Evaluation of the Contents of Partitions Obtained with Clustering Gene Expression Data

Katti Faceli[1], André C.P.L.F. de Carvalho[1], and Marcílio C.P. de Souto[2]

[1] Universidade de São Paulo,
Instituto de Ciências Matemáticas e de Computação,
Departamento de Ciências de Computação e Estatística,
Caixa Postal 668, 13560-970 - São Carlos, SP, Brasil
{katti, andre}@icmc.usp.br
[2] Universidade Federal do Rio Grande do Norte,
Departamento de Informática e Matemática Aplicada - DIMAp
Campus Universitario, 59072-970 - Natal, RN, Brazil
marcilio@dimap.ufrn.br

Abstract. This work investigates the behavior of two different clustering algorithms, with two proximity measures, in terms of the contents of the partitions obtained with them. An analysis of how the classes are separated by these algorithms, as different numbers of clusters are generated, is also presented. A discussion on the use of these information in the identification of special cases for further analysis by biologists is presented.

1 Introduction

Nowadays, gene expression data consists of an important source of information for the understanding of biological processes and diseases mechanisms. Clustering methods are one of the most important tools to support biologists in the analysis of gene expression data. As pointed out by [1], this type of analysis is of increasing interest in the field of functional genomics and gene expression data analysis. One of its motivation is the need for molecular-based refinement of broadly defined biological classes, with implications in cancer diagnosis, prognosis and treatment [1].

There is a huge diversity of clustering techniques described in the literature. Some of them have been employed to gene expression data. Examples are k-means [2], Self-Organizing Maps (SOM) [2], Self-Organizing Tree Algorithm (SOTA) [3] and the hierarchical clustering algorithms [2]. In this paper, k-means and SOTA, with both the Euclidean distance and Pearson correlation, are employed to generate a set of partitions (clusterings). Based on the partitions generated, two types of analysis are developed. First, a high level evaluation and comparison of the quality of the partitions are accomplished. For such, two different validation approaches are used: external validation employing the corrected Rand index [4] and the analysis of the variability of the algorithms by bootstrapping.

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 65–76, 2005.

The second type of analysis, which is the main focus of this work, is a finer study of the partitions obtained. More precisely, the best partitions according to the evaluation process from the first step have their contents analyzed in detail. A further analysis of the contents of each cluster in the partitions can bring important insights to the biologists. For example, this analysis can show patterns (samples or genes) that have a different behavior from that expected. These patterns could represent interesting cases to have a detailed investigation in the laboratory.

Furthermore, the analysis of partitions with different numbers of clusters can help in the identification of new subgroups in the data, when main groups are already known, as in the case of cancer classes. This can lead to the discovery of new classes, or subclasses, of cancer. The discovery of new classes of cancer is an issue that has received strong attention recently. Other possible contributions to biologists are discussed in Sect. 5.

2 Experiments

The experiments were carried out by applying two clustering algorithms, k-means and SOTA, to the dataset St. Jude leukemia [5, 1]. This dataset has a multi-class distinction (a phenotype) that will be considered as the gold standard partition, referred also as the true partition of the dataset. Following the conversion used in [1], the groups stated by the gold standard partition are referred as classes, while the notation cluster is reserved for the groups returned by the clustering algorithms.

For the detailed analysis described in Sect. 4, the class label associated to each pattern should be known, otherwise the coloring scheme cannot be applied.

This dataset consists of 248 diagnostic bone marrow samples from pediatric acute leukemia patients corresponding to six prognostically important leukemia subtypes. Each sample is composed of the expression values of 985 genes. Table 1 shows the classes and the number of patterns (samples) of each class present in the dataset. For short, the notation in parenthesis will be employed in the text, when it is the case. In the experiments, the samples were the patterns to be clustered and the genes were their attributes.

Table 1. Classes present in the dataset

Class	Number of patterns
BCR-ABL (BCR)	15
E2A-PBX1 (E2A)	27
'hyperdiploid>50' (hyperdip)	64
MLL	20
T-lineage ALL (T-ALL)	43
TEL-AML1 (TEL)	79

The experiments consisted of the generation of partitions having from 2 to 15 clusters, employing k-means and SOTA algorithms with the Euclidean and Pearson proximity measures. This range was chosen because the true number of clusters is six and having as a reference the work in [1], which also investigated such a range for this dataset.

K-means is one of the most traditional clustering algorithms [4]. It is a partitional algorithm that partition the dataset in a predefined number of clusters. In this work, k-means has been chosen as a reference, since it is widely employed in a number of applications, including gene expression analysis. In contrast to partitional features of the k-means, SOTA is a hierarchical divisive algorithm, based in the neural networks Self Organizing Maps (SOM) and Growing Cell Structures (GCS). It is a neural network that grows adopting the topology of a binary tree. Some of the main characteristics of this algorithm, desirable for gene expression data analysis, are its ability in dealing with high-dimensional data, scalability, robustness against noise and outliers and independence from the order of data presentation.

The experiments carried out with SOTA employed default values for the parameters, except for the maximum number of cycles (max). This parameter determines the number of clusters to be generated ($max + 1$ clusters). The value of max varied from 1 to 14 (2 to 15 clusters). Although SOTA can automatically determine the best number of clusters, the authors forced the algorithm to generate the partitions with the specific numbers of clusters that were being studied. The other parameters of SOTA are the variability and resource thresholds, that define the convergence of the network (default value of 0 for both parameters), the relative error threshold, that defines the convergence of a cycle (default value of 0.0001) and the actualization factors for the winning, mother and sister nodes (default values of 0). Other values for these parameters were not investigated, since the interest were not in the best adjustment of SOTA, but in the comparison among different numbers of clusters in different algorithms and similarity measures. For k-means, the only parameter of the algorithm is the number of clusters, that was varied from 2 to 15.

The algorithm k-means generate different partitions for the same dataset and number of clusters, depending on the random initialization of the centroids. SOTA generates the same partition for a specified number of clusters and just breaks the clusters as a higher number of clusters is specified.

The performance of a clustering method for gene expression data analysis depends on the employment of an appropriate proximity function, according to the properties the researcher wants to focus. As the interest of the authors are in looking for all potentially interesting groups in a dataset, two different proximity measures commonly employed to gene expression data clustering were employed: Euclidean distance and Pearson coefficient [2]. The Euclidean distance measures the absolute distance between two points in an n-dimensional space. According to this metric, similar patterns exhibits the same magnitude and direction. The Pearson correlation coefficient (linear correlation) measures the

angular separation of the patterns around their mean. This metric is usually described as a measure of the shape, as it is insensitive to differences in the magnitude of the attributes.

In the following sections, the experiments will be represented by three components. The first one is a letter representing the algorithm: K for k-means and S for SOTA. The second component is also a letter representing the proximity measure employed: E for Euclidean distance and P for Pearson correlation. The last component is the number of clusters generated. For example, the experiment employing the k-means and the Euclidean distance, generating six clusters will be represented by KE6.

3 High Level Evaluation

In this paper, the validation of the results was accomplished by means of two different approaches: external validation employing the corrected Rand index [4, 6] and the analysis of the variability of the algorithms by bootstrapping [7]. The first approach aims to assess how good the clustering techniques investigated are at recovering known clusters. This was performed by using the corrected Rand index (CR for short). In this context, the authors also checked if the partitions generated are valid. A partition can be considered valid, for example, if the value of its CR index is unusually high, according to a reference distribution [4]. In order to do so, the authors followed the procedure described in [6], but employing bootstrap samples as if they were a replication of a Monte Carlo experiment [4]. The number of bootstrap samples, B, considered in this paper was set to 100.

CR measures the agreement between the true partition (the gold standard) and the clustering generated by an algorithm. It can take values from -1 to 1, with 1 indicating a perfect agreement between the partitions, and the negatives or near 0 values corresponding to cluster agreements found by chance.

The other validation approach employed in this paper also uses bootstrapping, but to analyze the variability of each clustering algorithm [7]. The

Table 2. Variablilty and Corrected Rand for the best partitions

Five Best CR			Five Best V_{adj}		
Partition	V_{adj}	CR	Partition	V_{adj}	CR
KP5	0.254802	0.852346	SP3	0.181361	0.287574
KE4	0.260567	0.829675	SP4	0.186054	0.255003
KP6	0.267859	0.829643	KP2	0.191907	0.217778
KP5	0.234667	0.805082	SE3	0.194405	0.380157
SP11	0.204392	0.796235	SE4	0.198795	0.340644

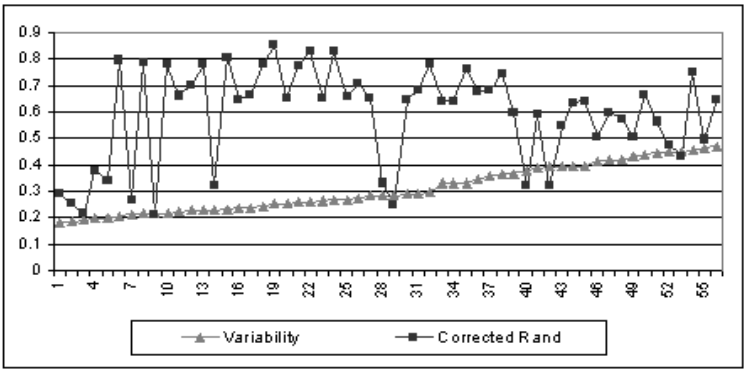

Fig. 1. Variability and Corrected Rand for all partitions generated

variability can be used, for instance, to compare partitions produced by different algorithms, by an algorithm with different parameters values or by an algorithm employing different proximity measures. Such approach sees a clustering algorithm as a point estimator (as in statistical theory) for the partition of the data space and uses bootstrapping to estimate the variability of the estimator. In this context, if the partition is valid, the variability should be low. In order to apply this validation, $B = 100$ bootstrap samples were also generated. The algorithm was run on each sample obtaining a set of partitions. The variability, V, was estimated using CR to calculate the distance between two partitions. Afterwards, 100 random partitions were generated and the variability on them, V_{ran} was also calculated. Finally, the adjusted variability V_{adj} was calculated by $V_{adj} = V/V_{ran}$. V_{adj} is the variability value employed to compare the partitions in the analysis that follows.

Each validation strategy employed led to different best partitions. The five best results according to each strategy are shown in Table 2 - all partitions obtained in the experiments with the external validation employing the corrected Rand index were found to be valid with a significance level of 0.05. Figure 1 is a plot of the values of CR and variability for all partitions obtained, ordered by their variability. Some interesting observations can be made from Table 2 and Fig. 1. Partitions presenting the lowest (best) variabilities show very poor quality according to CR. Variability favors small number of clusters. On the other hand, the best partitions according to CR were obtained for numbers of clusters close to the true number of clusters, six. The partitions presenting high CR values show variabilities slightly above the best variability values obtained. It can be observed that for the 6th to 38th variability values shown in the graphic, most of the corresponding values of corrected Rand lied above 0.6. It was observed that k-means presented the best results according to CR and SOTA showed the best results according to the variability.

4 Partitions Evaluation

This section takes a closer look at the partitions produced in order to evaluate the composition of the clusters and the influence of the clustering algorithms and the number of clusters used. This analysis also considers the true known structure of the dataset (classes).

In order to facilitate the analysis, a coloring scheme was applied to each partition. The first step in the procedure of coloring the partitions was the assignment of a color to each class. Next, the number of patterns from each class present in each cluster for a given partition was determined. Based on this information, the predominant class for each cluster was found (the class that presents more patterns in the cluster). Next, each cluster was labeled with the color of its predominant class. An intensity was also assigned to each cluster, aiming to distinguish the clusters with the same predominant class. An intensity of 0 was assigned to the cluster with the highest number of patterns from the predominant class, an intensity of 1 was assigned to that with the second highest number of patterns of the predominant class, and so on.

With the clusters colored, the partitions to be compared were plotted side by side in a datasheet, with all partitions ordered by the pattern identifier. It should be noticed that the pattern identifier has an indication of the class to which the pattern belongs. Otherwise, an indication of the class should be added to the identifier. This representation associated with the coloring scheme make it possible to readily distinguish the patterns wrongly assigned to a cluster and the most homogeneous clusters.

For a preliminary analysis, the three best partitions of each validation strategy described (Sect. 3) were selected. As the 5th best partition, according to CR, was the 6th best partition, according to the variability, this partition was also selected (SP11). This first analysis originated a question: What does it happen with partitions with a higher number of clusters? Is it possible that a partition with a number of clusters much higher than the true number presents good clusters, together with clusters of poor quality? To check this possibility, the partitions with 15 clusters generated with both algorithms and proximity measures investigated were analyzed. Another issue investigated with 15 clusters was the existence of problematic patterns that can interfere with the clustering result. With a higher number of clusters, these patterns could be isolated, so that the other patterns could be grouped into more homogeneous clusters.

From the coloring scheme and the observation of the clusters contents, useful information was obtained, which is summarized in this section. Table 3 details the amount of patterns from each class in the clusters from the best partition according to CR (KP5), the best partition according to variability (SP3), the partition SP11, described previously and the partition SP15, that present the best CR value among the partitions with 15 clusters.

Table 4 contains a summary of the clusters generated in each experiment considered. The clusters were classified into four types: pure clusters (P), large well defined clusters (LWD), large mixed clusters (LM) and small mixed clusters (SM). The pure clusters contain patterns of only one class. Clusters with one

Table 3. Clusters contents for the best clustering

Partition	Cluster	Class					
		BCR	E2A	hyperdip	MLL	T-ALL	TEL
KP5	1						78
	5	13		62			1
	4					39	
	2	1	27		3		
	3	1		2	16	4	
SP3	3	14	23	62	4		79
	1					39	
	2	1	4	2	16	4	
SP11	10						79
	9	14	1	62	1		
	4					19	
	3					12	
	1					4	
	2					4	
	5					3	
	11		22		3		
	8			2	14	1	
	6	1					
	7		4		2		
SP15	14						79
	13	14	1	62	1		
	3					8	
	6					8	
	8					7	
	1					4	
	2					4	
	7					4	
	4					3	
	9					3	
	5					1	
	15		22		3		
	11		4		2		
	12			2	14	1	
	10	1					

single pattern are also considered pure. Large well defined clusters have the majority of the patterns from the predominant class and just few patterns from other classes. Large mixed clusters have the majority of the patterns from 2 or more classes. Small mixed clusters contain few patterns from more than one class. The table included the number of each type of cluster and, when appropriate, the predominant class of each cluster (in the case of LMC, the classes with a large number of patterns in the cluster).

Table 4. Main structure of the clusters of each clustering

Partition	P	LWD	LM	SM
KP5	2 (T-ALL, TEL)	2 (MLL, E2A)	1 (hyperdip+BCR)	0
KE4	1 (T-ALL)	1 (TEL)	2 (hyperdip+BCR, E2A+MLL)	0 0
KP6	2 (T-ALL, TEL)	2 (MLL, E2A)	1 (hyperdip+BCR)	1
SP3	1 (T-ALL)	1 (MLL)	1 (hyperdip+BCR+ E2A+TEL)	0 0
SP4	2 (2 T-ALL)	1 (MLL)	1 (hyperdip+BCR+ E2A+TEL)	0 0
KP2	0	1 (T-ALL)	1 (hyperdip+BCR+ E2A+TEL+MLL)	0 0
SP11	7 (5 T-ALL, BCR, TEL)	2 (MLL, E2A)	1 (hyperdip+BCR)	1
KE15	9 (5 T-ALL, hyperdip, BCR, MLL, TEL)	5 (2 E2A, BCR, 2 TEL)	0	1
KP15	8 (3 T-ALL, 3 TEL, E2A, hyperdip)	3 (MLL, E2A, hyperdip)	1 (hyperdip+BCR)	3
SE15	9 (8 T-ALL, hyperdip)	6 (MLL, E2A, TEL, hyperdip, 2 BCR)	0	0
SP15	11 (9 T-ALL, BCR, TEL)	2 (MLL, E2A)	1 (hyperdip+BCR)	1

Table 5 shows the number of patterns assigned to a large cluster of another class (wrong assignment), the number of patterns assigned to the small mixed clusters and the number of patterns assigned to small pure clusters (with less than 5 patterns in the cluster), in each clustering analyzed. The patterns in the pure and small mixed clusters are better seen by looking at the clusters composition in Table 3.

Some conclusions can be drawn from the analysis of these data. First, patterns from each class were represented mostly with the same color, but in some cases with different intensities. This means that, even when the patterns from a class were separated in different clusters, they usually were assigned to clusters with the same predominant class. This was also true in the analysis of the partitions with 15 clusters (the highest number of clusters investigated). Even for the partitions with fewer clusters, most of the patterns of each class tended to appear together in the same cluster, even when the clusters were composed of different classes (LM). These were the cases of KP4 and KP2, which presented few wrong assignments due to the large mixed clusters that placed most of the patterns from several classes together.

The best partition according to CR (KP5) generated two pure clusters, two well defined clusters and one mixed cluster (BCR + hyperdip). This is a good partition, but it did not separate the classes hyperdip and BCR. Looking at the partitions of 15 clusters, most of them can separate all classes, including BCR and hyperdip (KE15, KP15 and SE15). The partition KE15 did not generate

large mixed clusters and generated just one small mixed cluster, with just four patters. Although the number of clusters was large, the clusters obtained were homogeneous. This partition can also be considered a good partition, in spite of its relatively low value of CR and high variability.

Almost all patterns from the class TEL always grouped together. There were just few cases in some of the partitions where a pattern from this class was associated to another cluster. The TEL patterns also appeared well separated from the other classes, except for the cases where few clusters were generated. The patterns from the class T-ALL formed a well separated cluster too. The algorithm SOTA tended to divide the patterns from the class T-ALL in several small sub-clusters before separating the patterns of the classes TEL, hyperdip-BCR and E2A. This was observed by looking at the clusters of T-ALL for the partitions of three and four clusters generated by SOTA with the Pearson correlation. This trend was confirmed by the analysis of the partitions with 15 clusters, where eight or nine small pure clusters of the class T-ALL were formed.

The patterns from the classes BCR and hyperdip were almost always grouped together in the same cluster. Even when there were clusters with the predominant class BCR and clusters with the predominant class hyperdip, most of these clusters still presented patterns from both classes (BCR or hyperdip). As there are 6 classes, the best solution of 6 clusters found (generated with the algorithm k-means with Pearson - KP6) was analyzed with more attention to compare the clusters with the true classes. This partition did not separate the patterns from the classes BCR and hyperdip, as the other partitions containing a smaller number of clusters. This partition presented a large cluster with most of the hyperdip and BCR patterns and a small cluster containing the other few hyperdip and BCR patterns together with patterns from three other classes The other clusters were similar to those obtained using a smaller number of clusters.

A question arises from the observation of the result obtained in the partition with 6 clusters, KP6. Does the generation of two clusters mixing BCR and hyperdip can indicate that if more clusters were generated, this classes could be separated? It was observed that when a large number of clusters were generated (11 or 15), small pure cluster started to appear. Also, in the analysis of the partitions with 15 clusters, pure clusters of hyperdip, clusters with almost all patterns belonging to this class and clusters with some hyperdip samples, but with the majority of the patterns belonging to the BCR class were found. Such results confirmed that, although the classes hyperdip and BCR are very similar, they have differences that can be found in some way (in this case, generating a higher number of clusters). This observations were valid for both algorithms, SOTA, which generated the same partition for a specified number of clusters, and k-means, which generated a different partition in each run. It should also be observed that the best partition of 15 clusters, according to CR, did not separate the classes BCR and hyperdip, as the other partitions with 15 clusters do. Both partitions of 15 clusters obtained with k-means presented several pure clusters. In the case of SOTA, the class T-ALL was divided into several small

pure clusters. Three of these four partitions with 15 clusters separated the class BCR from hyperdip.

Some heterogeneous clusters very similar in many of the partitions analyzed were found. One of this clusters was composed of the majority of the hyperdip and BCR patterns. Another similar case was the cluster composed of 16 MLL patterns, one BCR, two hyperdip and a few other patterns of other classes. This can indicate that the patterns wrongly assigned to this clusters found in all cases are really more similar to the patterns in this clusters than to those of their class, and that the wrong assignments did not occur just because of the variability of the algorithms. Maybe these patterns were either wrongly labeled or contained important information to be investigated, as they should be more similar to patterns from their class.

Other observation is that there were some patterns that were assigned to the same wrong cluster in most of the partitions analyzed. This is the case for the patterns "hyperdip.50.7" and "hyperdip.50.C19", almost always assigned to clusters with the predominant class MLL. Other patterns were also assigned to a wrong cluster, but in only one or two partitions. Table 6 shows the patterns wrongly assigned to at least five partitions. In this table, for each pattern, only the columns of the partitions in which a wrong assignment occurred are marked. This "marking" is made with the predominant class of the cluster to which the pattern was wrongly assigned. For example, the pattern "BCR.ABL.R1" was wrongly assigned to the class E2A in the partition KE4 and to the class MLL in the partitions KP5, KP6, SP3, SP4 and SE15. These wrongly assigned patterns were easily identified with the coloring scheme as they were shown with a different color from the majority of the other patterns from the same class. It should be noticed that for the clusters that encompassed more than one class (the majority of the patterns from more than one class), the patterns from the classes well represented in the clusters were not considered wrong assignment. Thus, for example, in the cluster composed of hyperdip and BCR, neither BCR

Table 5. Number of patterns in each type of cluster

Partition	Wrong assignments	Small mixed	Small pure
KP5	13	0	0
KE4	5	0	0
KP6	13	13	0
SP3	15	0	0
SP4	15	0	0
KP2	7	0	0
SP11	8	6	12
KE15	18	4	6
KP15	16	16	0
SE15	21	0	13
SP15	8	6	20

Table 6. Assignments to wrong clusters in at least 2 clusterings

Pattern	KP5	KE4	KP6	SP3	SP4	KP2	SP11	KE15	KP15	SE15	SP15
BCR.ABL.R1	MLL	E2A	MLL	MLL	MLL					MLL	
Hyperdip.50.7	MLL	E2A	MLL	MLL	MLL	T-ALL	MLL	MLL		MLL	MLL
Hyperdip.50.C19	MLL	E2A	MLL	MLL	MLL		MLL	MLL		MLL	MLL
MLL.C3	E2A		E2A	TEL	TEL		hyp	E2A	E2A	E2A	hyp
MLL.C4	E2A		E2A	TEL	TEL		E2A	E2A	E2A	TEL	E2A
MLL.C5	E2A		hyp	TEL	TEL		E2A	E2A		hyp	E2A
MLL.C6	E2A		E2A	TEL	TEL		E2A	E2A	E2A	TEL	E2A
T.ALL.C5	MLL		MLL	MLL			MLL			MLL	MLL

nor hyperdip patterns were considered wrongly assigned. An assignment was considered an error only if the pattern was assigned to a cluster with only few or no other patterns from its class. The small mixed clusters were not considered wrong assignments either.

Different wrong assignments of a pattern can be due to mixed clusters as they encompasses a large amount of patterns of more than one class and a cluster has a predominant class when this class has more patterns in the cluster than the other classes. For example, "BCR.ABL.R1" was assigned to a cluster of the class E2A in the partition KE4 and to clusters of the class MLL in all other partitions where a wrong assignment occurred. Even when assigned to the cluster from the class E2A, "BCR.ABL.R1" was assigned to a cluster with many MLL patterns, as the cluster E2A is a large mixed cluster of E2A and MLL.

5 Conclusion

This paper investigated two different clustering algorithms and two proximity measures to obtain a series of partitions of a gene expression dataset. For each algorithm and proximity measure, partitions containing from 2 to 15 clusters were generated. Each validation strategy pointed out a different technique as superior. The k-means algorithm presented better results according to CR and SOTA according to variability. The best partitions obtained had their contents analyzed in details.

The analysis carried out in this work can provide useful insights to the area of gene expression analysis. The information outlined can be used to point out new directions for further analysis by biologists. The large mixed clusters can indicate unexpected similarities of the classes. The subdivisions of the classes in smaller clusters can indicate possible important subdivisions of the data, supporting the discovery of new disease subtypes (such as those of great interest in cancer research). The small heterogeneous clusters can have important meaning as they present patterns with different behavior from that expected. They could represent interesting samples that could be further analyzed in laboratory. The samples that were always classified in the same wrong cluster can be either just noisy samples, or can indicate an error in the original classification of these

samples. Alternatively, they can occur because these samples really present an unexpected behavior, which may be worth of additional investigation.

Additional experiments are being carried out using other datasets. The results so far have confirmed the potential of the proposed approach. Other clustering algorithms are also being included. The authors also intend to have the support of biologists to identify the true contribution to gene expression data analysis, mainly in the discovery of new subclasses of the data. As a result, more general conclusions can be obtained. Future work includes the application of the same analysis performed in this paper, but comparing all partitions generated with all the different numbers of clusters investigated. The goals are to better analyze the isolation of problematic patterns and their influence in the good separation of the clusters and to investigate the identification of new subclasses in the data.

Acknowledgments

The authors would like to thank financial support of FAPESP and CNPq.

References

1. Monti, E., Tamayo, P., Mesirov, J., Golub, T.: Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning, 52 (2003) 91–118
2. Jiang, D., Tang, C., Zhang, A.: Cluster Analysis for Gene Expression Data: A Survey. IEEE Trans. Knowl. Data Eng. 16(11) (2004) 1370–1386
3. Herrero,J., Valencia,A., Dopazo,J.: A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics, 17(2) (2001) 126–136
4. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice Hall (1988)
5. Yeoh, E. J., et al.: Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. Cancer Cell., 1(2) (2002) 133–143
6. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Cluster validity methods: Part I. SIGMOD Record 31 (2) (2002) 40–45
7. Law, M. H., Jain, A. K.: Cluster validity by bootstrapping partitions. TR MSU-CSE-03-5, Dept. Comp. Science and Eng., Michigan State University (2003)