BIOINFORMATICS

Vol. 19 Suppl. 1 2003, pages i255–i263
DOI: 10.1093/bioinformatics/btg1036

Using hidden Markov models to analyze gene expression time course data

Alexander Schliep [1,], Alexander Schönhuth [2] and Christine Steinhoff [1]*

[1]Max Planck Institute for Molecular Genetics, Department of Computational Molecular Biology, Ihnestrasse 73, 14195 Berlin, Germany and [2]ZAIK, University of Cologne, Weyertal 80 50937 Cologne, Germany

Received on January 6, 2003; accepted on February 20, 2003

ABSTRACT

Motivation: Cellular processes cause changes over time. Observing and measuring those changes over time allows insights into the how and why of regulation. The experimental platform for doing the appropriate large-scale experiments to obtain time-courses of expression levels is provided by microarray technology. However, the proper way of analyzing the resulting time course data is still very much an issue under investigation. The inherent time dependencies in the data suggest that clustering techniques which reflect those dependencies yield improved performance.

Results: We propose to use Hidden Markov Models (HMMs) to account for the *horizontal* dependencies along the time axis in time course data and to cope with the prevalent errors and missing values. The HMMs are used within a model-based clustering framework. We are given a number of clusters, each represented by one Hidden Markov Model from a finite collection encompassing typical qualitative behavior. Then, our method finds in an iterative procedure cluster models and an assignment of data points to these models that maximizes the joint likelihood of clustering and models.

Partially supervised learning—adding groups of labeled data to the initial collection of clusters—is supported. A graphical user interface allows quering an expression profile dataset for time course similar to a prototype graphically defined as a sequence of levels and durations. We also propose a heuristic approach to automate determination of the number of clusters.

We evaluate the method on published yeast cell cycle and fibroblasts serum response datasets, and compare them, with favorable results, to the autoregressive curves method.

Availability: The software is freely available at http://algorithmics.molgen.mpg.de/ghmm.

Contact: schliep@molgen.mpg.de

*To whom correspondence should be addressed.

Keywords: Gene Expression, time course, model-based clustering, Hidden Markov Models

INTRODUCTION

Microarray experiments have become a staple in the experimental repertoire of molecular genetics. They can be used to detect or even quantify the presence of specific pieces of RNA in a sample. The experimental procedure is based on hybridization of these RNA-sequences to either oligo or cDNA sequences which are affixed to the array. Microarray experiments can measure the expression levels of thousands of genes simultaneously. The resulting so-called expression profiles allow, for example, investigation of differences in distinct tissue types or between healthy and diseased tissues. When microarray experiments are performed consecutively in time we call this experimental setting a time course of gene expression profiles. The questions this experimental setting tries to address are the detection of the cellular processes underlying the regulatory effects observed, inference of regulatory networks and, ultimately, assignment of function to the genes analyzed in the time courses.

There have been a number of approaches to analyzing such time courses. These can be divided into two classes, depending on whether they assume the different experiments to be independent or not. Methods in the first class do not consider any dependencies between profiles belonging to subsequent time-points, so called *horizontal dependencies*. Examples are hierarchical (Eisen *et al.*, 1998; Gasch *et al.*, 2000) and *k*-means clustering (Tavazoie *et al.*, 1999) or singular value decomposition (Rifkin and Kim, 2002). Note that for those methods permuting time points arbitrarily does not change the result of the clustering. Accounting for the inherent nature of the data and using the dependencies along the time-axis should lead to higher quality clusters.

The clustering methods that belong to the second class are all model-based. Instead of defining a distance mea-

sure, itself a formidable task for time course data, and grouping data points in a way that minimizes an objective function based on the distances between objects, statistical models are used to represent clusters. Cluster membership is decided based on maximizing the likelihood of data points given the cluster models and the assignment of data points to clusters. Model-based clustering is more suitable for time-series data (MacDonald and Zucchini, 1997). The main advantage of model-based clustering is that there is no longer any need to define a distance function between time courses. This is crucial, as several non-critical variances of signals—a delay, a slower rate, a premature cutoff—will be overly emphasized by, say, Euclidean distance. Hence, capturing the essential *qualitative* behavior of time-series is difficult with any method requiring the definition of a distance. Using stochastic models to represent clusters changes the question at hand from how close two given data points are to how likely one particular data point is under the model. A larger robustness with respect to noise is another virtue of the stochastic model. As it is straight-forward to generate artificial data given a model-based clustering, an analysis of the clustering quality based on the *predictive* performance of the inferred set of models becomes feasible. Examples of model-based clustering used for analysis of expression time courses are based on cubic splines (Bar-Joseph *et al.*, 2002) and autoregressive curves (Ramoni *et al.*, 2002a,b).

Another important aspect is suitability for cyclic profiles. Methods that assume cyclic or periodic behavior—as this is the case for cell cycle data—do not apply for differentiation or pathogen response experiments. Ideally, temporal dependencies indicative of cyclic behavior as well as gene profiles displaying non-periodic behavior should be detectable within the same framework.

Our objective was the design of a method that supports the prevalent knowledge discovery process in molecular biology and respects its peculiarities. Usually, clustering is considered an independent step, that during one invocation takes a user from no knowledge at all to a complete picture revealing all groups within the data; i.e. *unsupervised learning*. However, the typical cyclic succession of experiment and data analysis and the resulting incremental gain of information requires a somewhat different mind set, emphasizing different aspects to maximize the usefulness of a computational method in the process. A method should allow

- using prior knowledge, and
- visualizing and analyzing interactively, while
- maintaining a high robustness with respect to noisy and frequently missing data.

What do those requirements translate to? First, the clustering method should be able to cope effectively with unlabeled data, like all prior work applied to expression profiles. It should also cope with additional, labeled data included in the data set to be analyzed and use the information contained in the labeled data to yield a higher clustering quality. Methods that perform *partly supervised* learning (a survey is found in Seeger, 2001) have been the center of active research only recently, for example as extensions of Support Vector Machines (SVMs) (for an introduction see Cristianini and Shawe-Taylor, 2000), a classical *supervised learning* method, with which Mateos *et al.* (2002) classified tumor tissues based on expression profiles.

Second, a simple graphical user interface should provide interactive access to the underlying method and its parameters, ideally by providing an alternative, more easily accessible view and control thereof, surpassing simple entry fields for the usually difficult to interpret parameters. Robustness results from a proper choice of statistical models.

We propose using Hidden Markov Models (HMMs) to account for the horizontal dependencies in time course data. Besides their prevalent use for biological sequence analysis (cf. Durbin *et al.*, 1998), HMMs have been successfully applied for analyzing time course data in a wide range of different problem domains (MacDonald and Zucchini, 1997). They are particularly suitable, if essential types of qualitative behavior can be proposed, as 'grammatical' or 'structural' constraints in the data can be effectively and explicitly modeled. The HMMs are used in model-based clustering to partition a set of expression time courses into clusters. Note that, as there is a one-to-one correspondence between clusters and models, we shall use the terms interchangeably in the following. Starting from an initial collection of HMMs encompassing typical qualitative behavior, an iterative procedure finds cluster models and an assignment of data points to these models that maximizes the joint likelihood of the clustering. Partially supervised learning is achieved by adding models representing and learned from groups of labeled data to the initial collection of clusters and prohibiting reassignment of the labeled data to other clusters. Finally, a graphical user interface serves two distinct tasks. On the one hand it can be used for interactive explorative analysis of a dataset, on the other for the definition of the initial cluster models. In both cases an HMM, either query or initial prototype, can be graphically defined as a sequence of levels and durations, each with user-adjustable amount of variation. The querying mechanism is implemented as an HMM-search; that is, profiles are ranked in decreasing likelihood under the model.

We shall briefly review the prior work regarding analysis of expression time courses. Naturally, many of the well known clustering methods for expression profiles have also been applied to expression time course data. For reasons of brevity we shall not describe those here.

Bar-Joseph *et al.* (2002) base their approach on statistical models for clustering. To cope with the problem of missing values and non-equidistant time points they propose representing each cluster as a spline curve, namely

$$Y_i(t) = s(t)[\mu_i + \gamma_i] + \epsilon_i,$$

where $Y_i(t)$ denotes the observed value of gene i at time t, $s(t)$ the spline basis function, μ the mean of spline coefficients for genes in the respective cluster, γ the cluster specific variation coefficient and ϵ some normally distributed error term. The clustering is computed with an EM-type algorithm. The number of clusters is determined automatically in a penalized maximum-likelihood fashion. They mention that a cluster assignment based on prior biological data might be used and maintained in the clustering.

Ramoni *et al.* (2002a) (cf. also Ramoni *et al.* (2002b)) also use a model-based clustering approach, where the cluster models are autoregressive curves of fixed order. For each cluster of time series a posterior probability can be derived and models with maximum posterior probabilities are chosen agglomeratively, while deciding on the optimal number of clusters by applying the Akaike information criterion.

While clustering is an important step in the analysis, the following methods directly aim at inferring regulatory networks, performing a grouping of the time-series data implicitly.

Friedman *et al.* (2000) infer a Bayesian network which describes interaction between genes. Their graph-based model describes part of the regulatory interactions that underly a time course expression profile.

Chen *et al.* (1999) derive an edge-labeled directed graph from time course microarray experiments by representing all activation and inhibition relations between each pair of genes. After filtering out genes of low absolute or relative expression, they cluster the remaining data with average linkage. After further processing, edges defining activation and inhibition along the time axis are introduced. Subgroups of regulatory candidate genes are determined by applying an optimization method according to the strength of the edges.

Filkov *et al.* (2002) propose investigating pairs of regulated genes that show not only strong correlation between their expression profiles, but also have similarities between strong local signals. They model each time course as a piecewise linear function. Time-points are removed from consideration unless thresholds for either absolute or relative expression are exceeded, or if narrow peaks indicate a singular experimental failure. The resulting profiles for each gene are pairwise compared and a similarity score for each pair of adjusted curves is deduced.

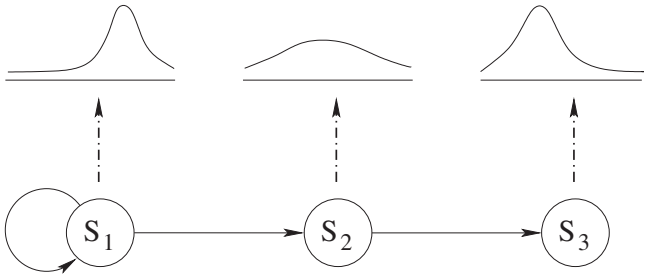

Fig. 1. A Hidden Markov Model visualized as directed graph, the emission pdfs are attached to the nodes. The model depicted is a prototype for down-regulation.

We apply our method to yeast cell cycle (Spellman *et al.*, 1998) and fibroblasts serum response Iyer *et al.* (1999) datasets, and compare them to the autoregressive curves method (Ramoni *et al.*, 2002a).

METHOD

Hidden Markov Models (HMMs) can be viewed as probabilistic functions of a Markov chain (Burke and Rosenblatt, 1958; Petrie, 1969) where each state of the chain can independently produce emissions according to so-called emission probabilities or densities. We shall restrict ourselves to univariate emission probability densities. Extensions to multivariates or mixtures thereof, as well as to discrete emissions, are routine. The following parameters fully determine a Hidden Markov Model, λ: the states S_i, $1 \leq i \leq N$; the probability of starting in state S_i, π_i; the transition probability from state S_i to S_j, a_{ij}; and $b_i(\omega)$, the emission probability density of a symbol $\omega \in \Sigma$ in state S_i. The obvious stochasticity constraints on the parameters apply. Rabiner (1989) gives a thorough introduction to HMMs.

Model-based clustering: The clustering problem we shall address can be formally defined as follows: Given n sequences O^i, not necessarily of equal length, with index set $\mathcal{I} = \{1, 2, \ldots, n\}$ and a fixed integer $K \ll n$. Compute a partition $\mathcal{C} = (C_1, C_2, \ldots, C_K)$ of $\mathcal{I}$ and HMMs $\lambda_1, \ldots, \lambda_K$ maximizing the objective function

$$f(\mathcal{C}) = \prod_{k=1}^{K} \prod_{i \in C_k} L(O^i|\lambda_k)\,. \qquad (1)$$

Here, $L(O^i|\lambda_k)$ denotes the likelihood function, that is, the probability density for generating sequence O^i by model λ_k: $L(O^i|\lambda_k) := P(O^i|\lambda_k)$.

It has been noted before (e.g. Smyth, 2000) that the problem of computing a k-means clustering can be formulated as a joint likelihood maximization problem.

Adapting the k-means algorithm, we propose the following maximum likelihood approach to solve a HMM cluster problem, given a collection of K initial HMMs $\lambda_1^0, \ldots, \lambda_K^0$.

1. **Iteration** ($t \in \{1, 2, \ldots\}$)**:**
   (a) Generate a new partitioning of the sequences by assigning each sequence O^i to the model k for which the likelihood $L(O^i|\lambda_k^{t-1})$ is maximal.
   (b) Calculate new parameters $\lambda_1^t, \ldots, \lambda_K^t$ using the re-estimation algorithm for each model with their start parameters $\lambda_1^{t-1}, \ldots, \lambda_K^{t-1}$ and their assigned sequences.
2. **Stop**, if the improvement of the objective function is below a given threshold, ε, the grouping of the sequences does not change or a given iteration number is reached.

Convergence: The nested iteration scheme does indeed converge to a local maximum. This follows directly from the convergence of the Baum-Welch algorithm and the observation that re-assignment of sequences cannot decrease the likelihood; indeed, this is in fact a nested EM-algorithm.

Determining number of clusters: How to determine the number of clusters poses a difficult problem for all clustering methods. This is aggravated in our case since we do not merely need to specify a *number* of clusters, but rather we need to provide a heterogeneous *collection* of models. While it is rather straightforward to populate this collection of initial models with examples of prototypical behavior, such as up-regulation or down-regulation, there are two relevant questions that need to be addressed. First, does the collection cover all prototypical behaviors occurring in models of expression profiles? Second, does the collection contain enough copies of each prototype to represent those clusters in the data sharing the same prototypical behavior, but differing enough in detail—imagine different levels or stages of up-regulation—to warrant separate clusters?

We address the question of completeness by use of an explicit *noise cluster*. A noise cluster is a simple model that can generate all possible expression profiles with low probability and that is excluded from training. Adding a noise cluster effectively translates to a slight variation of the re-assignment rule in the clustering algorithm. A re-assignment of a profile to the model that maximizes its likelihood *only* occurs when the likelihood exceeds that of the profile under the noise model.

Preserving the broad nature of the noise cluster, which as a consequence bounds the maximal likelihood given for any profile, is essential for keeping most of the profiles assigned to clusters instead of the noise model. The advantage of this approach is that clusters do not get 'smeared' so easily, as, in addition to prototypes unaccounted for in the initial collection, also profiles that are true noise are assigned to the noise cluster. This has been verified by observing a reduction of variances for Gaussian emission PDFs of cluster models when an error model is added (not shown).

There are a number of approaches that deal with the problem of determining the number of clusters. They usually rely on some measure that quantifies the quality of the clustering, for example the Bayesian Information Criterion (Hastie *et al.*, 2001). We propose a method, that is motivated by 'model surgery' Krogh *et al.* (1994) first proposed in the context of learning the topology of profile HMMs. Herwig *et al.* (1999) proposed a similar approach for adjusting the number of clusters in a k-means like algorithm.

Essentially there are two simple rules. First, if a cluster gets assigned very few profiles, delete the cluster and re-assign the profiles to the remaining clusters. Second, if a cluster gets assigned too many profiles, split the cluster into two parts. This is done by copying the model, randomly changing parameters of the two copies uniformly and independently, and continuing with the re-assignment step in the clustering algorithm. This corresponds directly to state splitting and state deletion in HMM model surgery, which, although clearly heuristic in nature, works very well in practice. The heuristic can be extended to avoid splitting clusters that are homogeneous. Rules can be easily formulated based on the variance of the emission PDFs.

Missing data: Missing data can be handled in a straightforward manner. The emission probability distribution of each state is replaced by a mixture of a discrete part emitting a symbol that represents the missing value and the usual probability density function. The mixture coefficient of the discrete part is set to a constant value representing the overall empirical frequency of missing data. The mixture coefficients are excluded from re-estimation in the Baum-Welch step of the clustering algorithm.

Partially supervised clustering

Implementing partially supervised clustering within our proposed method framework is straight-forward. Partially supervised learning refers to additionally learning from labeled data. That is, we have access to labels defining a designated group membership for some of the data we want to cluster. Assuming that distinct group labels in fact do imply differences in the profile, we can infer one HMM from each group of identically labeled profiles. This requires a choice of the model topology, or prototype, and training with the profiles. Note, that

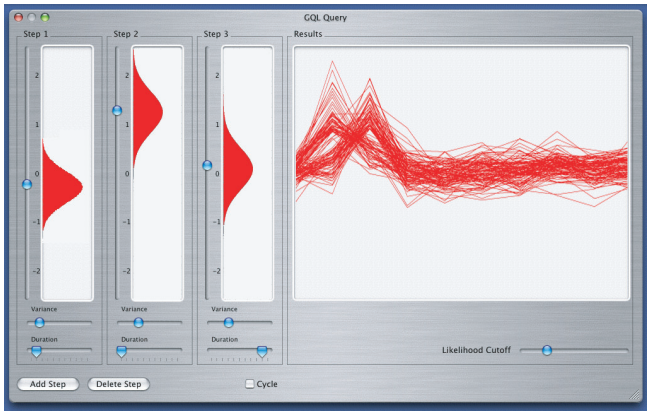

Fig. 2. A screen-shot of the graphical query language implemented as an alternative way of defining either queries or prototype models.

over-fitting is to be avoided in the training step. However, the model parameters are only relevant for the very first iteration of the clustering. Hence, a simple heuristic, for example running Baum-Welch re-estimation for only one or two steps, suffices. The labeled profiles are excluded from re-assignment in the clustering. In the clustering, unlabeled profiles can be assigned to clusters seeded with labeled profiles, driving the models towards generalization. Questions that need to be addressed are the size of the additional initial model collection, as well as its composition. That labeled data is excluded from re-assignment also implies that clusters this data is assigned to should not be modified in the model surgery setting.

A graphical query language for expression profiles

A graphical user interface, see Figure 2, provides a simple way of either defining prototypes, or, more importantly, of finding expression time courses matching a graphically defined query time course. The user can define a variable number of steps for a time course. For each step, the mean and variance of the Gaussian emission PDF can be defined with data sliders, as can the duration that is, for how many time-points the particular step should be inhabitated. These variables define an HMM, each 'step' corresponding to a state; its duration is interpreted as the expected state duration and an appropriate self-transition is computed. Time courses in a dataset can be scored by computing their likelihood under the specified model. Finally, in the pane on the right in Figure 2, all profiles exceeding a user-definable likelihood threshold are displayed.

Post-analyzing clusters

As a cluster may contain many different forms of prototype appearances, a further analysis within one cluster can

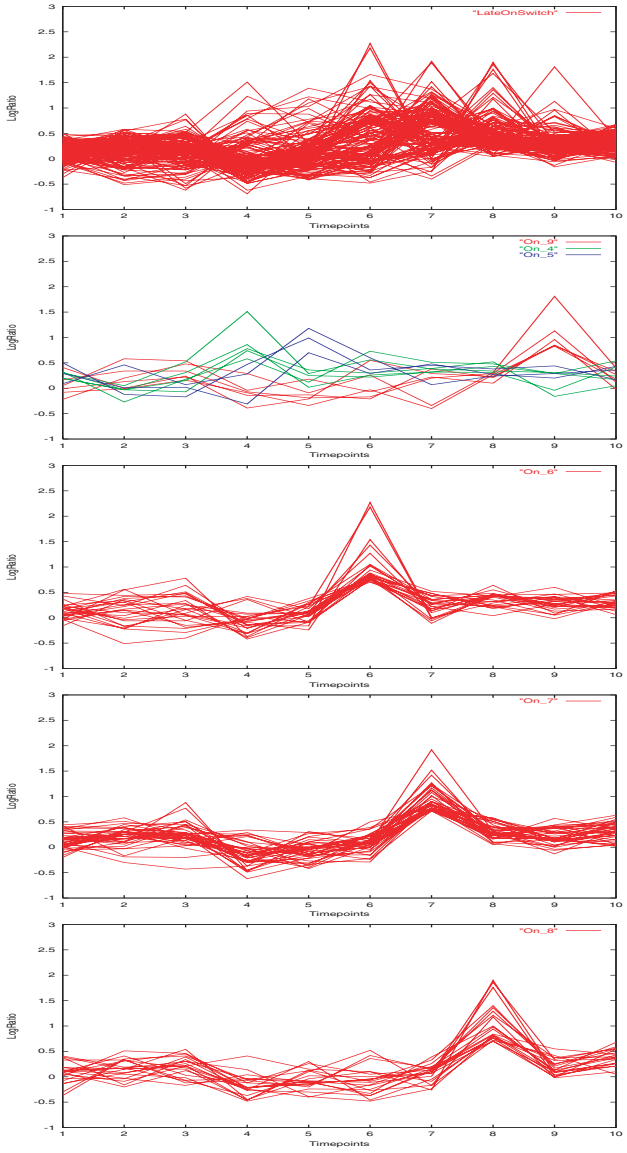

Fig. 3. A cluster representing on-switch behavior (that is a temporary, short up-regulation) and its decomposition according to when the state, in which over-expression occurs, is reached.

be useful. We apply the Viterbi algorithm, which, given a sequence and a model, computes the most probable sequence of the underlying hidden states. This sequence of states is then attached as a label to all sequences belonging to the same model. Sequences can then be sorted according to their labels. We can, for example, group the profiles of a cluster according to point in time and duration of a designated state (see Fig. 3). This makes delayed response phenomena or phase shifts of genes regulated in a cyclic manner easily detectable. See Figures 4, 5, 7 and 8 for further examples.

Implementation

The relevant data structures and algorithms are freely available in a portable C-library, the GHMM (Knab *et al.*, 2002), licensed under the Library GNU General Public License (LGPL). The graphical user interface is based on the GHMM, but additionally uses Python and a GUI-framework.

RESULTS AND EVALUATION

The clustering method has been tested on the following two data sets:

Yeast: Spellman *et al.* (1998) synchronized yeast cells by three different methods, one of which was α-factor based sychronization. The expression levels of 6178 genes were subsequently measured every 7 minutes for the next 140 minutes, which encompasses slightly more than two full cell cycles. (One cell cycle corresponds to about 8 time-points out of total of 18.) After normalization (Spellman *et al.*, 1998), genes that did not show two-fold over- or under-expression in at least one time-point were removed to curb artifacts due to noise in the data. Log ratios of the remaining 1044 gene expression time courses were taken as the input for the clustering algorithm.

Fibroblasts: Iyer *et al.* (1999) studied the physiological response of fibroblasts to serum after growth arrest. The expression levels of about 8,600 genes were measured at a number of non-equidistant time point after stimulation. The time course spanned a 24 hour period. Expression levels were normalized (Iyer *et al.*, 1999) and genes never showing at least a two-fold over- or under-expression were removed. The logarithms of the ratios (time point vs. control) of the remaining 3,384 time courses made up the input for the clustering algorithm.

Prototypes/models

Yeast: We choose an initial collection of 19 models, each having 9 states, in order to allow detection of significant changes along the 18 time-points. Models were designed such that they could be traversed only sequentially (Left-Right Models). This effectively limits the number of parameters. To force alignment of the sequences to the models, a designated end state, only able to emit the end symbol, was added and the designated end symbol was appended to every sequence. Cyclic as well as non-cyclic prototypes were provided so that cell-cycle regulated genes as well as genes showing other phenomena could be detected. We further added a noise model consisting of 18 consecutive states, each with emission probability density function centered around zero with a high variance. Parameters of the noise model were excluded from training in the Baum-Welch step to obtain a constant denoising threshold. The noise model precluded outliers from obfuscating clusters.

Fibroblasts: For this data set 30 left-right models were designed in the same manner as described above, each consisting of five states. Models were meant to represent typical behavior like (temporary) up- or down-regulation, to encompass all possible response patterns. Here, we did not use a noise model in the clustering.

Results

The output of the clustering algorithm was in both cases a collection of trained models and an assignment of each gene expression time course to one of them. The time courses usually adhere to the prototypic behavior of the model they were assigned to, but differ from each other with respect to, say, a possible onset of a up-regulation. These different characteristics within a cluster can be detected by inspection of their most probable sequence of states. Hence, a further classification using the Viterbi labels (see post-analyzing clusters) was performed.

Yeast: Here the result was a collection of prototypes that separate genes showing periodic or cyclic patterns from those that do not clearly. Most of the non-periodic patterns found show early on- or off-switch patterns that could be due to α-factor responses; cf. Figure 6. The cyclic prototypes all suggest a periodicity of about 8 time points, differing only in amplitude and phase. Figure 4 depicts genes sorted according to their duration in the first state of over-expression. Although their duration in the second state of over-expression was not accounted for while sorting, they follow the same behavior 8 time points later, which clearly is a indicator of cell cycle regulation. Figure 5 shows another cluster containing cell cycle regulated genes. Expression levels herein oscillate with greater amplitudes than those of the cluster depicted in Figure 4. Here as well, splitting profiles according to duration in the first peak state already leads to an overall separation with respect to phase shift. Even if the picture of the cluster itself doesn't immediately reveal the regulation patterns of the genes contained therein, a further decomposition will clarify their behaviour along the time axis (see Fig. 7).

We also carried out a CAGED clustering (Ramoni *et al.*, 2002b) using the default parameters; results are depicted in Figure 9. CAGED suggests two clusters. The result is somewhat unconclusive. A biological meaning of the clusters as well as a functional relationship of the genes therein remain unclear, at least at first sight. It seems that our method outperforms CAGED on this data set.

Fibroblasts: The clustering resulted, as above, in a collection of trained models and an assignment of expression time courses to the prototypes. Several response pat-

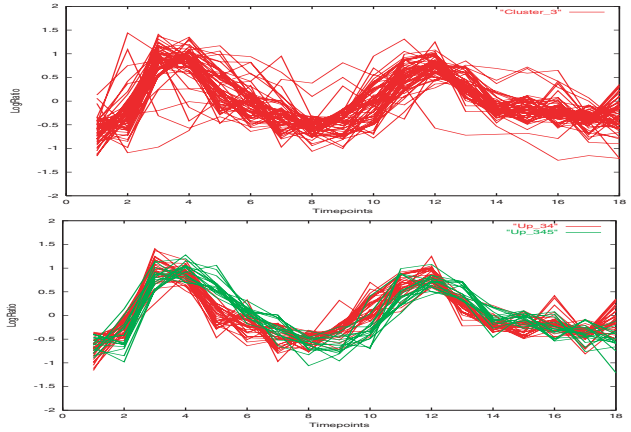

Fig. 4. Yeast: A cluster containing cell-cycle regulated gene expression time courses and a partial decomposition according to the first over-expression state.

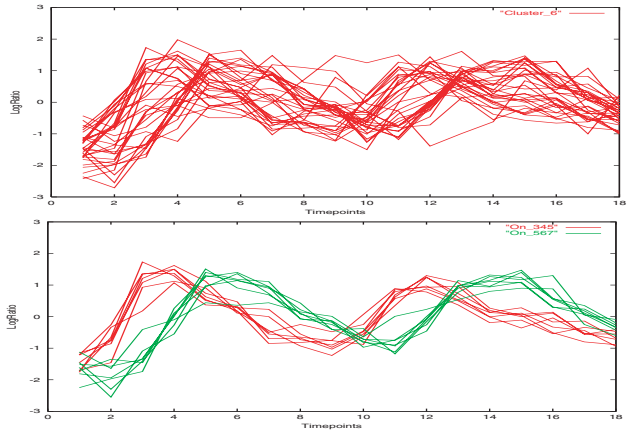

Fig. 5. Yeast:A cluster containing cell-cycle regulated gene expression time courses and a partial decomposition according to duration in the first over-expression state.

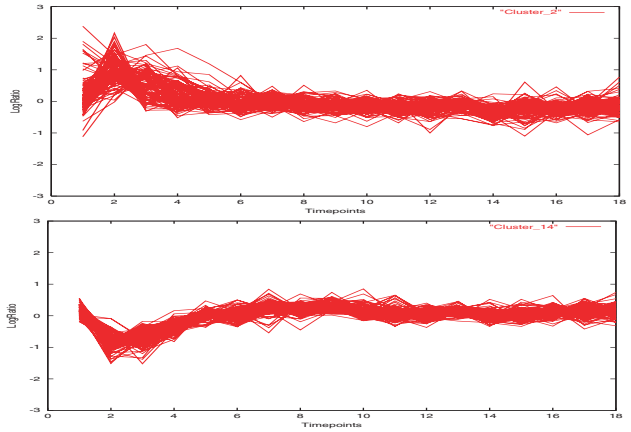

Fig. 6. Yeast: Two clusters of expression time courses containing genes being switched on resp. off (that is a temporary, short up- resp. down-regulation) early in the time course, likely as a response to α-factor treatment.

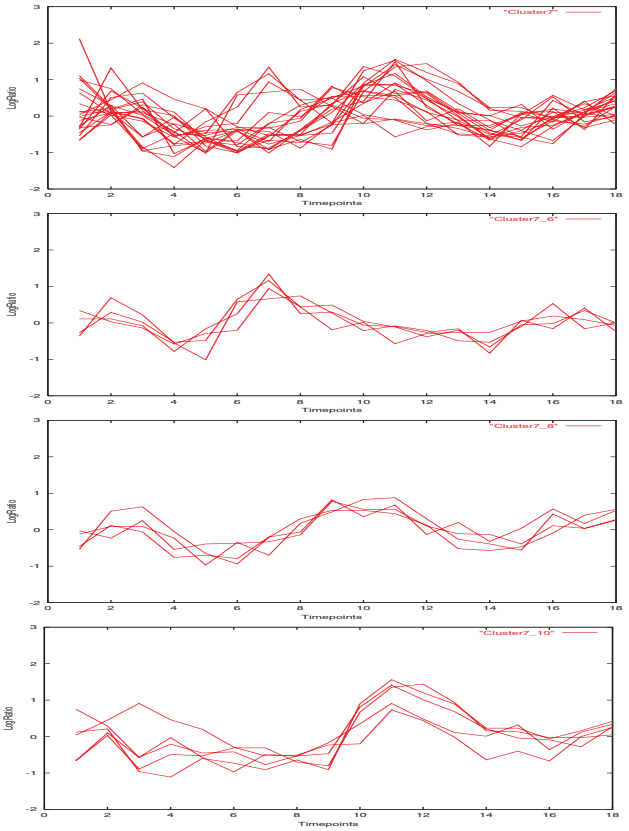

Fig. 7. Yeast: A cluster containing on-switch patterns and its decomposition. The third plot also could be interpreted as cyclic behaviour.

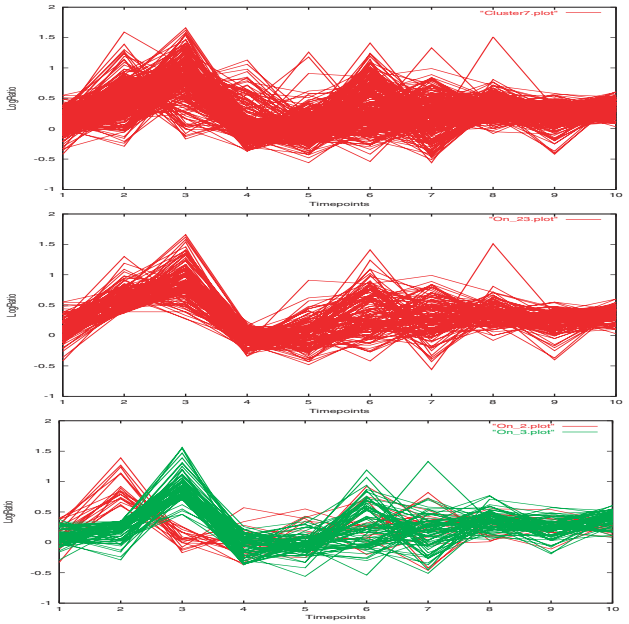

Fig. 8. Fibroblasts: A cluster representing on-switch behaviour early in the time course and its decomposition according to when the state, in which over-expression occurs, is reached.

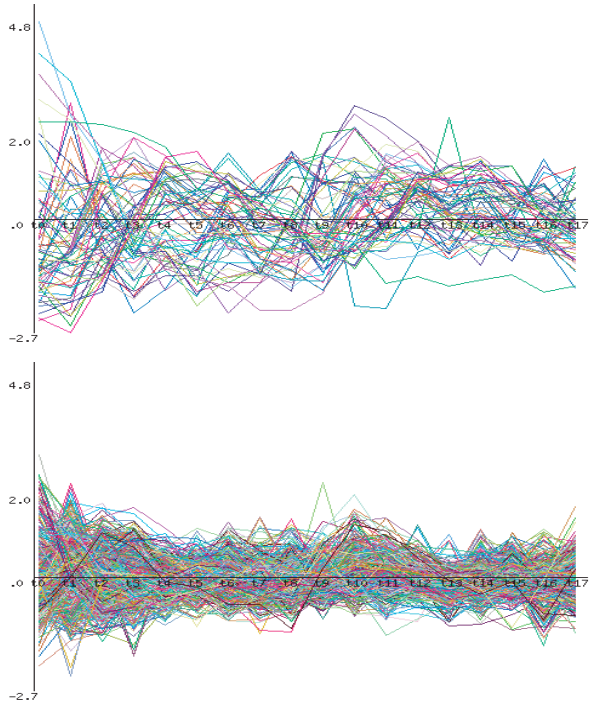

Fig. 9. The two clusters found by CAGED (Ramoni *et al.*, 2002b) run with default parameters.

terns could be detected, many of them showing significant changes at the beginning of the time course, clearly indicating fast responses to the serum stimulation. An example of a fast reaction to stimulation is shown in Figure 8. Genes that were up-regulated in the second and/or third time point were gathered by this model. A decomposition of the cluster in genes that were up-regulated either in only one of these time points or in both is also shown. Another cluster is depicted in Figure 3. This prototype represents a switching on later in the time course. The cluster was then decomposed into those parts that stayed in the over-expression state for exactly one time point.

CONCLUSION

To analyze expression profile series representing cyclic or non-cyclic biological time series, we propose a Hidden Markov Model-Based approach. This method allows us to use prior knowledge as it is given in many biological settings where for some genes the response is already known. Furthermore, the data can be visualized and analyzed interactively while a high robustness with respect to noisy and frequently missing data is maintained. We show that our method leads to very clear results on two different datasets, containing cyclic (Spellman *et al.*, 1998) as well as (partly) non-cyclic (Iyer *et al.*, 1999) data and contrasted this to the approach proposed by Ramoni *et al.* (2002b). Our clustering method combined with the Viterbi-based cluster decomposition leads to very homogeneous and fine grained clusters, which allow to resolve the multitude of distinct regulatory process classes in the data.

The flexible framework we proposed lends itself to a wide range of possible enhancements and additional applications. None of the methodological developments make stronger assumption about the data than being a time course. The HMM clustering can naturally be used with HMMs modeling other types of data, for example biological sequences. In preparation is an extension to represent a time course as a mixture of models, which given the ambiguous role played by many genes is closer to biological reality.

ACKNOWLEDGEMENTS

The second author would like to thank the BMBF for financial support. We would like to thank Tobias Müller for helpful discussions. Berhard Knab, Bernd Wichern and Barthel Steckemetz were instrumental in developing the methodology and implementing the GHMM, as were Achim Gädke, Wasinee Rungsarityotin, Thordis Linda Thorarinsdottir, and Peter Pipenbacher.

REFERENCES

Bar-Joseph,Z., Gerber,G., Gifford,D.K. and Jaakkola,T.S. (2002) A new approach to analyzing gene expression time series data. *6th Annual International Conference on Research in Computational Molecular Biology*.

Burke,C.J. and Rosenblatt,M. (1958) A Markovian function of a Markov chain. *Ann. Math. Stat.*, **29**, 1112–1120.

Chen,T., He,H.L. and Church,G.M. (1999) Modeling gene expression with differential equations. *Pac. Symp. Biocomput.*, 29–40.

Cristianini,N. and Shawe-Taylor,J. (2000) *An Introduction to Support Vector Machines*. Cambridge University Press.

Durbin,R., Eddy,S., Krogh,A. and Mitchison,G. (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. CUP.

Eisen,M., Spellman,P., Brown,P. and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, **95**, 14863–14868.

Filkov,V., Skiena,S. and Zhi,J. (2002) Analysis techniques for microarray time-series data. *J. Comput. Biol.*, **9**, 317–330.

Friedman,N., Linial,M., Nachman,I. and Pe'er,D. (2000) Using bayesian networks to analyze expression data. *J. Comput. Biol.*, **7**, 601–620.

Gasch,A., Spellman,P., Kao,C., Carmel-Harel,O., Eisen,M., Storz,G., Botstein,D. and Brown,P. (2000) Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell.*, **11**, 4241–4257.

Hastie,T., Tibshirani,R. and Friedman,J. (2001) *The Elements of Statistical Learning*. Springer.

Herwig,R., Poustka,A., Mller,C., Bull,C., Lehrach,H. and O'Brien,J. (1999) Large-scale clustering of cDNA-fingerprinting data. *Genome Res.*, **9**, 1093–1105.

Iyer,V.R., Eisen,M.B., Ross,D.T., Schuler,G., Moore,T., Lee,J.C., Trent,J.M., Staudt,L.M., Hudson,J.r., Boguski,J.M.S. *et al.* (1999) The transcriptional program in the response of human fibroblasts to serum. *Science*, **283**, 83–87.

Knab,B., Schliep,A., Steckemetz,B., Wichern,B., Gädke,A. and Thoransdottir,D. (2002) *The GNU Hidden Markov Model Library*. Available from http://www.zpr.uni-koeln.de/∼hmm.

Krogh,A., Brown,M., Mian,I.S., Sjolander,K. and Haussler,D. (1994) Hidden markov models in computational biology: applications to protein modelling. *J. Mol. Biol.*, **235**, 1501–1531.

MacDonald,I.L. and Zucchini,W. (1997) *Hidden Markov and other models for discrete-valued time series*. Chapman and Hall, London.

Mateos,A., Dopazo,J., Jansen,R., Tu,Y., Gerstein,M. and Stolovitzky,G. (2002) Systematic learning of gene functional classes from DNA array expression data by using multilayer perceptrons. *Genome Res.*, **12**, 1703–1715.

Petrie,T. (1969) Probabilistic functions of finite state Markov chains. *Ann. Math. Statist*, **40**, 97–115.

Rabiner,L.R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**, 257–285.

Ramoni,M.F., Sebastiani,P. and Cohen,P.R. (2002a) Bayesian clustering by dynamics. *Mach. Learn.*, **47**, 91–121.

Ramoni,M.F., Sebastiani,P. and Kohane,I.S. (2002b) Cluster analysis of gene expression dynamics. *Proc. Natl Acad. Sci. USA*, **99**, 9121–9126.

Rifkin,S.A. and Kim,J. (2002) Geometry of gene expression dynamics. *Bioinformatics*, **18**, 1176–1183.

Seeger,M. (2001) Learning with labeled and unlabaled data.

Smyth,P. (2000) A general probabilistic framework for clustering individuals. *Technical Report TR-00-09*. University of California, Irvine.

Spellman,P.T., Sherlock,G., Zhang,M.Q., Iyer,V.R., Anders,K., Eisen,M.B., Brown,P.O., Botstein,D. and Futcher,B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, **9**, 3273–3797.

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