# Cluster analysis of microarray data

Qizheng Sheng[†], Yves Moreau[†§], Frank De Smet[†], Kathleen Marchal[†], Bart De Moor[†]

[†] Department of Electrical Engineering, ESAT-SCD, K.U.Leuven,
Kasteelpark Arenberg 10, 3001 Leuven-Heverlee, Belgium,
http://www.esat.kuleuven.ac.be/~dna/BioI

[§] On leave at Center for Biological Sequence Analysis
Danish Technical University
Kemitorvet, Building 208, Lyngby, Denmark
http://www.cbs.dtu.dk

**Abstract**

The ability of microarray chip to capture the expressional level of thousands of genes in one snapshot becomes a major attraction for biologists. By performing parallel microarray experiments under different conditions, biologists seek useful information of the underlying biological process that lies in the hundreds of thousands of data points obtained. The first step of such a task is often to cluster the genes into biological meaningful groups according to their pattern of expression, based on the assumption that expressional similarity of genes implies their functional similarity. In this paper, we give an overview on various clustering techniques that have been developed for micorarray data analysis. The clustering methods reviewed in this paper include conventional clustering methods (such as hierarchical clustering, $k$-means clustering, and self-organizing maps), as well as several clustering methods devotedly developed for gene expression data analysis.

## 1 Introduction

During the past few years, microarray technology has emerged as an effective technique to measure the level of expression of thousands of genes in a single experiment. This capacity of microarrays becomes a major attraction for molecular biologists. By monitoring the expressional behavior of the genes under different experiments (e.g., experiments performed under different time points during a certain biological process such as different phases of the cycle of cell division, or experiments where samples are taken from tumor cells with a different histopathological diagnosis), molecular biologists seek useful information lying in the hundreds of thousands of data points obtained, which may provide answers to their questions. However, without the help of powerful computational and statistical techniques, analyzing data in such immense amount and of such complexity is impractical.

The first level of interest for molecular biologists is to identify genes whose expression level is significantly changed under different experimental conditions. Straightforward statistical techniques can be applied to solve this problem efficiently [4]. However, such an analysis treats the genes as individuals rather than exploring their relation with each other. On the other hand, for every gene, the detailed information about its expression profile as a whole over all

the experiment under study is neglected in this first-level analysis. To make better use of the full-scale information provided by microarray experiments, the next level of insight is provided by clustering genes into biological meaningful groups according to their pattern of expression. Comparing with the full data itself, such groups of related genes are much more tractable for further studies including gene function and regulation.

Based on the assumption that expressional similarity implies 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 to the problem of clustering genes into groups based on their similarity in expression profiles. Genes that have similar expression profiles are said to be coexpressed.

The first generation of clustering algorithms (e.g., hierarchical clustering [14], $k$-means [21] and self-organizing maps (SOM) [30]) applied to gene expression profiles were mostly developed outside biological research. Although encouraging results have been produced [47, 49, 48], some of their characteristics often complicate their use for clustering expression data [45]. They require, for example, the predefinition of one or more user-defined parameters that are hard to estimate by a biologist (e.g., the predefinition of number of clusters in $k$-means and SOM – this number is almost impossible to predict in advance). Moreover, changing these parameter settings will often have a strong impact on the final result. These methods therefore need extensive parameter fine-tuning, which means that a comparison of the results with different parameter settings is almost always necessary – with the additional difficulty that comparing the quality of the different clustering results is hard. Another problem is that the first-generation algorithms often force every data point into a cluster. In general, a considerable number of genes included in the microarray experiment do not contribute to the biological process studied, and these genes will therefore have seemingly constant or even random expression profiles rather than having similar expression profiles with the other genes. Including these noisy genes in one of the clusters will contaminate their content and make these clusters less suitable for further analysis. Finally, the computational and memory complexity of some of these algorithms often limit the number of expression profiles that can be analyzed at once.

In addition to the above limitations of the first-generation clustering algorithms, the specific characteristics of microarray data, such as the high dimensionality, the highly noisy measurements, the complex biological process hidden behind, in general has created the need for clustering methods particularly tailored. Accordingly, desired features for microarray data cluster analysis include fast calculation speed, robustness, determinism, easy interpretation, and so on.

A second generation of clustering algorithms have started to tackle some of the limitations of the earlier methods, while seeking to meet those specific requirements for microarray data. These algorithms include model-based algorithms [59, 20, 35], the self-organizing tree algorithm [24], quality-based algorithms [25, 46], biclustering algorithms [44, 7], simulated annealing [34], gene shaving [22], and the cluster affinity search technique [8]. Also, some procedures were developed that could help biologists to estimate some of the parameters needed for the first generation of algorithms (such as the number of clusters present in the data [20, 34, 59]).

While it is impossible to give an exclusive survey of all the clustering algorithms that have been developed for gene expression data, we are trying to present here a tutorial that illustrates the key issues on this subject. The clustering methods presented in this paper include both the first-generation and the second-generation algorithms. The selection of algorithms is based on their popularity, their ability to handle the specific characteristics of microarray data, and inevitably some personal biases. In the meantime, we try to cover as many techniques that are

used for the cluster analysis of gene expression data as possible to provide some inspiration for further development. This paper is organized as follows:

In Section 2, we review the main characteristics of microarray data. In Section 3, we discuss an important procedure before one can perform cluster analysis on gene expression profiles – the preprocessing of microarray data, which is needed to overcome some of the difficult artifacts of microarray data. The result of cluster analysis can be greatly enhanced by a proper preprocessing procedure. As the final preparation before we go into deeper discussion of clustering techniques on microarray data, in Section 4, we address some other basic but necessary ideas such as the orientation of clustering (clustering genes vs. clustering experiments) and distance metrics commonly used to compare gene expression profiles.

We discuss the application of classical clustering algorithms to microarray data in Section 5, 6, and 7, where hierarchical clustering, $k$-means clustering, and self-organization maps are addressed respectively. We refer to these algorithms as the first-generation clustering algorithms.

Then, in Section 8, we identify common drawbacks of the first-generation clustering algorithms. Taking together into account the characteristics of microarray data, we give a wish list of desirable features an ideal clustering algorithm should carry.

In the following sections, we give a survey of the second-generation clustering algorithms. These algorithms include the self-organizing tree algorithm (SOTA) [24] (Section 9), the quality-based clustering algorithms [25, 46] (Section 10), mixture models for microarray data [59, 35] (Section 11) and biclustering algorithms [22, 44, 43] (Section 12).

Changes in details such as the preprocessing procedures, the algorithm, or even the distance metrics might lead to different clustering result. Thus, in Section 13, we discuss methods used to validate clustering results.

## 2 Specific characteristics of microarray data

A microarray chip captures the levels of expression of hundreds of thousands of genes under one experiment. Studies are usually done by using microarray experiments under different conditions so that the parallel comparison between the expressional behaviors of the genes can be made. The outcome of a study as such is usually put in a matrix, where the genes are represented as rows and the experiments are listed in the columns. Each row of the matrix record the expression profile of a gene (across different experiments under study). Microarray data in general has the following main characteristics:

1. *Dimensionality:* While, as we mentioned, the number of rows of the matrix can contain hundreds of thousands of genes, the dimension of the columns is much smaller. Current cost for a microarray chip limits the number of experiments in a study. Most labs can afford studies done under around 10 experiments. A relatively large study can use up to a couple of hundred of chips, which is still a small number comparing to the number genes.

2. *Noise:* In a cDNA microarray experiment for example, the measurement of gene expression level depends on the RNA extraction from a biological sample, the preparation of fluorescently labeled complementary DNA (cDNA) from the obtained RNA, the hybridization of the cDNA to the corresponding spot on the chip, and the image processing procedure

to read out the hybridization intensity. Each of these steps can introduce a considerable amount of noise into the final microarray data matrix.

3. *Redundancy:* The biological process under scrutiny in a microarray study is assumably a complicated process, which involves concerted gene reactions in different pathways. While some genes can even be involved in more than one pathway, some others, however, might not be relevant to the biological process. These genes usually show little variation over the different experiments under study. Genes that show little variation over the different experiments are called constitutive with respect to the biological process studied. Constitutive genes often contribute to a large proportion of the whole population of the genes included in a microarry study.

All these features of microarray data should be taken care with precaution when performing analysis. In the next section, we talk about the preprocessing procedure that can help to moderate those unfavorable characteristics in microarray data.

## 3 Preprocessing: prepare microarray data for cluster analysis

A correct preprocessing strategy, which not only removes as much as possible the systematic noise presented in a microarray data but also provides a basis for the comparison between genes, is almost as important as the cluster analysis itself. A common procedure of preprocessing includes the following five steps [38].

1. *Normalization:* First, it is necessary to normalize the hybridization intensities within a single experiment before one can compare the results from different microarray experiments. In a two-channel cDNA microarray experiment, for example, several sources of noise (such as differences in labelling, in detection efficiency, and in quantity of initial RNA within the two channels) create systematic sources of biases. A normalization step can help to compute and remove the biases to correct the data. Since many sources can be considered and since they can be estimated and corrected in a variety of ways, many different normalization procedures exist. It is not the purpose of this paper to make a survey on this subject, we therefore refer to [39] for more details.

2. *Nonlinear transformation:* After normalization, it is common practice to pass expression values through a nonlinear function, such as a logarithm. This is especially suited for dealing with expression ratios (coming from two-channel cDNA microarray experiments, using a test and reference sample), since expression ratios are not symmetrical [39] in the sense that upregulated genes have expression ratios between one and infinity, while downregulated genes have expression ratios squashed between one and zero. Taking the logarithms of these expression ratios results in symmetry between expression values of up- and downregulated genes.

3. *Missing value replacement:* Microarray experiments often contain missing values (measurements absent because of technical reasons). The inability of many cluster algorithms to handle such missing values necessitates the replacement of these values. Simple replacements such as a replacement by zero or by the average of the expression profile often disrupt these profiles. Indeed, replacement by average values relies on the unrealistic assumption that all expression values are similar across different experimental conditions.

Because of an erroneous missing value replacement, genes containing a significant number of missing values can be assigned to the wrong cluster. More advanced techniques of missing value replacement (which use the $k$-nearest neighbor method or the singular value decomposition) have been described [54] and take advantage of the rich information provided by the expression patterns of other genes in the data set.

Finally, note that some implementations of algorithms use only the measured values to derive the clusters and as such obviates the need for missing value replacement [46].

4. *Filtering:* As stated in Section 2, for any microarray study, a considerable part of the data is composed of genes that do not contribute to the underlying biological progress and other constitutive genes. The expression profiles of these genes show little variation over the different experiments. In addition, these constitutive genes will have seemingly random and meaningless profiles after standardization (division by a small standard deviation results in noise inflation), which is also a common preprocessing step (see further). Another problem comes from 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 [14] removes gene expression profiles from the data set that do not satisfy some simple criteria. Commonly used criteria include a minimum threshold for the standard deviation of the expression values in a profile and a threshold on the maximum percentage of missing values.

5. *Standardization or rescaling:* Biologists are mainly interested in grouping gene expression that have the same relative behavior; i.e., genes that are up- and downregulated together. Genes showing the same relative behavior but with diverging absolute behavior (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 (for the discussion of different distance measures, see Section 4.2 for details). 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 to have zero mean and unit standard deviation. Gene expression profiles showing the same relative behavior will have a small(er) Euclidean distance after rescaling [39].

# 4 A few more notes before applying cluster analysis

Before we dive into the detailed discussion of various clustering algorithms, there are yet a couple of common issues shared by different clustering techniques, of which we would like to remind the readers.

## 4.1 Clustering genes vs. clustering experiments

So far, we have addressed the clustering problem of microarray data only as an analysis to find genes that behave similarly over the experimental conditions. As addressed above, underlying the clustering genes is the assumption that genes that have similar expressional behavior share functional similarities in the biological progress under study.

Instead of clustering genes, we can also cluster experimental conditions, where the task is to find groups of experiment conditions (which can be, for example, tumor samples) across which

the all the genes behave similarly. This types of clustering can be helpful for problems like histopathological discovery of tumors whose types are primarily unknown.

To put the clustering problems in a mathematical language, when we cluster genes, each expression profile is often represented as a data point (or a vector, pointing from the origin) in a high-dimensional space whose dimension equals to the number of experiments presented in the study. The Euclidean axis of the data point is given by its expression value under every experiment. Similarly, when experiments are to be clustered, it is the experiments that are presented as data points (or vectors) in a Euclidean space whose dimension equals the number of genes in the data set. In both of the cases, the problem then is to find clusters of the data points in these high dimensional spaces.

### 4.2 Distance metrics

Depending on the way to define a cluster, clustering methods can be divided into two types – model-based clustering methods and distance-based clustering methods. Model-based clustering methods consider the data points in the high-dimensional space under study as a whole. Their aim is to discover the underlying structure of these data points. To be more specific, model-based clustering algorithms assume that the data points in the high-dimensional space as 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 methods in detail in Section 11.

Distance-based clustering methods (to which most of the classical clustering methods, such as hierarchical clustering, $k$-means, and SOM belong), on the other hand, explore the pairwise relation between the data points in the high-dimensional space under study. In their case, a cluster contains data points whose pairwise distance between one another is lower than a threshold value. Here, the problem arises from how the distance should be defined. Hereunder, we give a brief overview of common distance metrics applied for clustering microarray data.

1. *Pearson correlation:* This is the most commonly used distance metric. It is often denoted by $r$. Pearson correlation is the dot product of two normalized vectors. In another word, it is the cosine between two vectors. This means that it measures the similarity in the shapes of two profiles, while not taking the magnitude of the profiles into account. Therefore, Pearson correlation suits well the intuition of biologists for what it means that two expression profiles to be "coexpressed" [14].

2. *Squared Pearson correlation:* This is the squared product of Pearson correlation. Therefore, squared Pearson correlation $r^2$ neglects the sign in front of a Pearson correlation measure and considers two vectors pointing to the exact opposite directions to be perfectly similar (i.e., in this case, $r = -1$ while $r^2 = 1$). This character makes the squared Pearson correlation able to capture inverse relationships among gene expression profiles, which might also be interesting for biologists.

3. *Euclidean distance:* Euclidean distance measures the absolute distance between two points in space. That is, it is the length of the straight line connecting the two points. As we mentioned in Section 3, the Euclidean distance measures the similarity between the absolute behaviors of genes, while the biologists are more interested in their relative behaviors. Thus, when using Euclidean distance metric, a standardization procedure is needed before clustering can be applied to the gene expression profiles. Note here that after the

standardization, the Euclidean distance is related to the Pearson correlation between two points $x$ and $y$ by $|x - y|^2 = 2(1 - |r|)$ [2].

4. *Jackknife correlation:* New techniques of distance measurements are developed to overcome some of the shortcomings of the classical distance metrics. One of the examples is the jackknife correlation [25], which is an improvement for Pearson correlation measure. Although Pearson correlation gives high score to coexpressed genes, it is not robust to outliers. An extreme case of a false positive is, for example, when two genes are totally unrelated under all but one experimental conditions, where they both have a high peak. These high peaks then pull the vector presentations of the two profiles to the same direction, and therefore the two genes will have a high Pearson correlation. Jackknife correlation increases the robustness to single outliers by computing a collection of all the possible leave-one-(experiment)-out Pearson correlation between two genes and then select the minimum of the collection as the final measure for the correlation.

# 5 Hierarchical clustering

The first introduction of hierarchical clustering to the world of biology is its application on the construction of phylogenetic trees. Early applications of the method on gene expression data analysis includes [56, 57, 14, 47, 1], where the usefulness (in for example finding functionally related genes) of hierarchical clustering on microarray data analysis was proved.

Hierarchical clustering has become the most widely used method for clustering gene expression data and can be seen as the *de facto* standard, probably because of its advantage of being a simple method and that the results can be nicely visualized. As the result of a hierarchical cluster analysis, not only a dendrogram is produced recording the whole clustering process as a tree, but also the original data is often reorganized in a heat map demonstrating the relationships between genes or conditions. We will discuss the visualization of hierarchical clustering in detail in 5.3.

Concerning the clustering procedure, two approaches to hierarchical clustering are possible: divisive clustering (a top-down approach as is used in [2], and agglomerative clustering (a bottom-up approach, see for example [14]). In the agglomerative approach, each vector is initially assigned to a single 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. The divisive approach processes the data the other way around. It first assigns all the vectors in the space to one cluster, and in each step a cluster is split up, until every vector is in a separate cluster. The stepwise computational complexity is simpler in agglomerative clustering than the divisive clustering, but the latter is more useful when one is more interested in the main structure of the data [27] or when the number of clusters (say, $k$) presented in the data is known in advance [13] (since the stopping criteria for the algorithm can then be modified so that the splitting procedure is no longer performed when $k$ clusters are produced).

However this advantage of the divisive approach is not often helpful in the analysis of gene expression data, because for the visualization of the resulting reorganized microarray data, a full dendrogram is usually desired by biologists. 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.

Because the agglomerative clustering is more commonly used in microarray data analysis, we will discuss the method in detail in Section 5.1 and 5.2. For divisive clustering, see [21, 27] for more information, and for its application on gene expression data, see [2].

Before we go further, we want to remind you of the major draw-back of hierarchical clustering. Whether agglomerative or divisive, hierarchical clustering can never repair a decision (to merge in agglomerative clustering, and to split in the divisive one) made in previous steps [27]. It is, after all, based on a stepwise optimization procedure rather than finding $k$ optimal clusters globally. Another against toward hierarchical clustering is that the nature of the hierarchical clustering (where data points are forced into a strict hierarchy of nested subsets) fits more into the context of building a phylogenetic tree [48] rather than that of grouping expression profiles.

### 5.1 Distance measure between two clusters

As we mentioned, in every step of the 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 three common options for agglomerative clustering in this regard.

1. *Single Linkage:* This method is also called the nearest neighbor method, because it defines the distance between two clusters as the distance between the two closest data points in different clusters.
2. *Complete Linkage:* On the contrary to single linkage, complete linkage uses the distance between two farthest data points in different clusters to define the distance between the two clusters under consideration, and thus is also called the farthest neighbor method.
3. *Average Linkage:* Both single linkage and complete linkage are sensitive to outliers [13]. Average linkage provides an improvement by defining the distance between two clusters as the average of the pairwise distances between data points in different clusters.

These methods yield similar results if the data consists of compact and well-separated clusters. However, if some of the clusters are close to each other or if the data has a dispersed nature, the results can be quite different [13].

### 5.2 Ward's method

At each step of the agglomerative clustering, instead of merging the two clusters that minimizes the pairwise distance between clusters, Ward's method [55] merges the two clusters that minimizes 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. (The sum-of-squared error is one of the criteria for assessing the quality of a cluster, see Section 13 for a detailed discussion on the validation of clusters.)

### 5.3 Visualization of the results

A heat map presenting the gene expression data, with a dendrogram to the side indicating the relationship between genes (or experimental conditions), see Figure 1, is the standard way

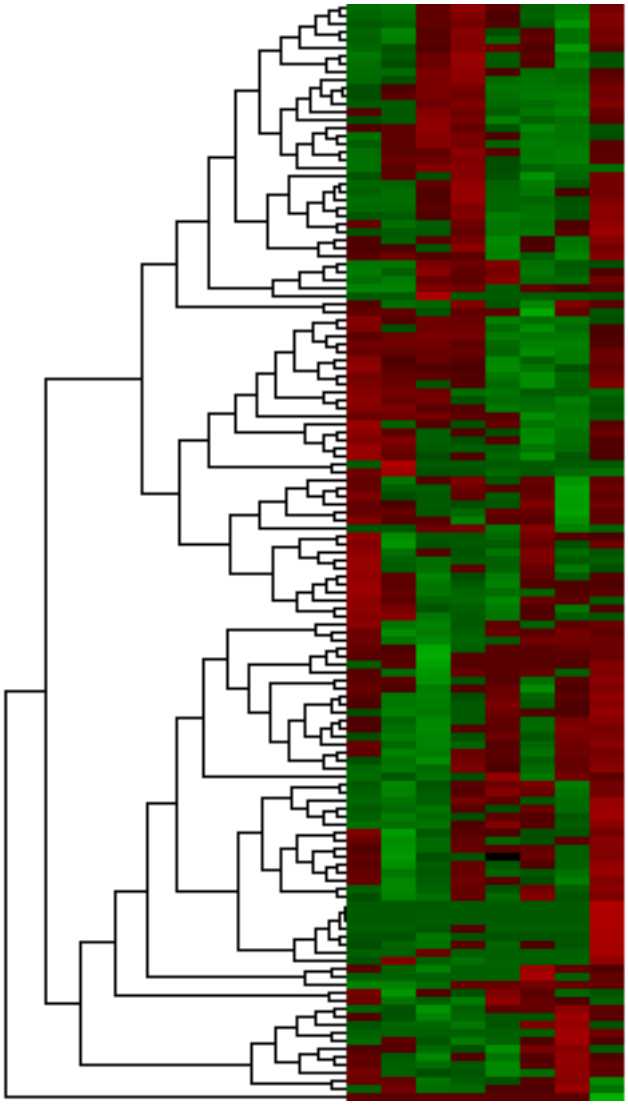

Figure 1: A typical output figure of hierarchical clustering to cluster genes in a microarray data set

to visualize the result of hierarchial cluster analysis on microrray data. In such a heat map, every row represents a gene expression profile, and the experiments are presented as columns. The heat map often colors the unchanged gene expression with black, the upregulated gene expression with red, and the downregulated one with green. However, the colors of red and green are sometimes replaced by other colors so that the heat map is also easily visible for color-blind people. The terminal branches (i.e., the leaves) of the dendrogram are linked with the individual genes (or experiments). The length of a branch is proportional to the pairwise distance between the clusters. The leaves of the dendrogram, and accordingly the rows (or the columns) of the heat map, can be swapped so that the similarity between adjacent genes (or experimental conditions) are 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 [14] by weighting genes using combined source of information, and then placing the genes with lower average weight earlier in the final ordering. In [6], Bar-Joseph *et al.* reports a dynamic programming method that helps to reduce the time and memory complexities for solving the optimal leaf-ordering problem.

## 6 $K$-means clustering

$K$-means clustering [21] is a simple and widely used partitioning method for data analysis. Its helpfulness in discovering groups of coexpressed genes has been demonstrated [49].

The number of clusters $k$ in the data is needed as an input for the algorithm, which then initializes the mean vector for each of the $k$ clusters either by hard assignment (e.g., from the input) or random generation. These initial mean vectors are called the seeds. Next, The iterative procedure of the $k$-means algorithm, consisting of the following two steps, begins. (1) Using the given mean vectors, the algorithm assigns each genes (or experiments) 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 converges when all the mean vectors of the clusters remain stationary.

From the above description, $k$-means method can be understood as a distance-based approach. Both the Pearson correlation and the Euclidean distance can be used as the distance measure. However, the squared Pearson correlation metric should not be used in combination with the $k$-means algorithm, since this distance measure takes the inverse relationships between the clustering subjects into account, which can lead to problems when calculating the means.

On the other hand, $k$-means algorithm is closely related to model-based methods. When the Euclidean distance metric is used, the step of recalculating the means of the clusters actually corresponds to minimizing the within-cluster sum of squared distances from the cluster mean [53]. We will show in Section 11.2 that in this case the $k$-means algorithm is an approximation to the expectation-maximization (EM) method for Gaussian mixture model parameterization [5].

A big problem associated with $k$-means algorithm is the arbitrariness of predefine of the number of clusters, since it is difficult to predict the number of clusters in advance. In practice, this makes it necessary to use a trial-and-error approach where a comparison and biological validation of several runs of the algorithm with different parameter settings are necessary [38]. Another parameter that will influence the result of $k$-means clustering is the choice of the seeds. The algorithm suffers from the problem of local-minimum. This means that with different seeds, the algorithm can yield very different result. Preferably, the seeds should be chosen close to the center of the natural clusters. Of course, this is hard to achieve if no prior knowledge about the clusters is available, which is often the case. Using principle component analysis (PCA) to provide prior knowledge on the number and the means of the clusters was proposed in [39].

# 7 Self-organizing maps

As a machine-learning method, a self-organizing map (SOM) [30] belongs to the category of neural networks. It provides a technique to visualize the high-dimensional input data (in our case, the gene expression data) on an output map of neurons (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 $6 \times 5$ 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) Update the coordinates of a prototype vector based on a weighted sum of all the input vectors that are assigned to it. The weight is given by the neighborhood function (a kernel function in nature), which can be a Gaussian distribution function, applied in the output space. That is, in the updating step, a prototype vector is pulled more toward input vectors that are closer to the prototype vector itself and is less influenced by the input vectors located farther 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) To put a simulated annealing kind of flavor, the initial variance of the Gaussian neighborhood function is chosen so that the neighborhood covers all the neurons, but then the variance is decreased during every iteration so as to achieve a smoother mapping. The algorithm terminates when convergence of the prototype vectors is achieved.

Instead of the batch procedure described above, there is also an online procedure for SOM training. The only difference is, in the online procedure, a random input vector is picked one at a time at the start of the iteration and all the prototype vectors (together with the neurons) are then adjusted accordingly.

From the cluster analysis point of view, SOM methods looks similar to $k$-means methods. SOM clustering differs from $k$-means clustering in that a cluster has two "faces" in an SOM – it is represented by the prototype vector in the input space and the neuron on the output space. In this way, an SOM provides a direct means to visualize relations among different clusters. Moreover, a prototype vector is adjusted according to not only the data points that are associated with it but also data points that are assigned to other prototype vectors. SOM clustering is reported to have satisfactory results on gene expression data [48, 52]. In these experiments, every neuron represents a cluster. As a result, clusters that represents similar gene expression profiles are located closer on the output map, while clusters with anti-correlated expression profiles are put into opposite corners of the grid [45].

Because of the advantage in visualization, choosing the geometry of nodes for an SOM is not as crucial a problem as the choice of the number of clusters for a $k$-means method. Of course, initializing an SOM with too few nodes will result in non-representative and non-distinctive clusters. However, on the contrary, if too many nodes are added to an SOM, clusters with great similarity (located in the same neighborhood on the output map) can be merged to get a more extensive cluster. Based on this idea, a tree-structured SOM clustering method is implemented for gene expression data in [52]. Like the $k$-means method, the initial choice of prototype vectors remains a problem that influence the final clustering result of SOM clustering. A good way to

seed the prototype vectors can be the result from a PCA analysis [30].

# 8 A wish list for clustering algorithms

The limitations of the first-generation algorithms together with the specific characteristics of gene expression data calls out the need of tailored clustering methods 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 wish list of the features that an ideal design of a clustering method for gene expression data should carry.

## 8.1 Lessons from the first-generation algorithms

A problem shared by the first-generation algorithms just described is the decision of the number of clusters in the data. In $k$-means clustering and SOM clustering, this decision has to be made before the algorithms are performed (which makes the problem even harder), while in hierarchical clustering it is postponed till the full dendrogram is formed, where the problem then is where to cut the tree. Choosing the number too small will result in clusters that are too degenerate so that the expression profiles each of them contain might not be related to each other. On the other hand, if the chosen number is too large, coexpressed genes can be separated into different clusters. Algorithms that provide automatic tuning for the number of clusters are preferred, despite the fact that both hierarchical clustering and SOM clustering provides visualization to help the decision-making on this matter, and that various methods exists to select the best choice for the number of clusters once, in the cases of $k$-means clustering and SOM clustering, a range of clustering analysis is done using different number of clusters [37, 51].

Another problem of these first-generation algorithm is that they all assign every gene in the data set (including those constitutive genes) to a particular cluster. A proper filtering step in the preprocessing (see Section 3) can only help to reduce the number of constitutive genes, but it is impossible to remove all of them. Therefore, a desirable clustering algorithm should be able to identify genes that are not relevant for any clusters and leave them as they are.

The third problem concerning these algorithms 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.

## 8.2 Demands defined by the specific characteristics of microarray data

Although a proper preprocessing procedure (see Section 3) can help to moderate the unfavorable characteristics of microarray data, clustering algorithms that are able to handle the specific characteristics of micorarry data (see Section 2) would solve the problem in essence.

First of all, the high dimensionality of microarray data requires the clustering algorithm to have a fast performance. With the development of microarray technology and the decrease in the cost of a microarray chip, the size of data produced by a single microarray study tends to run

up even higher. Thus, a desirable clustering algorithm should at least have a fast performance which can deal with data size up to, say, 100,000 genes by 1000 or more arrays.

Second, the whole data acquisition procedure of microarrays determines the highly noisy nature of the data. Variabilities inherited in biological process, the hybridization, and the image processing all influence the expression levels that are finally read out. Although proper preprocessing procedures help to remove part of the noise presents in the data, the amount of noise that cluster analysis has to face is still substantial. This requires that the clustering algorithms to produce robust results on the highly noisy data. One criterion on the robustness could be done by examining whether the clustering method give the same result on bootstrap replicates.

Third, 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 seems more favorable.

Finally, as we have addressed above, since genes scanned in a microarray experiment does not necessarily have to carry any function in the biological process under study, an ideal clustering algorithm should not sign the outlier genes to any cluster.

### 8.3 The final list

Combining the above analysis, we therefore reach the following list of requirements for an ideal clustering algorithm for gene expression data.

- Fast
- Robust
- Deterministic
- Able to assign a gene to different clusters
- Does not assign outliers to any cluster
- Able to decide the number of clusters automatically

## 9 Self-organizing tree algorithm

The self-organizing tree algorithm (SOTA) [24] combines both self-organizing maps and the divisive hierarchical clustering. Like 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 (contrary to SOM), the tree structure of the nodes grows during the clustering procedure.

With the given tree structure fixed, the gene expression profiles are sequentially and iteratively presented to the terminal nodes (located at the leaves of the tree – these nodes are also called cells). Subsequently, the gene expression profiles are associated with the cell that maps closest to it. The mapping of this cell and its neighboring nodes (including its parent node

and its sister cell) are then updated based on some neighborhood weighting parameters (these parameters perform the same role as the neighborhood function in SOM). Thus, the cells are 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, the cell 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 cell) is split into two sister 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. The determination of the threshold of variability involves the actual construction of a randomized data set – the distribution of distances between all possible pairs in the randomized data is calculated, and a threshold of significant level is applied to this distribution so that the corresponding threshold for the variability is defined.

The approach described in [24] has some properties that make it potentially useful for clustering gene expression profiles. First, the clustering procedure itself is linear in the number of gene expression profiles (compare this with the quadratic complexity of standard hierarchical clustering). Second, the number of clusters does not have to be known in advance. Moreover, the procedure provides a statistical procedure to stop growing the tree. Therefore, the user is freed from choosing an (arbitrary) level where the tree has to be cut (like in standard hierarchical clustering).

In our opinion, however, this method also has some disadvantages. First, the procedure for finding the threshold of variability is time-consuming. The entire process described in [24] is in fact quadratic in the number of gene expression profiles. Furthermore, the algorithm forces every gene (including the constitutive genes) into a cluster.

## 10 Quality-based clustering algorithms

In [25], Heyer *et al.* introduced a quality-based algorithm called QT_Clust. This algorithm produces clusters with a quality guarantee that ensures that all members of a cluster should be coexpressed with all other members of this cluster. The quality guarantee itself is defined, in this paper, as a fixed and user-defined threshold for the maximal distance between two points within a cluster. Later on, starting from the same principle, De Smet *et al.* [46] developed the adaptive quality-based clustering algorithm to circumvent some of the disadvantages of QT_Clust. One of the most convincing improvements is to transform the quality guarantee to a threshold on the significance level for the coexpression of the genes in a cluster, which has a more meaningful interpretation. We now discuss the two methods respectively in detail.

### 10.1 QT_Clust

QT_Clust is a greedy procedure that finds one cluster at a time. For each of the expression profile in the data, QT_Clust explores its neighborhood in the high-dimensional data space to find a candidate cluster that satisfies the quality guarantee while containing as many expression profiles as possible. After this step, the number of candidate clusters is equal to the number of expression profiles in the data set. 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 pre-specified threshold.

This approach was designed with cluster analysis of expression data in mind and has some properties that could make it useful for this task. By using a stringent quality guarantee, it is possible to find clusters with tightly related expression profiles (clusters containing highly coexpressed genes). These clusters might therefore be good "seeds" for further analysis. Moreover, genes not really coexpressed with other members of the data set are not included in any of the clusters.

However, there are some disadvantages for QT_Clust. First, the quality guarantee of the clusters in terms of a user-defined threshold on cluster diameter is hard to estimate and too arbitrary. The method is therefore hard for biologists to use in practise, and extensive parameter fine-tuning is necessary. Second, the algorithm produce clusters all having the same fixed diameter not optimally adapted to the local data structure. Third, the computational complexity is quadratic in the number of expression profiles. Finally, no ready-to-use implementation of the algorithm is available.

### 10.2 Adaptive quality-based cluster

Adaptive quality-based clustering provides improvements for QT_Clust. Adaptive quality-based clustering 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 center is located in an area where the density (i.e., the number) of gene expression profiles is localy maximal. Contrary to QT_Clust, the computational complexity of this first step is only linear in the number of expression profiles. This is done by initializing the cluster radius so that every data point is contained in the cluster, and then iteratively calculating the cluster center and decrease the radius of the cluster until the preliminary estimated radius is reached and the center of the cluster converges.

In the second step, called the adaptive step, given the cluster center, 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 center is fitted to the data using an EM algorithm. Note that the computational complexity of this step is negligible with respect to the computational complexity of the first step. After the probabilistic model is estimated, the radius of the cluster is determined so that expression profiles that are assigned to the cluster have a significance level higher than a user-defined value. In this way, the quality guarantee is transformed to the terms of significance level.

Finally, steps one and two are repeated, using the re-estimation of the radius as the preliminary estimate needed in the first step, until the relative difference between the preliminary and re-estimated quality is sufficiently small. 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.

The adaptive quality-based clustering approach has some additional advantages over QT_Clust that make it suitable for the analysis of gene expression profiles. First, in adaptive quality-based

clustering, the users has to specify a significance level as the threshold for quality control. This parameter has a strict statistical meaning and is therefore much less arbitrary (contrary 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%) 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 radius, adaptive quality-based clustering produces clusters adapted to the local data structure. Third, the computational complexity of the algorithm is linear in the number of expression profiles. Fourth, adaptive quality-based clustering was extensively biologically validated.

However, the method also has some limitations. One of the problems is that this is a heuristic approach not proven to converge in every situation. In addition, because of the model structure used in the second step, some additional constrains have to be imposed. They include the fact that only standardized expression profiles are allowed and that the method has to be used in combination with the Euclidean distance and cannot directly be extended to other distance measures.

## 11 Mixture model for gene expression data

Model-based clustering [21] is an approach that is not really new and has already been used in the past for other applications outside bioinformatics. In this sense it is not really a true second-generation algorithm. However, its potential use for cluster analysis of gene expression profiles has been proposed only recently [59, 20, 35]; thus, we treat it in this text as a second-generation method.

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 mean time, to find out the parameters for each of these distributions.

Usually, a multivariate normal distribution is used to describe the distribution of expression profiles in a same cluster. The choice of the normal distribution is partly based on its desirable analytic convenience . Moreover, the assumption for fitting normal distribution to gene expression profiles is considered to be reasonable especially when the standard preprocessing procedures (see Section 3) have been applied [59, 4]. Of course, other underlying distributions, such as gamma distributions or mixtures of Gaussian and gamma distributions, can also be used to describe expression profiles [58]. So far, no precise conclusions have been made on what is the most suitable distribution for gene expression data [4].

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 refers to both the (observed) microarray data and the assignment of the genes (or experiments) to the clusters. The likelihood of the model increases as the two steps iterates, and convergence is guaranteed.

In the following, we discuss respectively the work of Yeung *et al.* [59] where a mixture model of normal distributions is used for the cluster analysis of microarray data, and the work of McLachlan *et al.* [35] who use a mixture $t$ model for clustering genes and a mixture of factor

analysis for clustering experiments.

### 11.1 Mixture model of normal distributions

When multivariate normal distributions is applied, each cluster is represented by a hypersphere or a hyperellipse in the space which. The mean of the normal distribution gives the center 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 (i.e., the eigenvectors and eigenvalues) of the covariance matrix, one can control the tradeoff between model complexity (the number of parameters one has to estimate) and flexibility (for the model to fit the data).

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 modes 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 [41] for each model, which is not further discussed here.

Yeung *et al.* [59] reported good results of such analysis as described above using their MCLUST software [15] on several synthetic data sets and real expression data sets.

### 11.2 Relation of normal mixture learning with $k$-means approach

The $k$-means algorithm described in Section 6 can be viewed as a special case of applying EM algorithm to learn a mixture model of normal distributions where all the normal distributions in the mixture are characterized by spheres of the same volume but different means (the covariance matrices have the same form of an identity matrix multiplied by a constant). The E step of an EM algorithm is to evaluate the conditional probability of the latent variable of a data point (the latent variable indicates to which cluster the data point belongs) based on the observed data (in our case, the microarray data) and the current parameters. In this context, $k$-means method uses an index function (i.e., a distribution function of zero variance) to replace the conditional distribution and thus performs a hard assignment to put a data point in the cluster whose mean (i.e., center) is the closest. The next iterative step of $k$-means clustering is to minimize the cost function of sum-of-squared-distance within the cluster. Using this index function as the conditional distribution, the minimum of this cost function is exactly the maximum of the log-likelihood function [21]. Thus, this second step of $k$-means corresponds to the M step in the EM algorithm in this case. The parameters from the point of view of the optimization are the sample means of the data points of the clusters.

### 11.3 Mixture model of $t$ distributions and mixture of factor analysis

In [35], McLachlan *et al.* uses a mixture of multivariate $t$ distributions for clustering genes. A $t$ distributions has an additional parameter called the degree of freedom comparing 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 relative 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. EM algorithm is also applied to learn mixture of $t$ distributions for the clustering problem. Because of the thicker tail of a $t$ distribution, the model learnt 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. The disadvantage of using a $t$ mixture, of course, is to calculate the additional parameters in the M step.

For the clustering experiments (e.g., tissue samples), however, problem rises for fitting a normal mixture or a $t$ mixture to the data, because the dimension of the feature space of the multivariate distributions (i.e., the number of genes) is much larger than the dimension of the sample space (the number of experiments). To solve this problem, McLachlan *et al.* applies mixture of factor analysis to the clustering of experiments. The idea can be interpreted as follows. A signal factor analysis performs a dimensional reduction in the feature space of a cluster. That is to say, in factor analysis, vectors of experiments located in the original $n$-dimensional hyperellipse (which represents a cluster and where $n$ represents the number of genes) are projected onto their corresponding vectors of factors located in an $m$-dimensional unit sphere (usually $m << n$). The parameters of the projection (i.e., parameters in the factor loading matrix) explain the correlations between the vectors of experiments, and the residue of the data that is not projected is explained by a covariance matrix (in another word, a hyperellipse centered at the origin) whose orientation is aligned with the axis of the $n$ dimensional space (i.e., this covariance matrix is characterized by the independent variance of each gene) [40]. By using a mixture of factor analysis, clustering of the experiments is done on a reduced feature space (factor space) instead of on the original huge dimensional gene space. The EM algorithm is also used to learn the mixture factor analysis model. However, the EM algorithm for mixture factor analysis is quite time consuming, so the authors of [35] add another step of gene selection before performing the EM algorithm in order to further reduce the number of parameters in the model.

Their software EMMIX-GENE (where these methods are implemented) yields promising results on several biological data sets.

## 12 Biclustering algorithms

The prefix in the word biclustering indicates that this is a technique to cluster both the genes and the experiments at the same time. Although the existing biclustering algorithms originate from the same idea, they differ from each other by their particular emphasis on the problems they try to solve. Among early papers on biclustering methods, clustering algorithms are applied (iteratively) to both dimension of a microarray data set. As the 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, these algorithms divide the data into checkerboard units of patterns. Examples of these algorithms are the work by Alon *et al.* [2] and Getz *et al.* [19], which exploits existing clustering algorithms for the task. In addition, there are also algorithms particularly designed for this purpose. In [32], mixtures of normal distributions, which is called the plaid model by the authors, are used to describe the microarray data and EM is applied for parameter estimation. For another example, the spectral biclustering method [29] applies singular value decomposition for solving the problem.

However, this type of biclustering algorithm has its limitation when the expression profiles

of some genes under study divides the samples in corresponding with one biological explanation (say, tumor type) while profiles of another subset of the genes divides the samples according to another biological process (e.g., drug response) [22]. 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 [22], which searches for clusters of genes that vary as much as possible across the samples with the help of principle component analysis (PCA), and a minimum description length method [26] that identifies gene clusters responsible for classification of experimental conditions.

The third type of biclusteirng 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 behavior, these algorithms look for patterns where genes share similar expressional behavior 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 tumor samples of different histopathological diagnosis. The problem then is to find tumor samples that have similar gene expression level for a subset of genes (so as to obtain an expressional fingerprint for the tumor). 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 algorithms is pioneered by [11], where a heuristic approach is proposed to find patterns as large as possible that have minimum the 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 are also applied for this type of problem. In [7], EM algorithm is used for estimating the parameters of the model, and in [44], a Gibbs sampling strategy for model learning is proposed

The work of Segal *et al.* [43] brings the third type of biclustering problem to a higher level. In their paper, they are interested in identifying not only the relevant experimental conditions for which the relation between genes of a potential group exists but also the significant attributes associated with the genes and the conditions that are responsible for the generation of such patterns. The method incorporates additional information of the attributes (for example, for a gene, the attributes could be functional role, cellular location or the transcriptional factor (TF) binding sites in gene's promoter region, and for a experimental condition, they could be tumor type, or gene knock out information) together with the gene expression data in a probabilistic framework – the probabilistic relational models (PRMs) [17] in particular, which is an extension of Bayesian networks – and uses (structral) EM for the inference for the parameters of the probabilistic models. In [42], the capability of this framework is illustrated in unveiling the regulation programs of the genes from gene expression data.

In the following, we will discuss the gene shaving algorithm [22], the Gibbs sampling biclustering algorithm [44], and the PRMs for microarray data [43, 42] in more detail.

## 12.1 Gene shaving

As we mentioned previously, gene shaving is an algorithm that tries to find a small subset of genes that exhibit the largest variations across the experimental conditions. The intuition is that these genes may help in explaining the classification of the experiments. To search for a gene cluster, the algorithm performs a PCA on the $p$-dimensional space, where $p$ stands for the number of experiments in the data set. The largest principle component, called the eigengene, points to the direction where the cloud of data points (representing the genes) expands with

largest variation. Then the correlation between every gene with this eigengene is calculated. Genes with the smallest (absolute value of) correlation are "shaved off" (discarded). The proportion of the genes to be shaved off is typically 10%. This procedure is carried on until there is only one gene left in the data set. The remaining data set at every step is treated as a candidate gene cluster. To determine the output cluster from these candidate clusters, the ratio of the within-variance versus the between-variance of every candidate cluster is calculated. The within-variance of a cluster measures the variability of each gene about the cluster average, and the between-variance of a cluster examines the variance of a cluster across the experiments. The ratio of a candidate cluster is compared with the average ratio of the clusters (which contain the same number of genes) obtained from randomized data sets. The candidate cluster with the largest difference in the ratio is selected as the output cluster. Then, the expression profiles in the whole data set is orthogonalized with the mean of the expression profiles in the output cluster so as to encourage the discovery of an uncorrelated second cluster. The procedure restarts searching for the next cluster.

Hastie *et al.* [22] also shows that the algorithm can also be extended to a supervised form for the discovery of genes related with particular sample classification. Results in the paper illustrate a successful application of the algorithm on predicting patient survival.

### 12.2 Biclustering by Gibbs sampling

Inspired by the success of Gibbs sampling algorithm in solving the motif-finding problem [31, 33, 50], Sheng *et al.* [44] adapt the Gibbs sampling strategy to work on the biclustering problem of microarray data. The algorithm works only on discretized microarray data sets to make the adaption more straightforward. However, the argument is that the discretization of the original data provides a sensible base of comparison and biological interpretation for biologists when the aim of the algorithm is to identify of gene expressional fingerprints for tissue sample. In addition, the discretization also improves the robustness of the algorithm with regard to the high noise level in the microarray data.

The method first generalizes both the problems of biclustering genes and biclustering experiments to the problem of biclustering rows of a discrete data matrix. The method use a multinomial mixture to describe the data. The multinomial mixture contains a multinomial distribution for the background and a multinomial distribution for each of the columns in the bicluster. A binary random variable (label) is associated 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 a Bayesian approach for the estimation and examines the posterior distribution of the labels given the data. 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.

The Gibbs sampling technique [18, 10] is used to explore the posterior distribution of the labels. Gibbs sampling is a Markov chain Monte Carlo technique to simulate the joint distribution of the random variables of interest when the direct calculation of the joint distribution is difficult or impossible. This is done by sampling iteratively from the conditional distributions of these random variables. For the biclustering problem, the joint distribution of interest is the joint distributions of the labels (which corresponds to the posterior distribution that we mentioned above). Thus, after (random) initialization, the Gibbs biclustering algorithm picks one label at a time and keeps all the other labels fixed to their current value. For the picked label, the

algorithm first calculates the conditional distribution (which is no more than a Bernoulli distribution) given the value of all the other labels (and the data). Once the conditional distribution is obtained, a value (1 or 0) is drawn from the the distribution and is assigned to the label. The procedure iterates till the joint distribution (or all the conditional distributions) converges. The convergence is examined, in the paper, by monitoring the evolution of the likelihood of the data.

To find multiple biclusters in the data, the labels associated to the experiments for a found bicluster 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. The algorithm is able to find biclusters of relatively large size and will stop when no such biclusters can be found in the data (e.g., a bicluster of the size $2 \times 2$ will be neglected).

The paper [44] illustrate the efficiency of the algorithm on both synthetic data sets and a real-life data set, where expressional fingerprints were identified for three subtypes of leukemia.

### 12.3 Probabilistic relational models for microarray data

In the preparation to address the problem in the language of PRMs [17], Segal *et al.* [43] first define the relational schema as follows. The objects – genes, experiments, and gene expressions – are associated with their corresponding attributes. Among other attributes such as the TF binding sites for a gene and the knock-out information for an experiment, the class of a gene is one of the attributes of the gene; similarly, the class of an experiment is an attribute for the experiment; and for the expression of a specific gene in a particular array, the expression level is one of its attributes. The values of some of these attributes are given by the data, for instance, the TF binding sites, the knock-out information, and the expression level. However, the values of the other attributes are unknown. The particular examples of interest in this case are the class attributes of the genes and the experiments. The task now is not only to infer the unknown value of the attributes (such as the class attributes so as to obtain the biclusters) but also to find the relationships between all the attributes presented in the problem.

A PRM itself can be viewed as the bigger frame defining the relationships between classes of different objects. In our case, the frame is that gene expressions are influenced by both the genes and the experiments. However, when it comes to a particular case, where the given data includes the microarray data and accompanied information of the genes and the experiments, the PRM has to generate a more detailed probability model, namely a Bayesian network, for the specific problem. First, to put them in a probabilistic language, all the attributes are described as random variables. A Bayesian network is set up to address the dependency structures between the random variables (i.e., the attributes). A Bayesian network [23] is a graphical model where nodes (each representing a random variable) are connected with each other by directed edges. The direction of an edge between two nodes (pointing from a parent node to a child node) indicates the influence of one random variable (the parent node) to the other (the child node). The influence is quantified by the probabilistic distribution of the child node conditioned on the value of the parent node – the conditional probability distribution (CPD). A node is conditionally independent of all the other nodes given the value of its parents. For the microarray data, one can suppose that the attribute of expression level is always placed at the bottom of the Bayesian network structure; i.e., it is always a child node but never a parent node.

To learn the relationships between different attributes now means to decide the structure

of the Bayesian network (i.e., to draw the edges) and the CPDs between the nodes. In addition, the learning task also includes the estimation of the unknown values (or called missing values) of some of the attributes. The learning procedure is carried out by the Structural EM algorithm [16]. Briefly speaking, it iteratively executes a structural learning step and an EM step till convergence. In the structural learning step, a search algorithm is performed based on the current estimate of network parameters (CPDs) and missing values to find the structure that maximizes a Bayesian information score [41]. The EM step uses the learned structure to estimate its CPDs (the M step) and the missing values (the E step).

The efficiency of the method is illustrated in [43] on two yeast data sets as well as synthetic data sets. In [42], it is shown that the method can be tailored for unveiling the regulatory program of the genes.

## 13 Assessing cluster quality

As mentioned before, clustering will produce different results. Even random data often produce clusters depending on the specific choice of preprocessing, algorithm, and distance measure. 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 (see [49]) is also an appropriate biological validation of the cluster results. Some of the recent methodologies described in literature to validate cluster results will be highlighted in the following.

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 (maximal inter-cluster distance). Such criteria can be formalized in several ways, such as the sum-of-squares criterion of $k$-means [53], silhouette coefficients [27], or Dunn's validity index [3]. These can be used as stand alone statistics to mutually compare cluster results. They can also be used as an inherent part of cluster algorithms, if their value is optimized during the clustering process.

2. *Figure of Merit:* FOM [60] is a simple quantitative data-driven methodology that allows comparisons between outputs of different clustering algorithms. The methodology is related to the jackknife and leave-one-out cross-validation. The method goes as follows. 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, we compute the FOM for a clustering result by summing, for the left-out condition, the squares of the deviations of each gene relative to the mean of

the genes in its cluster for this condition. The FOM measures the within-cluster similarity of the expression values of the removed experiment and therefore reflects the predictive power of the clustering. It is expected that removing one experiment from the data should not interfere with the cluster output if the output is robust. For cluster validation, each condition is subsequently used as a validation condition, and the aggregate FOM over all conditions is used to compare cluster 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 (similar to the experimental noise in the data) 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. In this case, the cluster membership of that particular gene is robust with respect to sensitivity analysis, and a reliable confidence can be assigned to the clustering result of that gene. However, for genes with low signal-to-noise ratios, the outcome of the clustering result will be more sensitive to adding artificial noise. Through some robustness statistic [9], 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. [9] perturb 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. This implicitly assumes that ratios are unbiased estimators of relative expression, yet reality shows often otherwise.

   The bootstrap analysis methods described by Kerr and Churchill [28] to identify statistically significant expressed genes or to assess the reliability of a clustering result offers a more statistically founded basis for sensitivity analysis and overcomes some of the problems of the method described by Bittner et al. [9]. Bootstrap analysis uses the residual values of a linear analysis of variance (ANOVA) model as an estimate of the measurement error. By using an ANOVA model, nonconsistent measurement errors can be separated from variations caused by alterations in relative expression or by consistent variations in the data set. These errors are assumed to be independent with mean zero and constant variance $\sigma^2$ but no explicit assumption on their distribution is made. 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 that of Bittner et al. [9] are used to perform these comparisons.

   Biologists tend to prefer algorithms with a deterministic output, since this gives the illusion that what they find is "right". However, nondeterministic algorithms offer an advantage for cluster validation, since their use implicitly includes a form of sensitivity analysis.

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 [20, 49] representing their biochemical properties, biological roles, and so on. Finding clusters that have been significantly enriched for genes with similar function is proof that a specific clustering technique produces biologically relevant results.

   In this regard, the data set described in Cho et al. [12] that studies the cell development cycle of yeast in a synchronized culture is often used as a benchmark data set. It contains 6220 expression profiles taken over 17 time points (measurements over 10-min intervals, covering nearly two cell cycles; see also `http://cellcycle-www.stanford.edu`). One of the reasons that this data is so frequently used as benchmark data for the validation of new clustering algorithms is because of the striking cyclic expression patterns and because the majority of the genes included in the data have been functionally classified [36] (MIPS database; see `http://mips.gsf.de/proj/yeast/catalogues/funcat/index.html`), making it possible to biologically validate the results.

   Assume that a certain clustering method finds a set of clusters in the Cho et al. data [12]. We could objectively look for functionally enriched clusters as follows: Suppose that one of the clusters has $n$ genes where $k$ genes belong to a certain functional category in the MIPS database, and suppose that this functional category in its turn contains $f$ genes in total. Also suppose that the total data set contains $g$ genes (in the case of Cho et al. [12], $g$ would be 6220). Using the cumulative hypergeometric probability distribution, we could 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. Since there are about 200 functional categories in the MIPS database, only clusters where the $P$-value is smaller than 0.0003 for a certain functional category, are said to be significantly enriched (level of significance 0.05). Note that these $P$-values can also be used to compare the results from functionally matching clusters identified by two different clustering algorithms on the same data.

## References


[1] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson Jr., L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, and J. O. Armitage. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[2] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. 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, 1999.

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

[4] P. Baldi and S. Brunak. *Bioinformatics: the Machine Learning Approach*. Adaptive computation and machine learning. The MIT Press, second edition edition, 2001.

[5] P. Baldi and G. W. Hatfield. *DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling*. Cambridge University Press, 2002.

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

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

[8] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *J. Comput. Biol.*, 6:281–297, 1999.

[9] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Doughety, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, V. Sondak, N. Hayward, and J. Trent. Molecular classfication of cutaneous malignat melanoma by gene expression profiling. *Nature*, 406:536–540, 2000.

[10] G. Casella and E. I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[11] Y. Cheng and G. M. Church. Biclustering of expression data. In *ISMB 2000 proceedings*, pages 93–103, 2000.

[12] R. J. Cho, M. J. Campbell, E. A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. G. Wolfsberg, A. E. Gabrielian, D. Landsman, D. J. Lockhart, and R. W. Davis. A genome-wide transcriptional analysis of mitotic cell cycle. *Molecular Cell*, 2:65–73, 1998.

[13] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Willey & Sons, Inc., second edition edition, 2001.

[14] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 95:14863–14868, 1998.

[15] C. Fraley and E. Raftery. MCLUST: software for model-based cluster analysis. *Journal of Classification*, 16:297–306, 1999.

[16] N. Friedman. The bayesian structural EM algorithm. In *Proceedings of UAI'98*, 1998.

[17] N. Friedman, L. Gettor, D. Koller, and A. Preffer. Learning probabilistic relational models. In *Proceedings of IJCAI'99*, 1999.

[18] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 7:721–741, 1984.

[19] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad. Sci. USA*, 97(22):12079–12084, 2000.

[20] D. Ghosh and A. M. Chinnaiyan. Mixture modelling of gene expression data from microarray experiments. *Bioinformatics*, 18(2):275–286, 2002.

[21] J. A. Hartigan. *Clustering Algorithms*. Wiley Series i Probability. John Wiley & Sons, Inc., 1975.

[22] T. Hastie, R. Tibshirani, M. B. Eisen, A. Alizadeh, R. Levy, L. Staudt, W. C. Chan, D. Botstein, and P. Brown. 'gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, 1(2):research0003.1–0003.21, 2000.

[23] D. Heckerman. *Learning in Graphical Models*, chapter A Tutorial on learning with Bayesian networks, pages 301–354. The MIT Press, 1999.

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

[25] L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research*, 9:1106–1115, 1999.

[26] R. Jörsten and Bin Yu. Simultaneous gene clustering and subset selection for sample classification via MDL. *Bioinformatics*, 19(9):1100–1109, 2003.

[27] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: an Introcuction to Cluster Analysis*. John Wiley & Sons, Inc., 1990.

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

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

[30] T. Kohonen. *Self-Organizing Maps*. Springer Series in Information Sciences. Springer, 1995.

[31] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wooton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignments. *Science*, 262:208–214, 1993.

[32] L. Lazzeroni and A. Owen. Plaid models for gene expression data. Technical report, Department of Statistics, Standford University, 2000.

[33] J. S. Liu, A. F. Neuwald, and C. E. Lawrence. Bayesian models for multiple local sequece alignment and Gibbs sampling strategies. *J. Amer. Stat.*, 90:1156–1170, 1995.

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

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

[36] H. W. Mewes, D. Frishman, U. Güldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenster, M. Münsterkötter, S. Rudd, and B. Weil. MIPS: a database for genenomes and protein sequences. *Nucleic Acids Res.*, 30(1):31–34, 2002.

[37] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50:159–179, 1985.

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

[39] J. Quackenbush. Computational analysis of microarray data. *Nature Reviews*, 2:418–427, 2001.

[40] S. Roweiss and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11(2), 1999.

[41] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[42] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module networks: Identifying regulatory modules and their condition-specific regulators for gene expression data. *Nature Genetics*, 34(2):166–176, 2003.

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

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

[45] G. Sherlock. Analysis of large-scale gene expression data. *Current Opinion in Immunology*, 12:201–205, 2000.

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

[47] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehesive identification of cell cycle-regulated genes of the yeast *s*accaromyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.

[48] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA*, 96:2907–2912, 1999.

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

[50] G. Thijs, K. Marchal, M. Lescot, S. Rombauts, B. De Moor, P. Rouzé, and Y. Moreau. A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *J. Comput. Biol.*, 9:447–464, 2002.

[51] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the Gap statistic. Technical report, Department of statistics, Stanford University, March 2000.

[52] P. Törönen, M. Kolehmainen, G. Wong, and E. Gastrén. Analysis of gene expression data using self-organizing maps. *FEBS Letters*, 451:142–146, 1999.

[53] J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles*. Applied mathematics and computaion. Addison-Wesley Publishing Company, 1979.

[54] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[55] J. H. Ward. Hierarchical grouping to optimize an objective function. *Jour. Amer. Stat. Assoc.*, 58:239–244, 1963.

[56] J. N. Weinstein, T. G Myers, P. M. O'Connor, S. H. Friend, A. J. Fornace, Jr. K. W. Kohn, T. Fojo, S. E. Bates, L. V. Rubinstein, N. L. Anderson, J. K. Buolamwini, W. W. van Osdol, A. P. Monks, D. A. Scudiero, E. A. Sausville, D. W. Zaharevitz, B. Bunow, V. N. Viswanadhan, G. S. Johnson, R. E. Wittes, and K. D. Paull. An informative-intensive approach to the molecular pharmacology of cancer. *Science*, 275:343–349, 1997.

[57] X. Wen, S. Fuhrman, G. S. Michaels, D. B. Carr, S. Smith, J. L. Bakker, and R. Somogyi. Large-scale temporal gene expression mapping of central nervous system development. *Proc. Natl. Acad. Sci. USA*, 95:334–339, 1998.

[58] B. L. Wiens. When log-normal and gamma models give different results: a case study. *The American Statistician*, 53:89–93, 1999.

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

[60] K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17(4):309–318, 2001.