Gene expression

**TimeClust: a clustering tool for gene expression time series**

Paolo Magni*, Fulvia Ferrazzi, Lucia Sacchi and Riccardo Bellazzi

Dipartimento di Informatica e Sistemistica, Università degli Studi di Pavia, Via Ferrata 1, Pavia, Italy

Received and revised on November 16, 2007; accepted on November 30, 2007

Advance Access publication December 6, 2007

Associate Editor: Chris Stoeckert

**ABSTRACT**

**Summary:** TimeClust is a user-friendly software package to cluster genes according to their temporal expression profiles. It can be conveniently used to analyze data obtained from DNA microarray time-course experiments. It implements two original algorithms specifically designed for clustering short time series together with hierarchical clustering and self-organizing maps.

**Availability:** TimeClust executable files for Windows and LINUX platforms can be downloaded free of charge for non-profit institutions from the following web site: [http://aimed11.unipv.it/TimeClust](http://aimed11.unipv.it/TimeClust).

**Contact:** paolo.magni@unipv.it or for software support timeclust@aimed11.unipv.it

**Supplementary information:** A simple user’s guide (example.pdf) is available in the download area together with two trial data sets.

**1 INTRODUCTION**

Living systems are ruled by dynamic processes. Genes are modulated over time in dependence on cellular needs and environmental signals. Time is an important factor to understand cellular mechanisms. Consequently, there is an increasing interest in gene expression time series experiments to investigate a wide range of biological processes (Ernst et al., 2005). One of the most studied processes is the cell cycle, which plays a crucial role in several fields. As highlighted in Bar-Joseph (2004), the analysis of such kind of data opens new computational challenges. Clustering methods specifically designed for time-course experiments are necessary to explore gene expression data, taking advantage of the temporal information. The explicit exploitation of the temporal ordering of the data should allow a more sensitive detection of genes that display a consistent pattern over time. In this article, we present TimeClust, a software tool for the analysis of time series gene expression data. It makes available to the end-user an implementation of two clustering algorithms specifically designed for the analysis of short time series (Ferrazzi et al., 2005; Sacchi et al., 2005), not yet included in any other tool. It also implements the well-known hierarchical clustering and the self-organizing maps.

2 METHODS

2.1 Algorithms

In the following subsections, the aspects of each clustering algorithm that are more relevant for TimeClust users will be reported. Other comments will be provided in the Discussion and Conclusions section.

2.1.1 Hierarchical clustering (HC) HC is an agglomerative clustering technique based on a distance measure between clusters and on a linkage method. In the TimeClust implementation, the user can choose the linkage method among single, complete, average, centroid or ward, the distance measure among correlation, crosscorrelation or Euclidean distance, the number of clusters and the number of nodes to be shown in the tree-based dendrogram. The outputs are the dendrogram, the heat map of gene expression data and figures showing the gene expression time series contained in each cluster. The list of the genes included in each cluster is also produced.

2.1.2 Bayesian clustering (BC) BC is based on a representation of clusters through a stochastic population model. The method assumes that a cluster is characterized by a typical profile and that each gene profile of the cluster differs from the typical one due to random effects (individual variability). All the profiles are described by random walk processes, one of the simplest stochastic models able to fit ‘regular’ curves (Ferrazzi et al., 2005). Similarly to HC, BC adopts an agglomerative clustering procedure, in which the different cluster models are compared on the basis of the Bayesian Information Criterion (BIC) score. The optimal number of clusters is automatically obtained by maximizing the BIC. However, the user can compute a suboptimal solution considering a different number of clusters without rerunning the algorithm. In the TimeClust implementation, the user can also choose if a heuristic distance-based search strategy has to be applied to reduce the computational burden. If it is adopted, the BC solution is not anymore guaranteed to be the one with the highest BIC score (i.e. the best solution), even if it is certainly one with a high score. The complete dendrogram and the BIC score as a function of the number of clusters are always visualized. Moreover, as for HC, a figure for each cluster showing the gene expression time series and a list of the genes included in each cluster are generated.

2.1.3 Temporal abstraction clustering (TAC) TAC groups time series according to the qualitative profile obtained by applying Trend Temporal Abstraction (TTA). TTA transforms time series into a series of time intervals in which increasing, decreasing or steady conditions hold. The trend detection implemented in TimeClust is based on a piecewise linear approximation of the temporal profile. First, a linear regression is run over the points of an interval whose initial width is defined by the user through the parameter trend detection window. If the absolute value of its slope is greater than a fixed threshold (again chosen by the user through the parameter min slope) the algorithm tries

*To whom correspondence should be addressed.*
to aggregate new points to the current interval by shifting the right endpoint and checking the slope of the new regression line. This aggregation step is repeated until no more new points can be added to the current interval. The first point that cannot be aggregated becomes the starting point of a new interval (Sacchi et al., 2005). To make TTA less sensitive to noise, TimeClust allows the user to smooth data before TTA detection by averaging them over a moving window whose width is chosen by the user.

The qualitative profiles can be generated at different levels of details and the corresponding clusters are organized in a tree. In TimeClust, a three-level representation has been implemented: (i) L3 in which a trend label is attached to every interval of the time series; (ii) L2 in which consecutive L3 labels of the same kind are aggregated into a single label; (iii) L1 in which all the steady elements are removed from the L2 label and the resulting adjacent labels of the same kind are further aggregated. Clusters are dynamically identified at each level (starting from L1) iterating the following procedure over the three levels. First, the algorithm checks if a cluster with the same label as the one of the new time series already exists. If it is found, the gene is assigned to that cluster; otherwise, a new cluster is formed at the current and all the lower levels. As result of the TAC procedure, TimeClust can produce several figures of the clustered profiles and save results, on user’s request, in a directory-tree recalling the L1–L3 structure. TimeClust allows the user to directly browse this directory structure or to look for the clusters that show, at the L2 level, a shape similar to a user-specified template.

2.1.4 Self-organizing map clustering (SOM) SOM are unsupervised neural networks whose neurons (the map-units or nodes) are disposed over a two-dimensional map fixed at the beginning. A vector of weights (the codebook) of the same dimension as each input example (i.e. the gene expression time series) is associated to each map-unit. During the SOM training process, the codebook values change. At the end of this process, the input examples can be compared with the nodes by computing a similarity score between each example and each codebook. Generally, each example is then associated to the map-unit showing the highest score (the best matching unit—BMU). Therefore, each map-unit can be viewed as the cluster of the examples assigned to it. Its codebook represents the average pattern of that cluster. Moreover, similar SOM nodes (clusters) tend to group together in the map and to be as far as possible from different ones. In the TimeClust implementation, the user can choose the topology of the map between hexagonal or rectangular and its dimension. Several figures are produced. One of these depicts the map, in which each map-unit is associated with the genes for which it is the BMU. Other figures visualize the maps computed for each of the time instants selected by the user. This representation is very useful to verify, by visual inspection, if SOM maps are almost constant or significantly change over the considered time span. A further figure shows the map again, in which each map-unit is black filled in proportion to the number of genes for which that map-unit is the BMU. Finally a figure for each cluster showing the gene expression time series is provided. The results of the SOM analysis can be saved in two files: the first file contains the codebook of each node that is a BMU; the second one contains the gene names for which the node is the BMU. When SOM are used as a preprocessing tool, these two files can be used as input files for the other clustering methods.

2.2 Implementation

TimeClust is a program written in MATLAB (7.0.4) and compiled with the MATLAB Compiler toolbox (4.2). The graphical user interface has been developed using the MATLAB GUIDE tool. It uses functions included in the Optimization (3.0.2) and Statistics (5.0.2) toolboxes and in the SOM toolbox developed by Vesanto et al. (2000). It is distributed with the runtime MATLAB libraries and does not requires any license. The tool is available for Windows (98/NT/2000/XP) and LINUX (x86/x86_64) platforms. For details about its installation please refer to the TimeClust web site. The tool is very easy to use. When it is started, a window for the selection of the clustering algorithm appears. According to the user’s choice, a further window appears, providing a mask to fill with all the parameters required by the selected clustering algorithm. For user’s convenience, help menus are available in all the windows and a user’s guide can be downloaded from the web site.

3 DISCUSSION AND CONCLUSIONS

This article presented TimeClust, a software tool for clustering gene expression profiles obtained from DNA microarray time-course experiments. DNA microarray data analysis is a complex multi-step process. Gene clustering is usually performed after gene selection on a subset of few hundreds or few thousands of genes, in order to simplify the clustering process itself and the subsequent clustering interpretation. The main contribution of TimeClust is to make available to end-users two powerful algorithms recently proposed and not yet implemented in any software tool. The software is very intuitive, multiplatform and may support research and teaching. It includes several clustering algorithms that, in our experience, can be used in different conditions or for different purposes.

HC can be used for a first exploratory analysis, also on a very large number of genes (even before gene selection). It is considered a sort of reference in gene expression data analysis and provides an output (the dendrogram) that is extremely useful to visually inspect the data and to evaluate the likeness of the clustered genes. Its main cons are that: (i) it does not consider the dynamic nature of the temporal data and the correlation usually adopted as distance measure may not be a reliable criterion of association, mainly for short time series; (ii) it suffers from lack of robustness and the solutions it provides can depend on the ordering of the data; (iii) it does not give indication about the optimal number of clusters thus the choice has to be quite arbitrary.

BC, being a model-based evolution of the HC explicitly designed to consider the specific nature of temporal data, represents the most suitable algorithm for this kind of data. It is quite robust and computes the number of clusters in a non-arbitrary way. Moreover, it preserves the dendrogram representation of results. Its main con is the relatively high computational burden and consequently its applicability to no more than several hundreds of genes. Therefore, its use is advisable only after a suitable gene selection. If even after this the number of genes remains too high, SOM can be successfully used to pre-cluster genes and further reduce the problem dimension.

TAC is suitable to highlight or look for qualitative patterns. It is useful also for a preliminary exploratory analysis of the data (even before gene selection) to immediately discover typical patterns. It is the only method that considers qualitative profiles (i.e. shape and not values). If the goal of the analysis is to search for similar shapes, this approach can provide a valuable solution. In general, in our experience, the level L2 is the most informative but in some cases also the level...
L1 can be very interesting. The main cons are: (i) a relative difficulty in setting the parameters of the algorithm to suitable values; (ii) a possible explosion of clusters with long time series.

SOM are especially suitable for (qualitative) data survey because they have prominent visualization properties. The reduction of data space, which occurs since nodes are generally fewer than examples, is a very interesting property. They are reasonably fast, robust to noisy data, and can be easily scaled to large data sets. The main con is the lack of a tree structure that makes it impossible to detect higher order relationships between clusters of profiles. For these reasons, SOM can be considered as a good instrument to preprocess gene expression data and thus reduce the dimensionality of the problem before applying other more appropriate clustering methods such as BC.

**ACKNOWLEDGEMENT**
The authors thank G. Bocca and A. Tiengo for their contribution in the implementation of the tool. This project was partially funded by the Italian “Ministero dell’Università e della Ricerca” through PRIN2003, PRIN2005 and FIRB-ITALBIONET projects. F.F. was supported by an investigator Fellowship from Collegio Ghislieri, Pavia, Italy.

**Conflict of Interest:** none declared.

**REFERENCES**