Genetics and Molecular Biology, 27, 4, 623-631 (2004)
Copyright by the Brazilian Society of Genetics. Printed in Brazil
www.sbg.org.br

Research Article

Comparative analysis of clustering methods for gene expression time course data

Ivan G. Costa[1], Francisco de A.T. de Carvalho[1] and Marcílio C.P. de Souto[2]

[1] *Universidade Federal de Pernambuco, Centro de Informática, Recife, PE, Brazil.*
[2] *Universidade Federal do Rio Grande do Norte, Departamento de Informática e Matemática Aplicada, Campus Universitário, Lagoa Nova, Natal, RN, Brazil.*

Abstract

This work performs a data driven comparative study of clustering methods used in the analysis of gene expression time courses (or time series). Five clustering methods found in the literature of gene expression analysis are compared: agglomerative hierarchical clustering, CLICK, dynamical clustering, *k*-means and self-organizing maps. In order to evaluate the methods, a *k*-fold cross-validation procedure adapted to unsupervised methods is applied. The accuracy of the results is assessed by the comparison of the partitions obtained in these experiments with gene annotation, such as protein function and series classification.

Key words: clustering methods, gene expression time series, unsupervised cross-validation, cluster validation.

Received: August 15, 2003; Accepted: August 20, 2004.

Introduction

In time course experiments, the expression of a certain cell is measured in some time points during a particular biological process. By knowing groups of genes that are expressed in a similar fashion through a biological process, biologists are able to infer gene function and gene regulation mechanisms (Quackenbush, 2001; Slonim, 2002). Since these data consist of expression profiles of thousand of genes, their analysis cannot be carried out manually, making necessary the application of computational techniques such as clustering methods.

There has been a great deal of work on the application of such methods to gene expression data, each one using distinct data sets, clustering techniques and proximity indices. However, the majority of these works has given emphasis on the biological results, with no critical evaluation of the suitability of the clustering methods or proximity indices used. In the few works in which cluster validation was applied with gene expression data, the focus was on the evaluation of the validation methodology proposed (Lubovac *et al.*, 2001; Yeung *et al.*, 2001; Zhu and Zhang, 2000). As a consequence, so far, with the exception of (Costa *et al.*, 2002b; Costa *et al.*, 2002c; Datta and Datta, 2003), there is no validity study on which proximity indices or clustering methods are more suitable for the analysis of data from gene expression time series.

Based on this, a data driven comparative study of clustering methods used in the literature of gene expression analysis is carried out in this paper. More specifically, five algorithms are analyzed: agglomerative hierarchical clustering (Eisen *et al.*, 1998); CLICK (Sharan and Shamir, 2002); dynamical clustering (Costa *et al.*, 2002a); *k*-means (Tavazoie *et al.*, 1999) and self-organizing maps (Tamayo *et al.*, 1999). With the exception of the CLICK, all the other methods are popular in the literature of gene expression analysis (Quackenbush, 2001; Slonim 2002). Since the adequacy of the clustering algorithm could be dependent on the proximity metric used, versions of three proximity indices with support to missing values are used in the experiments (Gordon, 1999): Euclidean distance, Pearson correlation and angular separation.

All the experiments are performed with data sets of gene expression time series of the yeast *Saccharomyces cerevisiae*. This organism was chosen because there is a wide availability of public data, as well as the availability of an extensive functional classification of its genes. The functional classification will serve as external data information for the validation of the clustering results.

In order to evaluate the clustering methods, the validation method proposed in (Costa *et al.*, 2002c) is used. This method is based on an adaptation of the *k*-fold cross-validation procedure to unsupervised methods. The accuracy of the results obtained in the *k*-fold cross-validation is

Send correspondence to Marcílio C.P. de Souto. Universidade Federal do Rio Grande do Norte, Departamento de Informática e Matemática Aplicada, 59072-970 Campus Universitário, Lagoa Nova, Natal, RN, Brazil. E-mail: marcilio@dimap.ufrn.br.

assessed by an external index (corrected Rand), which measures the agreement between the clustering results and an *a priori* classification, such as gene functional classification or series classification (Jain and Dubes, 1988). Finally, in order to detect statistically significant differences in the results obtained by the distinct clustering methods, a bootstrap hypothesis test for equal means is applied (Efron and Tibshirani, 1993).

Material and Methods

Clustering methods

CLICK

CLICK (Cluster Identification via Connective Kernels) (Sharan and Shamir, 2002) is a recently developed method based on graph theory. Such a method is robust to outliers and does not make assumptions on the number or structure of the clusters. Although CLICK does not take the number of classes as an input, by the use of the homogeneity parameter, one can force the generation of a larger number of clusters.

The method initially generates a fully connected weighted graph, with the objects as vertices and the similarity between the objects as the weights of the edges. Then, CLICK recursively divides the graph in two, using minimum weight cut computations, until a certain kernel condition is met. The minimum weight cut divides the graph in two, in a way that the sum of the weights of the discarded vertices is minimized. If a partition with only one object is found, the object is put apart in a singleton set. The kernel condition tests if a cluster formed by a given graph is highly coupled, and consequently, if it should not be further divided. In order to do so, the algorithm builds a statistical estimator to evaluate the probability that the edges contained in a given graph belong to a single cluster.

Dynamical clustering

Dynamical Clustering is a partitional iterative algorithm that optimizes the best fitting between classes and their representation, using a predefined number of classes (Diday and Simon, 1980). Starting with prototypes values from randomly selected individuals, the method works on two alternates steps: an allocation step, where all individuals are allocated to the class with the prototype with lower dissimilarity, followed by a representation step, where a prototype is constructed for each class.

A major problem of this algorithm is its sensitivity to the selection of the initial partition. As a consequence, the algorithm may converge to a local minimum (Jain and Dubes, 1988). In order to prevent the local minimum problem, a number of runs with different initializations are executed. Then, the best run, based on some cohesion measure, is taken as the result (Jain and Dubes, 1988). Another characteristic of this method is its robustness to noisy data. In addition, when particular proximity index and prototype representations are used, the method guarantees optimization of local criterion (Diday and Simon, 1980). With respect to the proximity indices investigated in this work, only the use of the Euclidean distance version with data containing no missing data guarantees the minimization of the squared error.

More formally, this method looks for a partition P of k classes from an object set E and a vector L of k prototypes, where each prototype represents one class of P. This search is done by minimizing the criterion of fitting between L and P (Diday and Simon, 1980):

$$\Delta(P^{*},L^{*}) = \min\{\Delta(P,L) \,|\, P \in P_k , L \in L_k \} \qquad (1)$$

where P_k is the set of partitions of E in k classes and L_k is the set of prototypes associated to the classes. Specifically, let D be a given dissimilarity function; let e_i be the i_{th} object in the set E, where $i = 1, \ldots, n$; x_j denote the j_{th} quantitative value of a element e from E , where $j = 1, \ldots, p$; the criterion $\Delta(P,L)$ and a centroid G_l representing prototype L_l are, respectively, defined as:

$$\Delta(P,\, L) = \sum_{l=1}^{k} \sum_{x \in C_l} D(x, G_l) \qquad (2)$$

$$G_l = (m_{l1}, \ldots, m_{lj}, \ldots, m_{lp}),\ \ m_{lj} = \frac{1}{|C_l|} \sum_{e \in C_l} x_j \qquad (3)$$

k-*means*

k-means is another type of iterative relocation algorithm, which is widely used in cluster analysis studies (Jain *et al.*, 1999). This method is a special case of the dynamical clustering (Jain *et al.*, 1999). Thus, they share some characteristics, such as robustness to outliers, use of a predefined number of classes and sensitivity to the initial partition. Furthermore, like the dynamical clustering method, *k*-means also optimizes the squared-error criterion when the Euclidean distance is used and there is no missing data.

The main distinctions between the *k*-means and the dynamical clustering method are that the former only works with centroid representations of the classes (Jain *et al.*, 1999), and only one object is reallocated in each allocation step (dynamical clustering reallocates all objects in each allocation step). As a result, a strategy on how the objects are considered with respect to reallocation has to be defined. One of such strategies is to generate a random order of the input objects (Jain and Dubes, 1988).

Self-organizing map

The Self-Organizing Map (SOM) is a type of neural network suitable for unsupervised learning (Kohonen, 1997). SOMs combine competitive learning with dimensionality reduction by smoothing the clusters with respect to an *a priori* grid. One of the main characteristics of these networks is the topological ordering property of the clusters generated. Clusters objects are mapped in neighbor regions of the grid, delivering an intuitive visual representation of the clustering. SOMs are reported to be robust and accurate with noisy data (Mangiameli *et al.*, 1996). On the other hand, SOM suffers from the same problems such as those of dynamical clustering: sensibility to the initial parameters settings and the possibility of getting trapped in local minimum solutions (Jain *et al.*, 1999).

The SOM method works as follows. Initially, one has to choose the topology of the map. All the nodes are linked to the input nodes by weighted edges. The weights are first set at random, and then iteratively adjusted. Each iteration involves randomly selecting an object x and moving the closest node (and its neighborhood) in the direction of x. The closest node is obtained by measuring the Euclidean distance or the dot product between the object x and the weights of all nodes in the map. The neighborhood to be adjusted is defined by a neighborhood function, which decreases over time.

Such maps should often have a number of nodes well above the number of real clusters in the data (Vesanto and Alhoniemi, 2000). Also, by a visual inspection of the map, one can select the neighbor nodes that represent each cluster. However, this process is time consuming and open to subjectivity. In fact, it is not a good practice to include subjective procedures in the validation process. One way to overcome the problem just described is to cluster the nodes, after training the map, by using another clustering method. In this additional step, the number of cluster should be equal to the number of clusters in the data. The resulting partition will state which nodes are related to each cluster. In (Vesanto and Alhoniemi, 2000), *k*-means and hierarchical clustering are employed for this task, all of them obtaining good recovery accuracies. For the sake of simplicity, in this study only the average linkage hierarchical clustering will be applied to the SOM nodes.

Agglomerative hierarchical clustering

Agglomerative hierarchical methods are procedures for transforming a distance matrix into a dendrogram (Jain and Dubes, 1988). These algorithms start with each object representing a cluster, then the methods gradually merge theses clusters into larger ones. Intuitively, agglomerative methods yield a sequence of nested partitions starting with the trivial clustering in which each item is in a unique cluster, and ending with the trivial clustering in which all items are in the same cluster.

Among the different agglomerative methods, there are three broader used variations: complete linkage, average linkage, and single linkage. These variations differ in the way cluster representations are calculated; see Jain and Dubes (1988) for more details. Depending on the variation used, the hierarchical algorithm is capable of finding non-isotropic clusters, including well-separated, chain-like, and concentric clusters (Jain *et al*., 1999). However, since such methods are deterministic, individuals can be grouped based only on local decisions, which are not re-evaluated once decisions are made. As a consequence, these methods are not robust to noisy data (Mangiameli *et al.*, 1996).

In this paper, the focus will be on the average linkage hierarchical clustering method or UPGMA (unweighed pair group method average), as it has been extensively used in the literature of gene expression analysis (Eisen *et al.*, 1998). In such a method, the proximity between two clusters is calculated by the average proximity between the objects in one group and the objects in the other group.

Due to the fact that the methodology applied in this work is only suitable for the evaluation of partitions, the hierarchies are transformed into partitions before being evaluated. One way to do so is to cut the dendrogram in a certain level. Also, the hierarchical method can be used as initialization to the *k*-means and the dynamical clustering. This practice improves the initial conditions of these partitional methods that receive the hierarchical results as input (Jain and Dubes, 1988).

Cluster validity

The evaluation of clustering results in an objective and quantitative fashion is the main objective of cluster validity. Despite its importance, cluster validity is rarely employed in applications of cluster analysis. The reasons for this are, among others, the lack of general guidelines on how cluster validity should be carried out, and the great need of computer resources (Jain and Dubes, 1988). In this section, a methodology for cluster validity, which will be used to compare the clustering algorithms analyzed in this work, is described.

External indices

External indices are used to assess the degree of agreement between two partitions (U and V), where partition U is the result of a clustering method and partition V is formed by an *a priori* information independent of partition U, such as a category label (or classification) (Jain and Dubes, 1988). There are a number of external indices defined in the literature, such as Hubbert, Jacard, Rand and corrected Rand (or adjusted Rand) (Jain and Dubes, 1988). One characteristic of most of these indices is that they can be sensitive to the number of classes in the partitions or to the distributions of elements in the clusters. For example, some indices have a tendency to present higher values for partitions with more classes (Hubbert and Rand), others for partitions with a smaller number of classes (Jaccard) (Dubes, 1987). The corrected Rand index, which has its values corrected for chance agreement, does not have any of these undesirable characteristics (Milligan and Cooper, 1986). Thus, the corrected Rand index - CR, for short - is the external index used in the validation methodology used in this work.

More formally, let $U = \{u_1, \ldots, u_r, \ldots, u_R\}$ be the partition given by the clustering solution, and let $V = \{ v_1, \ldots, v_c, \ldots, v_C \}$ be the partition defined by the *a priori* classification. The equation for CR can be defined as follows:

$$\text{Correct Rand} = \frac{\sum_{i}^{R}\sum_{j}^{C}\binom{n_{ij}}{2} - \binom{n}{2}^{-1}\sum_{i}^{R}\binom{n_{i\bullet}}{2}\sum_{j}^{C}\binom{n_{\bullet j}}{2}}{\frac{1}{2}\left[\sum_{i}^{R}\binom{n_{i\bullet}}{2} + \sum_{j}^{C}\binom{n_{\bullet j}}{2}\right] - \binom{n}{2}^{-1}\sum_{i}^{R}\binom{n_{i\bullet}}{2}\sum_{j}^{C}\binom{n_{\bullet j}}{2}}$$

where n_{ij} represents the number of objects that are in clusters u_i and v_i; n_i indicates the number of objects in cluster u_i; $n_{.j}$ indicates the number of objects in cluster v_j; and n is the total number of objects.

CR can take values in [-1,1], where the value 1 indicates perfect agreement between the partitions, whereas values near 0 (or negatives) correspond to cluster agreement found by chance. In fact, an analysis by Milligan and Cooper (1986) confirmed that CR scores near 0 when presented to clusters generated from random data, and showed that values lower than 0.05 indicate clusters achieved by chance.

Cross-validation

The comparison of two supervised learning methods is, often, accomplished by analyzing the statistical significance of the difference between the mean of the classification error rate, on independent test sets, of the methods evaluated. In order to evaluate the mean of the error rate, several (distinct) data sets are needed. However, the number of data sets available is often limited. One way to overcome this problem is to divide the data sets into training and test sets by the use of a *k*-fold cross validation procedure (Mitchell, 1997).

This procedure can be used to compare supervised methods, even if only one data set is available. The procedure works as follows. The data set is divided into *k* disjoint equal size sets. Then, training is performed in *k* steps, each time using a different fold as the test set and the union of the remaining folds as the training set. Applying the distinct algorithms to the same folds with *k* at least equal to thirty, the statistical significance of the differences between the methods can be measured, based on the mean of the error rate from the test sets.

In unsupervised learning (or cluster analysis), when there is an *a priori* classification of the data set available, the comparison between two methods can also be done by detecting the statistical significance of the difference between the mean values of a certain external index. But again, the number of training sets available is also limited. In (Costa *et al.*, 2002c), a method to overcome this problem was presented. Such a method, which will be used in this work, is an adaptation of the *k*-fold cross-validation procedure for unsupervised methods, as described below.

The data set is, in the unsupervised *k*-fold cross-validation procedure proposed in (Costa *et al.*, 2002c), also divided in *k* folds. At each iteration of the procedure, one fold is used as the test set, and the remaining folds as the training set. The training set is presented to a clustering method, giving a partition as result (training partition). Then, the nearest centroid technique is used to build a classifier from the training partition. The centroid technique calculates the proximity between the elements in the test set and the centroids of each cluster in the training partition (the proximity must be measured with the same proximity index used by the clustering method evaluated). A new partition (test partition) is then obtained by assigning each object in the test set to the cluster with nearest centroid (as defined in Eq. (3)). Next, the test partition is compared with the *a priori* partition (or a *priori* classification) by using an external index (this *a priori* partition contains only the objects of the test partition). At the end of the procedure, a sample with size *k* of the values for the external index is available.

The general idea of the *k*-fold cross-validation procedure is to observe how well data from an independent set are clustered, given the training results. If the results of a training set have a low agreement with the *a priori* classification, so should have the results of the respective test set. In conclusion, the objective of the procedure is to obtain *k* observations of the accuracy of the unsupervised methods with respect to an *a priori* classification, all this with the use of independent test folds.

Bootstrap two-sample hypothesis testing

Two-sample hypothesis tests are applied to measure the significance of the difference between the sample means of two random variables. In this work, these two samples are formed by the values of the external index provided by the unsupervised *k*-fold cross-validation procedure for the two clustering methods to be compared. The test indicates if a sample mean of a clustering algorithm can be stated to be superior to the other. The hypothesis test used in this work is based on bootstrap resampling. The bootstrap method was chosen due to its capacity to build accurate estimates when a limited number of elements are available in the samples. Furthermore, the bootstrap method has the advantage of not making parametric assumptions about the sample distributions. The exact description of the bootstrap hypothesis test for equal means can be found in Efron and Tibshirani (1993) page 224.

Data sets

Since there is a wide availability of public data from the yeast *Saccharomyces cerevisiae*, as well as the availability of an extensive functional classification of its genes allowing the validation of the clustering results, in this paper the focus is on data from this organism. More specifically, one classification scheme and two data sets from the yeast are used.

Yeast functional classification

Munich Information Center for Protein Sequences Yeast Genome Database (MYGD) is the main scheme for classifying protein function of the yeast organism (Mewes *et al.*, 2002). This classification scheme is currently composed of a tree with 249 classes spread over five levels. Genes can be assigned to more than one class; consequently the overlap of classes is large, with genes being assigned to an average of 2.9 classes. Out of the 6,200 known yeast ORFs (Open Reading Frames), around 3,900 belong to at least one of the MYGD classes. (Original data available at: http://mips.sf.de/proj/yeast/catalogues).

These data are used as the external category label in order to evaluate the accuracy of the clustering results. In other words, these classification data do not contain any gene expression data, but they are used in conjunction with expression data sets, supplying a label for the genes contained in the expression data sets. In fact, two classification schemes were obtained from these data, the *FC* and the *REDUCED FC*, as described below.

The *FC* classification scheme is formed by the thirteen first level classes of the MYGD, as in (Zhu and Zhang, 2000). These classes are expected to show similar expression profiles. The *REDUCED FC* is composed of five MYGD classes that have shown a high tendency to cluster together (Eisen *et al.*, 1998*)*. Furthermore, genes belonging to these classes have been successfully used for building function prediction classifiers using supervised methods (Brown *et al.,* 2000).

Yeast all

This data set contains data from five yeast experiments, where 6,200 ORFs had their expression profiles measured using cDNA microarrays. The ORF profiles contain 71 time points, observed during the following five biological processes: the mitotic cell division (cycle alpha, cdc15, elutration); sporulation and diauxic shift (Eisen *et al.*, 1998). Some of the genes contain missing values, either because insignificant hybridization levels were detected, or because the genes were not measured in certain processes. (Data available at: http://genome-www.stanford.edu/clustering).

Two data sets were devised from the original *Yeast All* data set, the *FC Yeast All* and the *Reduced FC Yeast All*. The *FC Yeast All* data set contains only genes in the *FC* classification. A missing data filter was applied to this data set, excluding profiles with more than 20% of missing attributes. As in Heyer *et al.* (1999), a final filtering was employed in order to remove uninformative genes with low expression levels or with low variance between the time points.

In these removed ORFs, the expression level did not vary over time. Thus, these profiles were considered uninformative in relation to gene function. In order to apply this filtering, genes were ranked according to their variance, where the ones within the 45% lowest values (Heyer *et al.,* 1999), were removed. In the end, the *FC Yeast All* data set contained 1,765 genes. The *Reduced FC Yeast All* data set contains only genes from the *Reduced FC* classification. Since there is a reduced number of genes in this data set, only the missing filter was applied, leaving 205 genes.

Mitotic cell cycle (CDC 25)

This data set was obtained in an experiment from the Yeast organism during the mitotic cell division cycle (Cho *et al.,* 1998). The set contains the expression profiles measured with oligonucleotides arrays during 17 time points, with a similar set of ORFs as the one used in the *Yeast All* data set.

Two data sets were also devised from the *Mitotic Cell Cycle*, the *FC CDC 25* and the *Series CDC 25*. In the *FC CDC 25* dataset, only genes in the *FC* classification were considered. A variance filtering was employed in order to remove the 45% of the genes with lowest variance. These data sets did not contain any missing data. The final number of genes in this data set was 1,869. The *Series CDC 25* data set contains genes belonging to a visual classification of the series shape performed by Cho *et al.* (1998). In this classification, 420 genes were assigned to one of five known phases of the cell cycle (some of the genes were assigned to a multiple phase class). There was no need to pre-process this data set, as only informative gene profiles were included in the classification.

Experiments

The experiments compare five different types of clustering algorithms: SOM, dynamical clustering, *k*-means, and dynamical clustering and *k*-means with initialization from the hierarchical method. Each of these algorithms was implemented with versions of three proximity indices widely used in the literature of gene expression data analysis: Angular Separation (*AS*), Pearson Correlation (*PC*) and Euclidean Distance (*ED*) (Costa *et al.*, 2002c). As the implementation of the CLICK algorithm used in this work does not support the Euclidean distance version, such an algorithm was tested only with *AS* and *PC*. Furthermore, with respect to the Euclidean distance version, experiments are performed with the data vectors in three forms, namely, original (ED_1), normalized (ED_2) and standardized (ED_3) values. This yields five distinct settings of proximity indices and pre-processing.

In order to demonstrate the usefulness of the validation methodology, a random assignment method was also included in this evaluation. This method simply assigns randomly the objects in the input data set to a cluster. The results (means) obtained with the random assignment method are taken as the worst case. All other clustering methods should obtain values signi?cantly higher than it.

The experiments were accomplished by presenting the four data sets (*FC Yeast All*, *Reduced FC Yeast All*, *FC CDC 25* and *Series CDC 25*) to all these methods and indices settings, with the exception of the CLICK algorithm that was presented only to the *FC CDC 25* and *Series CDC 25* data sets. This was the case for the implementation of the

CLICK algorithm used which does not support missing data - from the data sets employed, only *FC CDC 25* and *Series CDC 25* data set did not present missing data.

More specifically, for each method, proximity index, and data set a thirty-fold unsupervised cross-validation was applied. Afterwards, the mean values of the corrected Rand index (CR) for the test folds were measured. Next, the mean of CR obtained by the five settings of proximity indices and pre-processing were compared two by two, using the bootstrap hypothesis test with 1,000 bootstrap samples. Initially, the hypothesis tests only compared the results of experiments developed with the same clustering methods and data sets. From this, only the proximity indices with best accuracy for a given clustering method and data set were selected (Costa *et al.*, 2002) for further comparison. Once this selection was accomplished, the clustering algorithms were compared by using hypothesis tests for each data set.

In order to perform the experiments with dynamical clustering and *k*-means methods, the implementation in Costa *et al.* (2002a) was used. In terms of the parameters of these two methods, the number of clusters was set to the number of *a priori* classes (the number of clusters was also set to the number of *a priori* classes in the other methods), and the number of distinct initializations used was 100.

In relation to the CLICK method, an implementation available in the software Expander was utilized. (Expander available at: http://www.cs.tau.ac.il/~rshamir/expander/expander.html). As previously mentioned, this implementation supports neither the Euclidean distance version, nor missing data. The other parameters were set to their default value.

The SOM Toolbox for Matlab was used to run the SOM experiments (SOM Toolbox available at: http//www.cis.hut.fi/projects/somtoolbox). The original implementation only upports the Euclidean distance. Thus, in order to include Pearson correlation and angular separation, modifications were made in the code. As the SOM requires many parameterization experiments, in this work only the topology was varied. This choice is based on a previous study with gene expression data. In such a study, the authors found that the topology was the parameter with the highest impact on the results (Jonsson, 2001).

In order to set the other parameters of the SOM, a method available in the SOM toolbox that uses a number of heuristics to set the parameters was employed. As not all the results obtained with this parameterization were satisfactory, another parameterization based on the one used in Vesanto and Alhoniemi (2000) was used (this parameterization is refereed to as *VESANTO*, whereas the former is referred to as *DEFAULT*). The *VESANTO* parameterization used 10 epochs and a learning rate of 0.5 during the ordering phase. The initial radius was set to the topology highest dimension and the final radius to half the highest dimension. In the convergence phase, 10 epochs and a learning rate of 0.05 were used. The initial radius was set to half the highest topology dimension minus 1 and the final radius to 1. In both phases, the neighborhood function was the Gaussian. With respect to the topology, the following procedure was applied. An initial topology is chosen. Additionally, experiments with a larger and smaller topology are also performed. If the initial topology obtains the best results, then no more experiments are performed. Otherwise, the same process is repeated for the topology with the best result.

The software R was used with the hierarchical clustering experiments (software available at: http://www.r-project.org/). As the external index used in this work is suitable only for partition comparison, the results of the hierarchical methods were supplied as input to the dynamical clustering and the *k*-means methods. In order to build the initial partition from the hierarchical methods, the trees were run from root to the leaves, then the *n* first sub-trees were taken as the clusters (sub-trees with less than 5 elements were ignored). Next, these *n* clusters were used to build the initial partition.

Results

Only the proximity indices with best accuracy for a given clustering method and data set were selected (Costa *et al*., 2002c). These proximity indices are illustrated in Table 1.

According to Figure 1, the dynamical clustering obtained a higher accuracy than the other clustering methods. The null hypotheses were rejected in favor of the dynamical clustering in comparison to the random assignment and the hierarchical clustering at $\alpha = 0.01$, where α stands for the significance level of the equal means hypothesis test. SOM and *k*-means also achieved a significant higher accuracy than the random assignment and the hierarchical clustering. In these cases, the null hypotheses were rejected in favor of the *k*-means and the SOM in comparison to the random assignment ($\alpha = 0.02$) and the hierarchical clustering ($\alpha = 0.05$). The dynamical clustering and the *k*-means both with hierarchical initialization also achieved a significantly higher accuracy than the random assignment and the hierarchical clustering. In these cases, the null hypotheses were

Table 1 - Proximity metrics with best accuracy.

	FC *Yeast All*	*Red. FC* *Yeast All*	*FC* *CDC 25*	*Series* *CDC 25*
SOM	*PC*	*PC*	ED_2	*PC*
Hierarchical Clust.	*AS*	*PC*	*PC*	*PC*
Dynamical Clust.	*AS*	ED_1	ED_2	ED_3
k-means	*AS*	ED_1	ED_2	*AS*
Hierarchical + dynamical	*AS*	ED_2	*AS*	*AS*
Hierarchical + *k*-means	*AS*	ED_2	ED_2	*AS*
CLICK	-	-	*AS*	*PC*

rejected in favor of the dynamical clustering and the *k*-means in comparison to the random assignment ($\alpha = 0.05$) and the hierarchical clustering ($\alpha = 0.05$).

The mean values of corrected Rand for the experiments with the *Reduced FC Yeast All* data set are presented in Figure 2. The random assignment method obtained the lowest accuracy in comparison to all the other methods. The null hypotheses were rejected in favor of the SOM, the hierarchical clustering, the dynamical clustering and the *k*-means (with or without hierarchical initialization) in relation to the random assignment method at a $\alpha = 0.01$. No other significant differences were detected among the methods.

Figure 3 illustrates the mean values of the corrected Rand of the experiments with the *FC CDC 25* data set. The CLICK method obtained a lower value when compared to those achieved by the other methods, including the random assignment. In these cases, the null hypotheses were rejected in favor of all the other methods at $\alpha = 0.01$. The *k*-means (with or without hierarchical initialization) and the SOM obtained significantly higher accuracy than the random assignment and the hierarchical clustering. The null hypotheses were rejected in favor of the SOM and the *k*-means at $\alpha = 0.01$. Dynamical clustering (with or without hierarchical initialization) also obtained significantly higher accuracy than the random assignment and the hierarchical clustering. The null hypotheses were rejected in favor of the dynamical clustering at $\alpha = 0.05$.

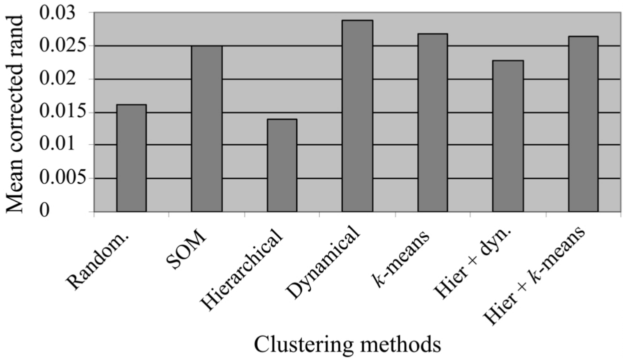

Figure 1 - Mean of corrected Rand values from the *FC Yeast All* experiments.

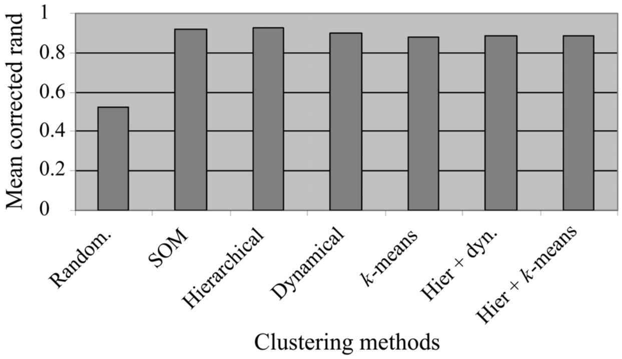

Figure 2 - Mean of corrected Rand values from the *Reduced FC Yeast All* experiments.

Figure 4 shows the mean values of the corrected Rand for the experiments performed with the *Series CDC 25* data set. The random assignment method obtained the lowest values in comparison to all the other methods. In these experiments, the null hypotheses were rejected in favor of the SOM, hierarchical clustering, CLICK, dynamical clustering and *k*-means (with or without hierarchical= initialization) at $\alpha = 0.01$. No other significant differences were detected among the methods.

Discussions

In terms of the hierarchical clustering, low accuracies were achieved in experiments with the *FC CDC 25* and *FC Yeast All 25* data sets. This was not the case of the two other data sets (*Reduced FC Yeast All* and *Series CDC 25*), as the hierarchical clustering obtained accuracies as high as the other methods. One could conclude that the hierarchical clustering has some trouble in clustering larger data sets formed by the complete *Functional Classification* (*FC*) scheme. The clusters in the data sets based on the *FC* scheme are not so compact and isolated when compared to the ones with the *Reduced FC* and the series shape classification. The *FC* data sets have a larger number of genes and

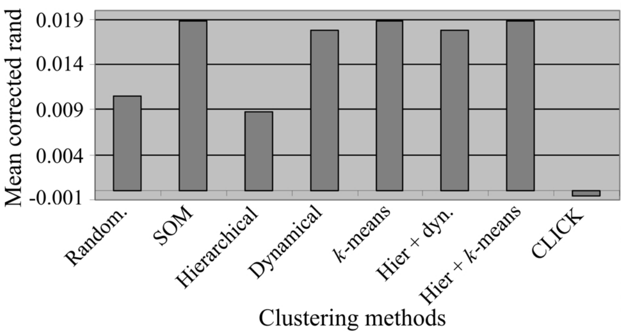

Figure 3 - Mean of corrected Rand values from the *FC CDC 25* experiments.

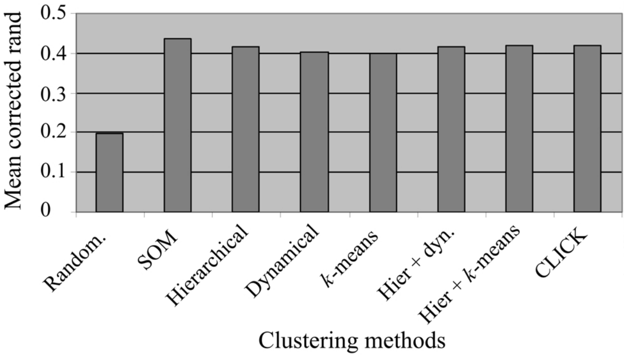

Figure 4 - Mean of corrected Rand values from the *Series CDC 25* experiments.

their classifications were not devised from gene expression analysis. Given the lack of robustness of the hierarchical clustering methods to outliers and noisy data, the low accuracies for the *FC* data sets are expected. These results are also compatible with other comparative analyses of clustering methods for gene expression. In Datta and Datta (2003), the average hierarchical clustering also obtained worse results than other clustering methods, such as the *k*-means and model-based methods. The hierarchical methods also showed a low stability in the experiments presented in Costa *et al.* (2002b).

In the *Series CDC 25* experiments, CLICK achieved the highest mean for the corrected Rand in relation to all the other methods. On the other hand, this very same algorithm obtained negative values for the *FC CDC 25* data set. As mentioned before, the CLICK method finds the number of clusters automatically. This task was perfectly performed for the *Series CDC 25*, where six clusters were encountered in most of the experiments. This was not the case for the *FC CDC 25* experiments, where the number of clusters varied from around 20 to 26 with the *PC* and from around five to seven with the *AS*. These results suggest that CLICK showed instability in clustering the *FC CDC 25* data set. In fact, one could argue that CLICK presented similar problems as those presented by the hierarchical clustering. However, since only one data set with the complete *Functional Classification* was used, further experiments are necessary to investigate this issue properly.

As a whole, *k*-means, dynamical clustering (both with or without hierarchical initialization) and SOM obtained high accuracies in all experiments. The use of the hierarchical initialization does not affect the accuracy of *k*-means and dynamical clustering, even if the hierarchical method alone does not achieve a good accuracy. Indeed, the hierarchical initialization reduces the run time of both dynamical clustering and *k*-means experiments, as there is no need for several random initializations.

The SOM has one main disadvantage in relation to the *k*-means and the dynamical clustering, since such an algorithm required more complex experiments for selecting the parameters. On the other hand, SOM returns a topological map, where the clusters have neighborhood relations. This structure is much more informative than simple partitions returned by the *k*-means and the dynamical clustering.

With respect to the different results achieved with the data sets used, both the *reduced FC Yeast All* and the *Series CDC 25* consist of filtered data sets obtained by a computational clustering analysis followed by an analysis carried out by a human specialist. These data sets have separable classes and a reduced level of noise. On the other hand, the *FC Yeast All* and the *FC CDC 25* are very crude data sets containing noisy data, inseparable clusters and outliers. The data sets obtained by gene expression experiments are more similar to the ones in the second category. Only after the application of the clustering methods it is possible to obtain as "nice" data sets as the one in the first class. In other words, the clustering methods should be able to "easily" obtain results in the first class; however, in the real world applications the second class of data set is the one that is in fact more important.

Regarding the use of gene annotation as an *a priori* classification, in *FC Yeast All* and *FC CDC 25* data sets, where the complete functional classification was used, a low agreement with the clustering results was found. In these experiments, the mean values of the corrected Rand were smaller than 0.05. A previous study (Gertein and Janssen, 2000), using similar data sets, had already indicated that the functional classification has only a weak relation to the clustering of gene expression profiles. The reasons for this are, among others, the vague definitions of some functions and the great overlap of the classes (Gertein and Janssen, 2000).

The overlap among classes has also a direct impact on the value of the corrected Rand index. This is mainly due to the fact that the correction for randomness contained in the corrected Rand index considers only hard (crisp) partitions. Thus, such a correction is too strict for partitions with class overlap, such as the complete *FC* used in this work (see Section Yeast Functional Classification). This is because the number of disagreements (elements in the same class but at distinct clusters and vice-versa) grows considerably. Despite these problems, the corrected Rand is the external index with the best-reported characteristics. Also, so far, there is no index suitable to analyze partitions overlapping classes in the literature.

Finally, it is important to point out that, although the values obtained for the corrected Rand index with the different clustering methods were low, such values were still significantly higher than those obtained with the random clustering method. The latter had corrected Rand values nearer zero (around 0.01).

Acknowledgements

This research was supported partially by grants from CNPq (Proc. 301801/2003-7, 130916/2001-3 and 350200/2004-1) and CAPES-Procad (Proc. 009/01-8).

References

Brown MP, Grundy WN, Lin D, Cristianini N, Sugnet CW, Furey TS, Ares MJR and Haussler D (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc Natl Acad Sci USA 97:262-267.

Cho R, Campbell M, Winzeler E, Steinmetz L, Conway A, Wodicka L, Wolfsberg T, Gabrielian A, Landsman D, Lockhart J and Davis W (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. Mol Cell 2:65-73.

Costa IG, de Carvalho FAT and de Souto MCP (2002a) A symbolic approach to gene expression time series analysis. In: Ludermir, TB and de Souto, MCP (eds) Proc. of the Brazilian Symposium on Neural Networks, IEEE Computer Society, pp 24-30.

Costa IG, de Carvalho FAT and de Souto, MCP (2002b) Stability evaluation of clustering algorithms for time series gene expression data. In: Bazzan ALC and Carvalho ACPL (eds), Proc. of the Brazilian Workshop on Bioinformatics, pp 88-90.

Costa IG, de Carvalho FAT and de Souto MCP (2002c) Comparative study on proximity indices for cluster analysis of gene expression time series. J Intell Fuzzy Syst 13:133-142.

Datta S and Datta S (2003) Comparisons and validation of statistical clustering techniques for microarray gene expression data. Bioinformatics 19:459-466.

Diday E and Simon JC (1980) Clustering Analysis, Digital Pattern Recognition, Springer-Verlag, New York, pp 47-92.

Dubes R (1987) How many clusters are best? An experiment. Pattern Recogn 20:645-663.

Efron B and Tibshirani R (1993) An Introduction to the Boostrap, Chapman & Hall, New York, 456 p.

Eisen MB, Spellman PT, Brown PO and Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci USA 95:14863-14868.

Gordon AD (1999) Classification, Chapman & Hall, New York, 250 p.

Heyer LJ, Kruglyak S and Yooseph S (1999) Exploring expression data: Identification and analysis of coexpressed genes. Genome Res 9:1106-1115.

Jain AK and Dubes RC (1988) Algorithms for clustering data, Prentice Hall, New Jersey, 320 pp.

Jain AK, Murty MN and Flynn PJ (1999) Data clustering: A review. ACM Comput Surv 31(3):264-323.

Jonsson P (2000) Improving clustering of gene expression patterns. Master Dissertation, Department of Computer Science, University of Sködve, Sweden.

Kohonen T (1997) Self-Organizing Maps, Springer-Verlag, Berlin, 501 p.

Lubovac Z, Olsson B, Jonsson P, Laurio K and Andersson ML (2001) Biological and statistical evaluation of clustering of gene expression profiles. In: D'Attellis CE, Kluev VV and Mastorakis NE (eds), Proc. of Mathematics and Computers in Biology and Chemistry, WSES Press, pp 149-155.

Mangiameli P, Chen SK and West D (1996) A comparison of SOM neural network and hierarchical clustering methods. Eur J Operational Res 93:402-417.

Milligan GW and Cooper MC (1986) A study of the comparability of external criteria for hierarchical cluster analysis. Multivariate Behav Res 21:441-458.

Mitchell T (1997) Machine Learning, McGraw Hill, New York, 432 p.

Quackenbush J (2001) Computational analysis of cDNA microarray data. Nat Rev Genet 6:418-428.

Sharan R and Shamir R (2002) CLICK: A clustering algorithm with applications to gene expression analysis. In: Glasgow J and Rost B (eds) Proc Int Conf Intell Syst Mol Biol, pp. 307-316.

Slonim D (2002) From patterns to pathways: Gene expression data analysis come of age. Nat Genet 32:502-508.

Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Ler ES and Golub TR (1999) Interpreting patterns of gene expression with self-organizing maps: Methods & application to hematopoietic differentiation. Proc Natl Acad Sci USA 96:2907-2912.

Tavazoie S, Hughes JD, Campbell MJ, Cho RJ and Church GM (1999) Systematic determination of genetic network architecture. Nat Genet 22:281-285.

Vesanto J and Alhoniemi E (2000) Clustering of the self-organizing map. IEEE T Neural Networ 11:586-600.

Yeung KY, Haynor DR and Ruzzo WL (2001) Validating clustering for gene expression data. Bioinformatics 17:309-318.

Zhu J and Zhang MQ (2000) Cluster, function & promoter: Analysis of yeast expression array. Pac Symp Biocomput 479-490.