarray Data

Qizheng Sheng, Yves Moreau, Frank De Smet, Kathleen Marchal and Bart De Moor

Abstract

Clustering genes into biological meaningful groups according to their pattern of expression is a main technique of microarray data analysis, based on the assumption that similarity in gene expression implies some form of regulatory or functional similarity. We give an overview of various clustering techniques, including conventional clustering methods (such as hierarchical clustering, *k*-means clustering and self-organizing maps), as well as several clustering methods specifically developed for gene expression analysis.

Keywords

microarray, clustering, biclustering

10.1 Introduction

The first question in microarray data analysis is to identify genes whose expression levels are significantly changed under different experimental conditions. Basic statistical techniques can solve this problem efficiently (Baldi and Brunak, 2001). However, such an analysis treats the genes separately rather than exploring their relation with each other. For a gene, the detailed relations between the levels of expression in the different conditions are neglected in this first-level analysis. Based on the assumption that expressional similarity (i.e. coexpression) implies some kind

Data Analysis and Visualization in Genomics and Proteomics Edited by Francisco Azuaje and Joaquin Dopazo
© 2005 John Wiley & Sons, Ltd. ISBN: 0-470-09439-7

of regulatory or functional similarity of the genes (and vice versa), the challenge of finding genes that might be involved in the same biological process is thus transformed into the problem of clustering genes into groups based on their similarity in expression profiles.

The first generation of clustering algorithms applied to gene expression profiles (e.g. hierarchical clustering (Eisen *et al.*, 1998), *k*-means (Hartigan, 1975) and self-organizing maps (SOM; Kohonen, 1995) were mostly developed outside biological research. Although encouraging results have been produced (Spellman *et al.*, 1998; Tavazoie *et al.*, 1999; Tamayo *et al.*, 1999), some of their characteristics (such as determination of the number of clusters, clustering of outliers and computational complexity) often complicate their use for clustering expression data (Sherlock, 2000).

For this reason, a second generation of clustering algorithms has started to tackle some of the limitations of the earlier methods. These algorithms include, among others, model-based algorithms (Yeungo *et al.*, 2001; McLachlan, Bean and Peel, 2000), the self-organizing tree algorithm (Herrero, Valencia, and Dopazo, 2001), quality-based algorithms (Heyer, Kruglyak and Yooseph, 1999; De Smet *et al.*, 2002), and biclustering algorithms (Cheng and Church, 2000; Sheng, Moreau and De Moor, 2003). Also, some procedures have been developed to help biologists estimate some of the parameters needed for the first generation of algorithms, such as the number of clusters present in the data (Lukashin and Fuchs, 2000; Yeungo *et al.*, 2001).

While it is impossible to give an exclusive survey of all the clustering algorithms that have been developed for gene expression data, we try here to illustrate some key issues. The selection of algorithms is based on their popularity, their ability to handle the specific characteristics of microarray data and inevitably some personal biases. This chapter is organized as follows.

In Section 10.2, we address a few common issues for the discussion of clustering algorithms. In particular, we first discuss the preprocessing of microarray data, which is needed to overcome some difficult artifacts before clustering. Then, we address the basic but necessary ideas of the orientation of clustering (clustering genes versus. clustering experiments) and the distance metrics commonly used to compare gene expression profiles.

We discuss the application of classical clustering algorithms to microarray data in Sections 10.3–10.5, where hierarchical clustering, *k*-means clustering, and self-organization maps are respectively addressed. Then, in Section 10.6, we identify common drawbacks of the first-generation clustering algorithms and give a wish list of some desirable features that an ideal clustering algorithm should carry.

Next, we look at some second-generation clustering algorithms, such as the self-organizing tree algorithm (SOTA; Herrero, Valencia and Dopazo, 2001) in Section 10.7, the quality-based clustering algorithms (Heyer, Kruglyak and Yooseph, 1999; De Smet *et al.*, 2002) in Section 10.8, mixture models for microarray data (Yeungo *et al.*, 2001; McLachlan, Bean and Peel, 2002) in Section 10.9, and biclustering algorithms (Sheng, Moreau and De Moor, 2003) in Section 10.10.

Changes in details such as the preprocessing procedures, the algorithm or even the distance metrics might lead to different clustering results. Thus, in Section 10.11, we discuss methods used to validate clustering results.

10.2 Some Preliminaries

Before going into clustering algorithms *per se*, there are a few issues worth recalling.

Preprocessing microarray data

A correct preprocessing strategy, which not only removes as for as possible the systematic noise present in microarray data but also provides a basis for the comparison between genes, is truly essential to an effective cluster analysis (in accordance with the '*garbage in, garbage out*' principle). Common procedures for preprocessing include the following five steps (Moreau *et al.*, 2002).

1. *Normalization*. First, it is necessary to normalize hybridization intensities within a single experiment or across experiments by computing and removing the biases to correct the data, before one can compare the results from different microarray experiments (Quackenbush, 2001).

2. *Nonlinear transformation*. Expression ratios (e.g. from two-channel cDNA microarray experiments using a test and reference sample) are not symmetrical in the sense that upregulated genes have expression ratios between one and infinity, while downregulated genes have expression ratios squashed between one and zero (Quackenbush, 2001). Taking the logarithms of these expression ratios results in symmetry between expression values of up- and downregulated genes. Furthermore, the noise on a microarray measurement is multiplicative as a function of the intensity of the signal. Taking the logarithm of the expression values makes noise approximately additive, except for low-intensity signals. The generalized log transformation combines normalization and transformation to provide this property over the whole signal range (Durbin and Rocke, 2004).

3. *Missing value replacement*. Microarray experiments often contain missing values that need to be replaced for many cluster algorithms. Techniques of missing value replacement (e.g. using the *k*-nearest-neighbour method or the singular value decomposition, SVD) have been described (Troyanskaya *et al.*, 2001), taking advantage of the rich information provided by the expression patterns of other genes in the data set.

4. *Filtering*. For any microarray study, many genes do not contribute to the underlying biological progress and show little variation over the different experiments.

These genes will have seemingly random and meaningless profiles after standardization (see further). Another problem arises from the highly unreliable expression profiles containing many missing values. The quality of the cluster would significantly degrade if these data were passed to the clustering algorithms as such. Filtering removes such expression profiles typically by putting a minimum threshold for the standard deviation of the expression values in a profile and a maximum threshold on the percentage of missing values (Eisen *et al.*, 1998).

5. *Standardization or rescaling*. Biologists are mainly interested in grouping gene expression profiles that have the same relative behaviour, i.e. genes that are up- and downregulated together. Genes showing the same relative behaviour but with diverging absolute behaviour (e.g. gene expression profiles with a different baseline or a different amplitude but going up and down at the same time) will have a relatively high Euclidean distance (see Section 10.2.3). Cluster algorithms based on this distance measure will therefore wrongfully assign the genes to different clusters. This effect can largely be prevented by applying standardization or rescaling to the gene expression profiles so that they have zero mean and unit standard deviation.

Clustering genes versus clustering experiments

Instead of clustering genes, we can also cluster experimental conditions, where the task is to find groups of experimental conditions (which can be, for example, tumour samples) across which all the genes behave similarly. This type of clustering can be helpful for problems such as the discovery of histopathological tumours. While most of the discussion will be oriented towards clustering genes, most of it can be applied *mutatis mutandis* to clustering conditions.

Distance metrics

Depending on the way we define a cluster, clustering methods can be divided into two types – model-based clustering methods and distance-based clustering methods. Model-based clustering algorithms assume that the data points in the high-dimensional space are generated by a mixture of probabilistic models with different parameters. Each of these models is thus defined as a cluster. We will talk about this type of clustering method in detail in Section 10.9.

Distance-based clustering methods (to which most of the classical clustering methods belong, such as hierarchical clustering, *k*-means and SOM), in contrast, cluster data points according to some function of their pairwise distances. Some common distance metrics for clustering microarray data are the following.

1. *Pearson correlation*. The Pearson correlation *r* is the dot product of two normalized vectors, or in other words, the cosine between two vectors. It measures the

similarity in the shapes of two profiles, while not taking the magnitude of the profiles into account, and therefore suits well the biological intuition of coexpression (Eisen *et al.*, 1998).

2. *Squared Pearson correlation*. This is the square of the Pearson correlation, which considers two vectors pointing in the exact opposite directions to be perfectly similar (i.e., in this case, $r = -1$ while $r^2 = 1$), which might also be interesting for biologists (because repression is a form of coexpression).

3. *Euclidean distance*. Euclidean distance measures the length of the straight line connecting the two points. It measures the similarity between the absolute behaviours of genes, while the biologists are more interested in their relative behaviours. Thus, a standardization procedure is needed before clustering using Euclidean distance. Importantly, after standardization, the Euclidean distance between two points x and y is related to the Pearson correlation by $|x - y|^2 = 2(1 - |r|)$ (Alon *et al.*, 1999).

4. *Jackknife correlation*. The jackknife correlation (Heyer, Kruglyak and Yooseph, 1999) is an improvement for the Pearson correlation (which is not robust to outliers). Jackknife correlation increases the robustness to single outliers by computing a collection of all the possible leave-one-(experiment-)out Pearson correlations between two genes and then selecting the minimum of the collection as the final measure for the correlation.

10.3 Hierarchical Clustering

The first introduction of hierarchical clustering to the world of biology was its application to the construction of phylogenetic trees. Early applications of the method to gene expression data analysis (Eisen *et al.*, 1998; Spellman *et al.*, 1998) have proved its usefulness.

Hierarchical clustering has almost become the *de facto* standard for gene expression data analysis, probably because of its intuitive presentation of the clustering results. The whole clustering process is presented as a tree called a dendrogram; the original data are often reorganized in a heat map demonstrating the relationships between genes or conditions.

In hierarchical (agglomerative) clustering (Eisen *et al.*, 1998), each expression profile is initially assigned as one cluster; at each step, the distance between every pair of clusters is calculated and the pair of clusters with the minimum distance is merged; the procedure is carried on iteratively until a single cluster is assembled.

After the full tree is obtained, the determination of the final clusters 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 cluster is thus rather arbitrary.

Distance measure between two clusters

As we mentioned, in every step of agglomerative clustering, the two clusters that are closest to each other will be merged. Here comes the problem of how we define the distance between two clusters. There are four common options:

1. *Single linkage*. The distance between two clusters is the distance between the two closest data points in these clusters (each point taken from a different cluster).

2. *Complete linkage*. The distance between two clusters is the distance between the two furthest data points in these clusters.

3. *Average linkage*. Both single linkage and complete linkage are sensitive to outliers (Duda, Hart, and Stork, 2001). Average linkage provides an improvement by defining the distance between two clusters as the average of the distances between all pairs of points in the two clusters.

4. *Ward's method.* At each step of agglomerative clustering, instead of merging the two clusters that minimize the pairwise distance between clusters, Ward's method (Ward, 1963) merges the two clusters that minimize the 'information loss' for the step. The 'information loss' is measured by the change in the sum of squared error of the clusters before and after the merge. In this way, Ward's method assesses the quality of the merged cluster at each step of the agglomerative procedure.

These methods yield similar results if the data consist of compact and well separated clusters. However, if some of the clusters are close to each other or if the data have a dispersed nature, the results can be quite different (Duda, Hart, and Stork, 2001). Ward's method, although less well known, often produces the most satisfactory results.

Visualization of the results

A heat map presenting the gene expression data, with a dendrogram to its side indicating the relationship between genes (or experimental conditions), is the standard way to visualize the result of hierarchical cluster analysis on microrray data. The length of a branch in the dendrogram is proportional to the pairwise distance between the clusters. Importantly, the leaves of the dendrogram, and accordingly the rows of the heat map, can be swapped (without actually changing the information contained in the tree) so that the similarity between adjacent genes is maximized, and hence the patterns embedded in the data become obvious in the heat map. However, the time complexity of such an optimal organization of the dendrogram is $O(2^{N-1})$ (because for each of the $N-1$ merging steps there are two possible orders to arrange the concerned clusters). Yet, the structure of the dendrogram remains an important problem, because although the dendrogram itself does not determine the clusters for the users, a good ordering of the leaves can help the users to identify and interpret the clusters. A heuristic approach aiming to find a good solution was developed (Eisen *et al.*, 1998) by weighting genes using combined source of information, and then placing the genes with lower average weight earlier in the final ordering. Further, Bar-Joseph, Gifford and Jaakkola (2001) reported a dynamic programming method that helps to reduce the time and memory complexities for solving the optimal leaf-ordering problem.

10.4 *k*-Means Clustering

k-means clustering (Hartigan, 1975) is a simple and widely used partitioning method for data analysis. Tavazoie *et al.* (1999) provided an example for applying *k*-means clustering to microarray data.

The number of clusters *k* in the data is needed as an input for the algorithm. The algorithm then initializes the mean vector for each of the *k* clusters either by hard assignment (e.g. from the input, or by random generation). These initial mean vectors are called the seeds. Next, the *k*-means algorithm proceeds iteratively with the following two steps: (1) using the given mean vectors, the algorithm assigns each gene (or experiment) to the cluster represented by the closest mean vector; (2) the algorithm recalculates the mean vectors (which are the sample means) for all the clusters. The iterative procedure converges when all the mean vectors of the clusters remain stationary.

A significant problem associated with the *k*-means algorithm is the arbitrariness of predefining the number of clusters, since it is difficult to predict the number of clusters in advance. In practice, this implies the use of a trial-and-error approach where a comparison and biological validations of several runs of the algorithm with different parameter settings are necessary (Moreau *et al.*, 2002). Another parameter that will influence the result of *k*-means clustering is the choice of the seeds. The algorithm suffers from the problem of converging to local minima. This means that with different seeds the algorithm can yield very different result.

10.5 Self-Organizing Maps

SOM (Kohonen, 1995) is a technique to visualize the high-dimensional input data (in our case, the gene expression data) on an output map of neurons, which are sometimes also called nodes. The map is often presented in a two-dimensional grid (usually of hexagonal or rectangular geometry) of neurons. In the high-dimensional input space, the structure of the data is represented by prototype vectors (serving similar functions as the mean vectors in the *k*-means algorithm), each of which is related to a neuron in the output space.

As an input for the algorithm, the dimension of the output map (e.g. a map of 6 × 5 neurons) needs to be specified. After initializing the prototype vectors, the algorithm iteratively performs the following steps. (1) Every input vector (e.g. representing a gene expression profile) is associated with the closest prototype vector, and thus is also associated with the corresponding neuron on the output space. (2) The coordinates of a prototype vector are updated based on a weighted sum of all the input vectors that are assigned to it. The weight is given by the neighbourhood function applied in the output space. As a result, a prototype vector is pulled more towards input vectors that are closer to the prototype vector itself and is less influenced by the input vectors located further away. In the meantime, this adaption procedure of the prototype vectors is reflected on the output nodes – nodes associated with similar prototype vectors are pulled closer together on the output map. (3) The initial variance of the neighbourhood function is chosen so that the neighbourhood covers all the neurons, but then the variance decreases during every iteration so as to achieve a smoother mapping. The algorithm terminates when convergence of the prototype vectors is achieved or after completing a pre-defined number of training iterations.

Because of the advantage in visualization, choosing the geometry of the output map is not as crucial a problem as the choice of the number of clusters for a *k*-means method. Like the *k*-means method, the initial choice of prototype vectors remains a problem that influences the final clustering result of SOM clustering. A good way to seed the prototype vectors is to use the result from a principal component analysis (PCA) (Kohonen, 1995).

The usefulness of SOM on clustering microarray data is illustrated by Tamayo *et al.* (1999).

10.6 A Wish List for Clustering Algorithms

The limitations of the first-generation algorithms together with the specific characteristics of gene expression data call out for clustering methods tailored for microarray data analysis. Collecting the lessons from the first-generation algorithms and the demands defined by the specific characteristics of microarray data, we compose here a subjective wish list of the features of an ideal clustering method for gene expression data.

A problem shared by the first-generation algorithms is the decision on the number of clusters in the data. In *k*-means clustering and SOM clustering, this decision has to be made before the algorithms are executed, while in hierarchical clustering it is postponed until the full dendrogram is formed, where the problem then is to determine where to cut the tree.

Another problem of the first-generation algorithms is that they all assign every gene in the data set (even outliers) to a particular cluster. A proper filtering step in the preprocessing (see Section 10.2.1) helps to reduce the number of outliers, but is insufficient. Therefore, a clustering algorithm should be able to identify genes that are not relevant for any clusters and leave them as they are.

A third problem is robustness. For all the three clustering techniques addressed above, difference in the choice of distance metrics (either for the vectors or for the clusters) will result in different final clusters. In *k*-means clustering and SOM clustering, the choices of seeds for the mean vectors or the prototype vectors also greatly influences the result. Taking into account the noisy nature of microarray data, improving the robustness should be one of the goals when designing novel clustering algorithms for gene expression data.

A fourth problem is the high dimensionality of microarray data, which requires the clustering algorithm to be fast and not memory hungry (a major problem of hierarchical clustering where the full distance matrix should be computed).

Finally, the biological process under study in a microarray experiment is a complicated process, where genes interact with each other in different pathways. Consequently, a gene under study might be directly or indirectly involved in several pathways. With this idea in mind, clustering algorithms that allow a gene to belong to multiple clusters would be favourable.

The desirable properties here are not exhaustive, but they give a number of clear directions for the development of clustering algorithms tailored to microarray data.

10.7 The Self-Organizing Tree Algorithm

SOTA (Herrero, Valencia, and Dopazo, 2001) combines both SOM and (divisive) hierarchical clustering. As in SOM, SOTA maps the original input gene profiles to an output space of nodes. However, the nodes in SOTA are in the topology (or geometry) of a binary tree instead of a two-dimensional grid. In addition, the number of nodes in SOTA is not fixed from the beginning (in contrast to SOM); the tree structure of the nodes grows during the clustering procedure. Starting from a binary tree with two leaves, the algorithm iterates between the following two steps (see Figure 10.1).

With the given tree structure fixed, the gene expression profiles are sequentially and iteratively presented to the nodes located at the leaves of the tree (these nodes are called cells). Subsequently, each gene expression profile is associated with the cell that maps closest to it. The prototype vector of this cell and its neighbouring nodes, including its parent node and its sister cell, are then updated based on some neighbourhood weighting parameters (which perform the same role as the neighbourhood function in SOM). Thus, a cell is moved into the direction of the expression profiles that are associated with it. This presentation of the gene expression profiles to the cells continues until convergence.

After convergence of the above procedure is reached, the cell containing the most variable population of expression profiles (the variation is defined here by the maximal distance between two profiles that are associated with the same cell) is replicated into two daughter cells (causing the binary tree to grow), whereafter the entire process is restarted.

The algorithm stops (the tree stops growing) when a threshold of variability is reached for each cell. In this way, the number of clusters does not need to be specified in advance. The threshold variability can be determined by means of permutation test of the data set.

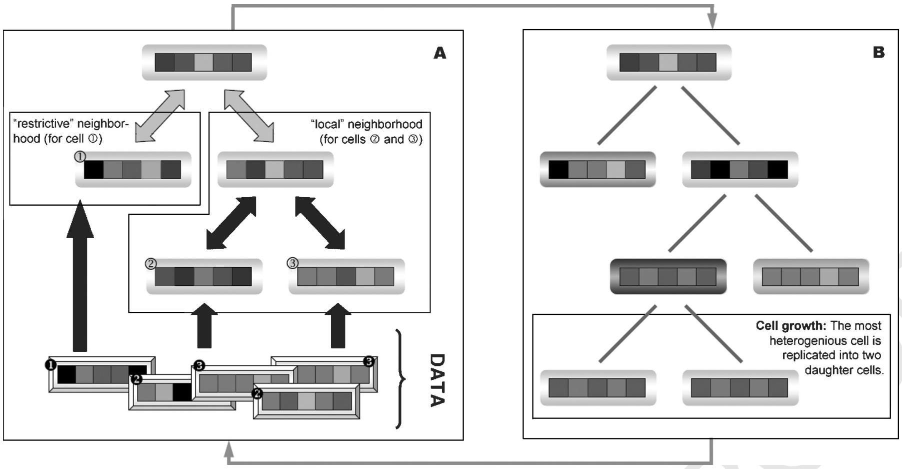

Figure 10.1 The iterative procedure of SOTA consists of two steps: (A) Each gene profile is associated with the cell whose prototype vector is located closest to it. Then the prototype vectors of the cells are updated based on the neighbourhood weighting parameters. (The black arrows between the nodes indicate where the updates take place, while the grey ones indicates where the updates are no longer performed.) This procedure iterates until convergence is reached. (B) The cell whose associated profiles exhibits the largest variability is duplicated into two daughter cells (the darker the cell, the more heterogeneous it is)

10.8 Quality-Based Clustering Algorithms

Quality-based algorithms produces clusters with a quality guarantee that ensures that all members of a cluster are coexpressed.

QT_Clust

Heyer, Kruglyak and Yooseph, (1999) introduced the concept of quality-based clustering. Their implementation is called QT_Clust; it is a greedy procedure that finds one cluster at a time. It considers each expression profile in the data in turn. For each expression profile, it determines which other profiles are within the specified distance in its neighbourhood. This specified distance therefore serves as the quality guarantee. In this way, a candidate cluster is formed for every expression profile. The candidate cluster with the largest number of expression profiles is selected as an 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 prespecified threshold.

By using a stringent quality guarantee, it is possible to find clusters with tightly related expression profiles (i.e. clusters containing highly coexpressed genes). Moreover, genes that are not really coexpressed with other members of the data set are not included in any of the clusters.

Adaptive quality-based cluster

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 center. Using a preliminary estimate of the radius (i.e. the quality) of the cluster, a cluster centre is located in an area where the density (i.e. the number) of gene expression profiles is locally maximal. 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 coexpressed. To this end, a bimodal and one-dimensional probability distribution (the distribution consists of two terms: one for the cluster and one for the rest of the data) describing the Euclidean distance between the data points and the cluster centre is fitted to the data using an expectation-maximization (EM) algorithm. The cluster is subsequently removed from the data and the whole procedure is restarted. Only clusters whose size exceeds a predefined number are presented to the user.

In adaptive quality-based clustering, the users have to specify a significance level as the threshold for quality control. This parameter has a strict statistical meaning and is therefore much less arbitrary (in contrast to the case in QT_Clust). It can be chosen independently of a specific data set or cluster and it allows for a meaningful default value (95 per cent) that in general gives good results. This makes the approach user friendly without the need for extensive parameter fine-tuning. Second, with the ability to allow the clusters to have different radii, adaptive quality-based clustering produces clusters adapted to the local data structure.

10.9 Mixture Models

Model-based clustering (Hartigan, 1975) has already been used in the past for other applications outside bioinformatics, but its application to microarray data is comparatively recent (Yeungo *et al.*, 2001; McLachlan, Bean and Peel, 2002).

Model-based clustering assumes that the data are generated by a finite mixture of underlying probability distributions, where each distribution represents one cluster. The problem, then, is to associate every gene (or experiment) with the best underlying distribution in the mixture, and at the same time to find out the parameters for each of these distributions.

Mixture model of normal distributions

When multivariate normal distributions are used, each cluster is represented by a hypersphere or a hyperellipse in the data space. The mean of the normal distribution gives the centre of the hyperellipse, and the covariance of the distribution specifies its orientation, shape and volume. The covariance matrix for each cluster can be represented by its eigenvalue decomposition, with the eigenvectors determining the orientation of the cluster, and the eigenvalues specifying the shape and the volume of the cluster. By using different levels of restrictions on the form of the covariance matrix (i.e. its eigenvectors and eigenvalues), one can control the trade-off between model complexity (the number of parameters to be estimated) and flexibility (the extent to which the model fits the data).

The choice of the normal distribution is partly based on its desirable analytic convenience. Moreover, the assumption for fitting a normal distribution to gene expression profiles is considered to be reasonable, especially when the standard preprocessing procedures (see Section 10.2.1) have been applied (Yeungo *et al.*, 2001; Baldi and Brunak, 2001). Of course, other underlying distributions, such as gamma distributions or mixtures of Gaussian and gamma distributions, can also be used to describe expression profiles. So far, no precise conclusions have been made on what is the most suitable distribution for gene expression data (Baldi and Brunak, 2001).

Regardless of the choice of underlying distributions, a mixture model is usually learned by an EM algorithm. Given the microarray data and the current set of model parameters, the probability to associate a gene (or experiment) to every cluster is evaluated in the E step. Then, the M step finds the parameter setting that maximizes the likelihood of the complete data. The complete data refer to both the microarray data (observed data) and the assignment of the genes (or experiments) to the clusters (unobserved data). The likelihood of the model increases as the two steps iterates, and convergence is guaranteed.

The EM procedure is repeated for different numbers of clusters and different covariance structures. The result of the first step is thus a collection of different models fitted to the data and all having a specific number of clusters and specific covariance structure. Then, the best model with the most appropriate number of clusters and covariance structure in this group of models is selected. This model selection step involves the calculation of the Bayesian information criterion (BIC) for each model.

Yeung *et al.* (2001) reported good results of such analysis as described above using their MCLUST software on several synthetic and real expression data sets.

Mixture of factor analysis

For the clustering experiments (e.g. tissue samples), however, a problem arises in fitting a normal mixture to the data because the number of genes is much larger than the number of experiments. To solve this problem, McLachlan, Bean and Peel (2002) applied a mixture of factor analysis to the clustering of experiments (see Figure 10.2). The idea can be interpreted as follows. A single factor analysis performs a dimensional reduction in the gene space of a cluster. That is to say, in factor analysis, vectors of experiments located in the original n-dimensional hyperellipse (where n represents the number of genes) are projected onto their corresponding vectors of factors located in an m-dimensional unit sphere (usually $m \ll n$). By using a mixture

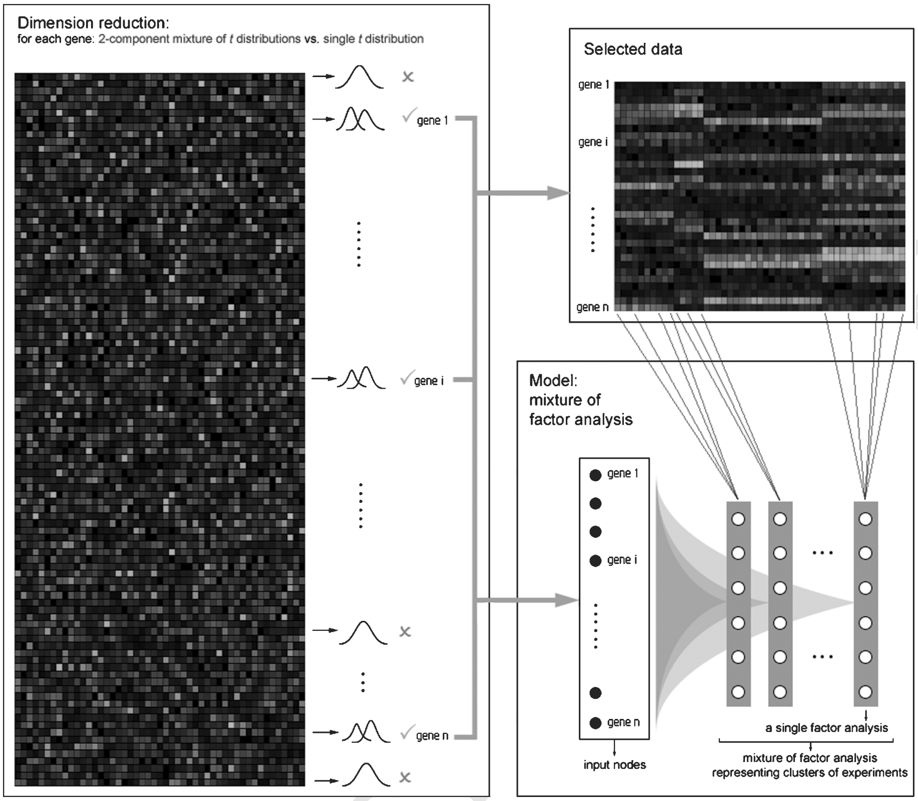

Figure 10.2 McLachlan, Bean and Peel (2002) use a two-component mixture model of *t* distributions to examine every gene expression profile against a single *t* distribution. Expression profiles to which the mixture models fit better (in terms of, for example, likelihood) are selected for further analysis. A mixture of factor analysis is applied on the selected data to cluster the experimental conditions

of factor analysis, clustering of the experiments is done on a reduced feature space (i.e. the *m*-dimensional factor space) instead of on the original huge-dimensional gene space. The EM algorithm is also used to learn the mixture of factor analysis model.

However, the choice for the number of factors in such a model remains a dilemma. If the number is too small, the full correlation structure of the genes cannot be captured; while if it is too large, the EM algorithm for the parametrization of the model can encounter computational difficulties. To alleviate the problem, McLachlan, Bean and Peel (2000) added another stage to reduce the dimension of the gene space before applying the mixture of factor analysis to the clustering of the experiements. In this stage, both a two-component mixture model of univariate *t* distributions (where the association of the experiments to the two components is unknown) and a single *t* distribution are fitted to the data for each gene. A threshold on the likelihood ratio between the two models is then applied to determine whether the gene is responsible for the clustering of experiments.

A *t* mixture model is more suitable for describing a gene expression profile than a normal mixture model because the former is more robust to outliers. A *t* distribution has an additional parameter called the degree of freedom compared with a normal distribution. The degree of freedom can be seen as a parameter for adjusting the thickness of the tail of the distribution. A *t* distribution with a relatively small degree of freedom will have a thicker tail than a normal distribution with the same mean and variance. However, as the degree of freedom goes to infinity, the *t* distribution approaches the normal distribution. Because of the thicker tail of a *t* distribution, the model learned for the *t* mixture is more robust to the outliers in gene profiles. Therefore, the degree of freedom can be viewed as a robustness tuning parameter.

10.10 Biclustering Algorithms

Biclustering means to cluster both the genes and the experiments at the same time. Among early papers on biclustering methods, clustering algorithms were applied (iteratively) to both dimensions of a microarray data set (Alon *et al.*, 1999; Getz, Levine and Domany, 2000). As a result, genes and experiments are reorganized so as to improve the manifestation of the patterns inherited in both the genes and the experiments. In other words, biclustering algorithms of this type divide the data into checkerboard units of patterns. More recently, other algorithms specifically designed for finding this kind of pattern have also been developed. An example is provided by Lazzeroni and Owen (2000), who used a plaid model – a specific form of mixture of normal distributions – to describe microarray data. EM was used for the parametrization of the model. For another example, the spectral biclustering method (Kluger *et al.*, 2003) applies SVD for solving the problem. However, this type of biclustering algorithm has limitations (Hasite *et al.*, 2000) when the expression profiles of some genes under study divide the samples by one biological explanation (say, tumour type) while some others divide the samples according to another biological process (e.g. drug response).

The second type of biclustering algorithm aims to find genes that are responsible for the classification of the samples. Examples are the gene shaving method (Hasite *et al.*, 2000), which searches for clusters of genes that vary as much as possible across the samples with the help of PCA; and a minimum description length method (Jörsten and Yu, 2003).

The third type of biclustering algorithm questions conventional clustering algorithms by the idea that genes that share functional similarities do not have to be coexpressed over all the experimental conditions under study. Instead of clustering genes based on their overall expressional behaviour, these algorithms look for patterns where genes share similar expressional behaviour over only a subset of experimental conditions. The same idea can be used for clustering the experimental conditions. Suppose a microarray study is carried out on tumour samples of different histopathological diagnoses. The problem then is to find tumour samples that have similar gene expression levels for a subset of genes (so as to obtain an expressional fingerprint for the tumour). To distinguish the two orientations for this type of biclustering problem, we will refer to the former case as biclustering genes, and the latter case as biclustering experiments. This type of biclustering algorithm was pioneered by Cheng and Church (2000), where a heuristic approach is proposed to find patterns as large as possible that have minimum mean squared residues, while allowing variance to be present across the experiments when biclustering genes (or across the genes when biclustering experiments). Model-based approaches have also been applied for this type of problem. Barash and Friedman (2002) used an EM algorithm for model parametrization, while Sheng *et al.* (2003) proposed a Gibbs sampling strategy for model learning.

The idea of applying Gibbs sampling to clustering was inspired by the success of the Gibbs sampling algorithm in solving the motif-finding problem (Thijs *et al.*, 2002). The model consists in associating a binary random variable (label) with each of the rows and each of the columns in the data set so that a value of 1 indicates that the row or the column belongs to the bicluster and a 0 indicates otherwise. Then the task of the algorithm is to estimate the value for each of these labels. The algorithm opts for Gibbs sampling, a Bayesian approach for the estimation, and examines the posterior distribution of the labels given the data (see Figure 10.3). Finally, a threshold is put on the posterior distribution and selects the rows and columns that have probabilities larger than the threshold as the positions of the bicluster. To find multiple biclusters in the data, the labels associated with the experiments for a found bicluster

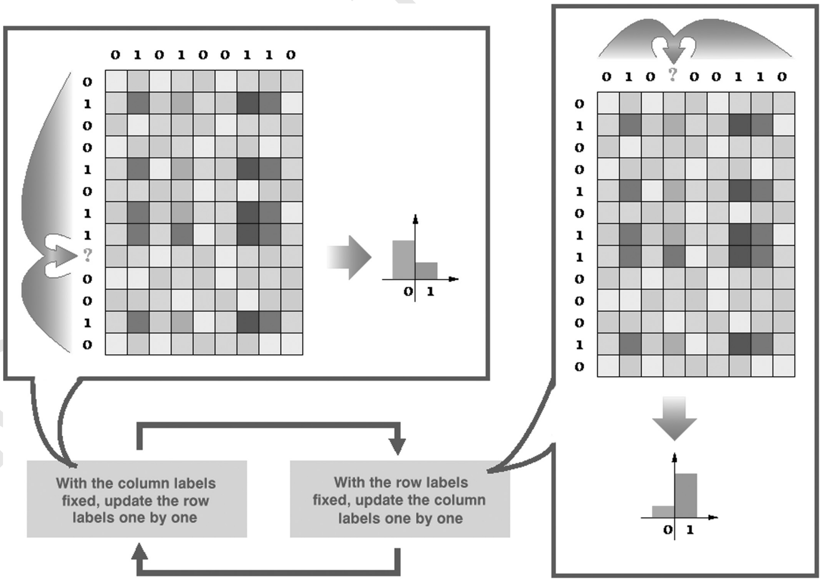
0 1 0 1 0 0 1 1 0
0 1 0 0 1 0 1 1 ? 0 0 1 0
0 1
0 1 0 ? 0 0 1 1 0
0 1 0 0 1 0 1 1 0 0 0 1 0
0 1
With the column labels fixed, update the row labels one by one
With the row labels fixed, update the column labels one by one

Figure 10.3 With all the other labels fixed, the Gibbs biclustering algorithm calculates the posterior conditional distribution of a label (indicating whether a gene or a condition belongs to the bicluster) at each iteration. Subsequently, a label is drawn from the obtained conditional distribution and is assigned to the gene or the experimental condition

are set permanently to zero when looking for further clusters. The masking of the experiments is chosen for both biclustering the genes and biclustering the experiments based on the idea that a gene should be allowed to belong to different clusters.

10.11 Assessing Cluster Quality

As mentioned before, different runs of clustering will produce different results, depending on the specific choice of preprocessing, algorithm, distance measure, and so on. Many methods often produce clusters even for random data. Therefore, validation of the relevance of the cluster results is of utmost importance. Validation can be either statistical or biological. Statistical cluster validation can be done by assessing cluster coherence, by examining the predictive power of the clusters, or by testing the robustness of a cluster result against the addition of noise.

Alternatively, the relevance of a cluster result can be assessed by a biological validation. Of course it is hard, not to say impossible, to select the best cluster output, since 'the biologically best' solution will be known only if the biological system studied is completely characterized. Although some biological systems have been described extensively, no such completely characterized benchmark system is now available. A common method to biologically validate cluster outputs is to search for enrichment of functional categories within a cluster. Detection of regulatory motifs is also an appropriate biological validation of the cluster results (Tavazoie *et al.*, 1999). Some of the recent methodologies described in the literature to validate clustering results are discussed as follows.

1. *Testing cluster coherence*. Based on biological intuition, a cluster result can be considered reliable if the within-cluster distance is small (i.e., all genes retained are tightly coexpressed) and the cluster has an average profile well delineated from the remainder of the data set (i.e. a maximal inter-cluster distance). Such criteria can be formalized in several ways, such as the sum-of-squared-error criterion of *k*-means, silhouette coefficients (Kaufman and Rousseeuw, 1990) or Dunn's validity index (Azuaje, 2002).

2. *Figure of merit*. The FOM (Yeung, Haynor and Ruzzo, 2001) is a simple quantitative data-driven methodology that allows comparisons between outputs of different clustering algorithms in terms of their predictive power. The methodology is related to the jackknife approach and the leave-one-out cross-validation. The clustering algorithm (for the genes) is applied to all experimental conditions (the data variables) except for one left-out condition. If the algorithm performs well, we expect that if we look at the genes from a given cluster their values for the left-out condition will be highly coherent. Therefore, for each cluster, the sum of squared deviations is computed for the expression levels under the left-out condition and over all the genes in the cluster. With the left-out condition fixed, the

FOM is subsequently calculated as the root mean of these sums obtained for all the clusters. The aggregate FOM is further computed as the sum of the FOMs over all the experimental conditions so as to compare different clustering algorithms.

3. *Sensitivity analysis*. Gene expression levels are the superposition of real biological signals and experimental errors. A way to assign confidence to a cluster membership of a gene consists in creating new in silico replicas of the microarray data by adding to the original data a small amount of artificial noise and clustering the data of those replicas. If the biological signal is stronger than the experimental noise in the measurements of a particular gene, adding small artificial variations (in the range of the experimental noise) to the expression profile of this gene will not drastically influence its overall profile and therefore will not affect its cluster membership. Through some robustness statistics (Bittner *et al.*, 2000), sensitivity analysis lets us detect which clusters are robust within the range of experimental noise and therefore trustworthy for further analysis.

   The main issue in this method is to choose the noise level for sensitivity analysis. Bittner *et al.* (2000) perturbed the data by adding random Gaussian noise with zero mean and a standard deviation that is estimated as the median standard deviation for the log-ratios for all genes across the experiments.

   The bootstrap analysis methods described by Kerr and Churchill (2001) use the residual values of a linear analysis of variance (ANOVA) model as an estimate of the measurement error. By using an ANOVA model, non-consistent measurement errors can be separated from variations caused by alterations in relative expression or by consistent variations in the data set. The residuals are subsequently used to generate new replicates of the data set by bootstrapping (adding residual noise to estimated values).

4. *Use of different algorithms*. Just as clustering results are sensitive to adding noise, they are sensitive to the choice of clustering algorithm and to the specific parameter settings of a particular algorithm. Many clustering algorithms are available, each of them with different underlying statistics and inherent assumptions about the data. The best way to infer biological knowledge from a clustering experiment is to use different algorithms with different parameter settings. Clusters detected by most algorithms will reflect the pronounced signals in the data set. Again, statistics similar to those of Bittner *et al.* (2000) are used to perform these comparisons. (See Chapter 11 for a further discussion of the use of different algorithms.)

5. *Enrichment of functional categories*. One way to biologically validate results from clustering algorithms is to compare the gene clusters with existing functional classification schemes. In such schemes, genes are allocated to one or more functional categories (Tavazoie *et al.*, 1999; Segal *et al.*, 2001) representing their biochemical properties, biological roles and so on. Finding clusters that have been

significantly enriched for genes with similar function is a proof that a specific clustering technique produces biologically relevant results.

Using the cumulative hypergeometric probability distribution, we can measure the degree of enrichment by calculating the probability or *P*-value of finding by chance at least *k* genes in this specific cluster of *n* genes from this specific functional category that contains *f* genes out of the whole *g* annotated genes

$$P = 1 - \sum_{i=0}^{k-1} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}} = \sum_{i=k}^{\min(n,f)} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}}.$$

These *P*-values can be calculated for each functional category in each cluster. Note that these *P*-values must be corrected for multiple testing according to the number of functional categories.

10.12 Open Horizons

When research on clustering of microarray data started, a common opinion was that clustering was a ‘closed’ area of statistical research where little innovation was possible. Dozens of papers about clustering microarray data have now been published, demonstrating time and again significant improvements over classical methods. Yet, classical methods (in particular hierarchical clustering) remain dominant in biological applications, despite real shortcomings. The conclusion most probably is that new methods have not demonstrated sufficient added value to overcome the *status quo* established by a few pioneering works. As an example, Table 10.1 provides a summary of how well the second-generation clustering algorithms described in this paper meet our wish list presented in Section 10.6.

Lack of benchmarking significantly impairs the demonstration of major improvements. This situation is itself created by the subjectivity of interpreting clustering results in many situations and weak benchmarks (such as the yeast cell cycle data set by Cho *et al.*, 1998) have only added to the confusion. The most likely way out is the production of a large, carefully designed set of microarray experiments, specifically dedicated to the evaluation of clustering algorithms.

Another major open problem is the limited connection between clustering and biological knowledge. Clustering does not stand by itself but is tightly linked to the biological interpretation of its results and the subsequent use of these results. Cluster methods that incorporate functional, regulatory and pathway information directly in the algorithm are highly desirable. Also, clustering is only the starting point for further analysis, so strategies that integrate clustering tightly with its downstream analysis (e.g. regulatory sequence analysis, guilt by association) will improve on the final biological predictions (Moreau *et al.*, 2002). Probabilistic relational models and

Table 10.1 How well do the second-generation clustering algorithms meet our wish list?

	Decision on no. of clusters	Assigen every gene to a particular cluster?	Robustness	Time complexity	Allow a gene in multiple clusters?
SOTA	By putting a threshold on the variability of the cells	Yes	Comparable to that of SOM	Linear in no. of expression profiles	No
QT_clust	By putting a threshold on the quality of a cluster	No	Global solution	Quadratic in no. of expression profiles	No
Adap. qual. based	By specifying a significance level	No	Global solution	Linear in no. of expression profiles	No
Model based	By model comparison in terms of BIC	No	The use of EM leads to local minimum solutions	Depends on the implementation	Yes
Gibbs biclustering	Automatic decision	No	The chance for finding local minima is reduced (comparing with EM)	Linear performance can be achieved depending on the implementation	Yes

their variants, such as biclustering algorithms, hold a great potential in this regard, as already demonstrated in some applications (Segal *et al.*, 2001, 2003).

References

Alon, U., Barkai, N., Notterman, D. A. *et al.* (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci USA*, **96**, 6745–6750.

Azuaje, F. (2002) A cluster validity framework for genome expression data. *Bioinformatics*, **18**(2), 319–320.

Baldi, P. and Brunak, S. (2001) *Bioinformatics: the Machine Learning Approach, Adaptive Computation and Machine Learning*, 2nd edn. MIT Press, Cambridge, MA.

Barash, Y. and Friedman, N. (2002) Context-specific bayesian clustering for gene expression data. *J. Comput. Biol.*, **9**, 169–191.

Bar-Joseph, Z., Gifford, D. K. and Jaakkola, S. (2001) Fast optimal leaf ordering for hierachical clustering. *Bioinformatics*, **17**(Suppl. 1), S22–S29.

Bittner, M., Meltzer, P., Chen, Y. *et al.* (2000) Molecular classfication of cutaneous malignant melanoma by gene expression profiling. *Nature*, **406**, 536–540.

Cheng, Y. and Church, G. M. (2000) Biclustering of expression data. In *ISMB 2000 Proceedings*, 93–103.

Cho, R. J., Campbell, M. J., Winzeler, E. A. (1998) A genome-wide transcriptional analysis of mitotic cell cycle. *Mol Cell*, **2**, 65–73.

De Smet, F., Mathys, J., Marchal, K., Thijs, G., De Moor, B. and Moreau, Y. (2002) Adaptive quality-based clustering of gene expression profiles. *Bioinformatics*, **18**(5), 735–746.

Duda, R. O., Hart, P. E. and Stork, D. G. (2001) *Pattern Classification*, 2nd edn. Wiley, New York.

Durbin, B. P. and Rocke, D. M. (2004) Variance-stabilizing transformations for two-color microarrays. *Bioinformatics*, **20**(5), 660–667.

Eisen, M. B., Spellman, P. T., Brown, P. O. and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, **95**, 14 863–14 868.

Getz, G., Levine, E. and Domany, E. (2000) Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad. Sci. USA*, **97**(22), 12 079–12 084.

Hartigan, J. A. (1975) *Clustering Algorithms (Wiley Series in Probability)*. Wiley, New York.

Hastie, T., Tibshirani, R., Eisen, M. B., (2000) ‘Gene shaving’ as a method for identifying distinct sets of genes with similar expression patterns. *Genome Bio*, **1**(2), research0003.1–0003.21.

Herrero, J., Valencia, A. and Dopazo, J. (2001) A hierachical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics*, **17**(2), 126–136.

Heyer, L. J., Kruglyak, S. and Yooseph, S. (1999) Exploring expression data: identification and analysis of coexpressed genes. *Genome Res*, **9**, 1106–1115.

Jöorsten, R. and Yu, B. (2003) Simultaneous gene clustering and subset selection for sample classification via MDL. *Bioinformatics*, **19**(9), 1100–1109.

Kaufman, L. and Rousseeuw, P. J. (1990) Finding Groups in Data: an Introduction to Cluster Analysis, Wiley, New York.

Kerr, M. K. and Churchill, G. A. (2001) Bootstrapping cluster analysis: Assessing the reliability of conclusions from microarray experiments. *Proc. Natl. Acad. Sci. USA*, **98**(16), 8961–8965.

Kluger, Y., Basri, R., Chang, J. T. and Gerstein, M. (2003) Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Res*, **13**, 703–716.

Kohonen, T. (1995) *Self-Organizing Maps (Springer Series in Information Sciences)*. Springer.

Lazzeroni, L. and Owen, A. (2000) *Plaid Models for Gene Expression Data*, technical report. Department of Statistics, Stanford University.

Lukashin, A. V. and Fuchs, R. (2000) Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters. *Bioinformatics*, **17**(5), 405–414.

McLachlan, G. J., Bean, R. W. and Peel, D. (2002) A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, **18**(3), 413–422.

Moreau, Y., De Smet, F., Thijs, G., Marchal, K. and De Moor, B. (2002) Functional bioinformatics of microarray data: From expression to regulation. *Proc. IEEE*, **90**(11), 1722–1743.

Quackenbush, J. (2001) Computational analysis of microarray data. *Nature Rev.*, **2**, 418–427.

Segal, E., Shapira, M., Regev, A. (2003) Module networks: identifying regulatory modules and their condition-specific regulators for gene expression data. *Nature Genetics*, **34**(2), 166–176.

Segal, E., Taskar, B., Gasch, A., Friedman, N. and Koller, D. (2001) Rich probabilistic models for gene expression. *Bioinformatics*, **17**(Suppl. 1), S243–S252.

Sheng, Q., Moreau, Y. and De Moor, B. (2003) Biclustering microarray data by Gibbs sampling. *Bioinformatics*, **19**(Suppl. 2), II196–II205.

Sherlock, G. (2000) Analysis of large-scale gene expression data. *Curr. Opin. Immunol*, **12**, 201–205.

Spellman, P. T., Sherlock, G., Zhang, M. Q. *et al.* (1998) Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol Bio Cell*, **9**, 3273–3297.

Tamayo, P., Slonim, D., Mesirov, J. *et al.* (1999) Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc Natl Acad Sci USA*, **96**, 2907–2912.

Tavazoie, S., Hughes, J. D., Campbell, M. J., Cho, R. J. and Church, G. M. (1999) Systematic determination of genetic network architecture. *Nature Genetics*, **22**, 281–285.

Thijs, G., Marchal, K., Lescot, M. (2002) A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *J Comput Biol*, **9**, 447–464.

Troyanskaya, O., Cantor, M., Sherlock, G. *et al.* (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**(6), 520–525.

Ward, J. H. (1963) Hierarchical grouping to optimize an objective function. *J Am Stat Assoc*, **58**, 239–244.

Yeung, K., Fraley, C., Murua, A., Raftery, A. and Ruzzo, W. (2001) Model-based clustering and data transformations for gene expression data. *Bioinformatics*, **17**(10), 977–987.

Yeung, K., Haynor, D. and Ruzzo, W. (2001) Validating clustering for gene expression data. *Bioinformatics*, **17**(4), 309–318.