BIOINFORMATICS ORIGINAL PAPER

Vol. 23 no. 7 2007, pages 842–849
doi:10.1093/bioinformatics/btl667

Gene Expression

A hidden Markov model-based approach for identifying timing differences in gene expression under different experimental factors

Takashi Yoneya[1,2,*] and Hiroshi Mamitsuka[1]

[1]Bioinformatics Center, Kyoto University, Gokasho Uji, 611-0011, Japan and [2]Pharmaceutical Research Laboratories, Pharmaceutical Division, Kirin Brewery Co. Ltd, 3 Miyahara, Takasaki, Gunma 370-1295, Japan

Received on October 23, 2006; revised on December 1, 2006; accepted on December 28, 2006

Advance Access publication January 19, 2007

Associate Editor: Satoru Miyano

ABSTRACT

Motivation: Time series experiments of cDNA microarrays have been commonly used in various biological studies and conducted under a lot of experimental factors. A popular approach of time series microarray analysis is to compare one gene with another in their expression profiles, and clustering expression sequences is a typical example. On the other hand, a practically important issue in gene expression is to identify the general timing difference that is caused by experimental factors. This type of difference can be extracted by comparing a set of time series expression profiles under a factor with those under another factor, and so it would be difficult to tackle this issue by using only a current approach for time series microarray analysis.

Results: We have developed a systematic method to capture the timing difference in gene expression under different experimental factors, based on hidden Markov models. Our model outputs a real-valued vector at each state and has a unique state transition diagram. The parameters of our model are trained from a given set of pairwise (generally multiplewise) expression sequences. We evaluated our model using synthetic as well as real microarray datasets. The results of our experiment indicate that our method worked favourably to identify the timing ordering under different experimental factors, such as that gene expression under heat shock tended to start earlier than that under oxidative stress.

Contact: t-yoneya@kirin.co.jp

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Experiments by cDNA microarray, which have been widely used for comprehensive gene expression profiling, can be classified into two types: static and time series. A static experiment is used to measure a snapshot of the expression profile in an experimental condition, while a time series experiment is used to measure a continuous change under an experimental condition. Analyzing time series microarray data is important to understanding the dynamic mechanism of biological phenomena, and, in this paper, we focus on the data from time series microarray experiments. In fact, time series experiments have been used to analyze a variety of biological phenomena, including environmental stresses (Gasch *et al.*, 2000; Tirosh *et al.*, 2006), immune responses (Guillemin *et al.*, 2002), developmental studies (Arbeitman *et al.*, 2002), etc.

A major purpose of conducting time series microarray analysis is to check the genes that are expressed in some expected manner under an experimental condition. Computational approaches for assisting this purpose often attempt to check the similarity/difference in time series expression between genes, and clustering time series sequences is a typical example. A lot of techniques including clustering expression sequences have been used for time series analysis (Bar-Joseph, 2004; Filkov *et al.*, 2002). They contain dynamic time warping (Aach and Church, 2001), singular value decomposition (Alter *et al.*, 2000), ANOVA and related approaches (Park *et al.*, 2003; Storey *et al.*, 2005), hidden Markov models (Costa *et al.*, 2005; Schliep *et al.*, 2003, 2004), kernel-based approaches (Borgwardt *et al.*, 2006), clustering with predefined expression patterns (Ernst *et al.*, 2005; Ernst and Bar-Joseph, 2006), etc. We emphasize that the attention of all these methods is concentrated on genes.

In contrast, our focus is not on genes but on factors in microarray experiments, such as experimental conditions. That is, the purpose of this paper is to find the timing difference in the effects of experimental factors on genes. This purpose is obviously important, since we often have to evaluate the timing difference by experimental factors, to understand their exact effects on genes (Chen *et al.*, 2003). In particular, we address the issue of finding the timing difference made by overall genes rather than by specific ones. In order to examine the effect on overall genes, we'll not check the behavior of each gene, but use a set of time series expression profiles obtained under the same experimental factor. We describe our data usage more concretely by using a sample dataset. Table 1 shows a sample of time series expression sequences under two conditions. We can modify this dataset into a set of pairwise sequences, which is shown in Figure 1. By doing so, time series sequences between two experimental conditions can easily be compared. That is, we can see that a gene under Condition A is always expressed earlier that

*To whom correspondence should be addressed.

Table 1. Data example of three genes (genes 1, 2 and 3) with five time points (T1 to T5) under two conditions (Conditions A and B)

	Condition A					Condition B				
	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5
Gene 1	1.2	0	0	0	0	0	1.5	0	0	0
Gene 2	0	0	0	1.8	0	0	0	0	0	1.5
Gene 3	0	1.8	0	0	0	0	0	0	1.8	0

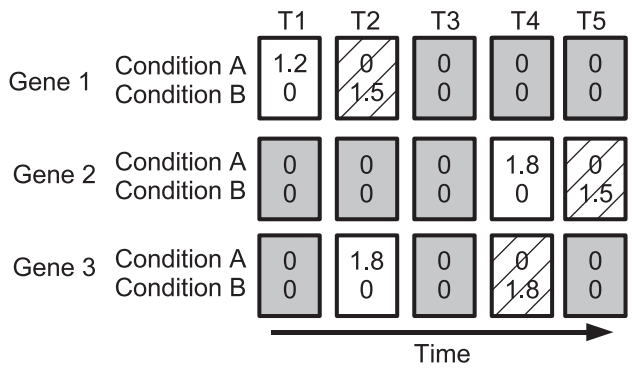

Fig. 1. An input dataset example, which is just a modification of Table 1.

under Condition B. Thus, the next issue is to build a method to capture this type of rules found in a set of pairwise (generally triplewise or more) time series sequences.

For this issue, we present a systematic approach based on a hidden Markov model (HMM) (Durbin *et al.*, 1998; Rabiner *et al.*, 1986). Our model has two specific characteristics: First, a real-valued vector is generated at each state of our model. This is because the output/input of our model at each time point is a pair (or more generally multiple) of expression values. Second, the state transition diagram of our model has two types of states: the first state for relatively average expression values and the second for expression values that are different from the average. The parameters of the first state are fixed and not trained. This setting is useful to capture time points with expression values that are very different from the average, because they will be generated at the second states while others are generated at the first states. By using these two characteristics, our approach can identify an expression pattern, like that found in Figure 1. An important feature of our model is that the given pairwise (generally multiplewise) sequences can vary in length, because of the nature of hidden Markov models.

We have conducted a variety of experiments, using both synthetic and real datasets, to evaluate the effectiveness of our approach of finding the timing difference in two or more experimental conditions. The results obtained by synthetic datasets showed that our method could capture an embedded timing difference in a set of pairwise time series sequences. We then checked the difference in time series gene expressions of four different strains under a certain stress, using real microarray datasets. From the comparison of all six combinations of four different strains, our method could find a clear time series ordering in gene expression of four strains. Finally, we compared the effects of four different stresses on gene expression of a certain strain and found a clear timing difference in gene expression caused by two stresses. These results indicate that our method can identify timing differences in gene expression that are caused by different experimental factors.

2 METHOD

2.1 Notations

Let Y be a set of real-valued sequences, and let all sequences in Y have the same length. Let $N(Y)$ be the length of a sequence in Y, and K be the number of sequences in Y. In practice, Y is a set of time series microarray expression sequences, and K is the number of conducted time series microarray experiments. We call Y an *example* in this paper. Let $\mathbf{Y}$ be a set of Ys, and $|\mathbf{Y}|$ be the number of Ys in $\mathbf{Y}$. In practice, $|\mathbf{Y}|$ is the number of genes for which time series microarray experiments are conducted under K conditions. Figure 1 shows a simple example of $\mathbf{Y}$ with $|\mathbf{Y}| = 3$, $K = 2$, and $N(Y)$ is 5 for all $Y \in \mathbf{Y}$. We note that K must be kept the same in $\mathbf{Y}$, but $N(Y)$ not. That is, our model can deal with time series microarray datasets with different time points. In fact, in synthetic datasets of our experiments, we will deal with the case that $N(Y)$ takes a value from 6 to 10. Let N be $\max_{Y \in \mathbf{Y}} N(Y)$. In Y, let y_t be the K real-valued expression values at time t, which we call a *vector* in this paper. Each square in Figure 1 is a vector. Let $y_{t,k}$ be the k-th (expression) value of y_t. For example, in the first example of $\mathbf{Y}$ in Figure 1, $y_{1,1} = 1.2$ and $y_{2,2} = 1.5$. For simplicity, we sometimes just write y for y_t.

Let q be a state of our model, and let $Q = (q_1, \ldots, q_{|Q|})$ be a state transition of a given example in our proposed model, which will be described in detail later.

2.2 Definition of the proposed model

The proposed model is a special case of the so-called hidden Markov model (HMM). Our model has two types of probabilities: *state transition probability* $a_{i,j}$ for a transition from state i to j and *continuous value generation probability* $b_i(y)$ to generate real-valued vector y at state i, which satisfy that $\sum_j a_{i,j} = 1$ and $\int_{-\infty}^{+\infty} b_i(y)dy = 1$.

Let μ_i and v_i be real-valued vectors of size K, and $\mu_{i,k}$ and $v_{i,k}$ be the k-th values of vectors μ_i and v_i, respectively. Assuming that $y_{t,k}$ $(k = 1, K)$ are independent of each other, the probability b_i (y_t) is defined as a normal (Gaussian) distribution, which has μ_i as the *average* and v_i as the *variance*, as follows:

$$b_i(y_t; \mu_i, v_i) = \prod_k b_{i,k}(y_{t,k}; \mu_{i,k}, v_{i,k})$$

$$= \prod_k \left(\frac{1}{2\pi v_{i,k}}\right)^{1/2} e^{-\sum_k \frac{1}{v_{i,k}}(y_{t,k}-\mu_{i,k})^2}$$

Thus, totally, our model has three types of parameters: $a_{i,j}$, $\mu_{i,k}$ and $v_{i,k}$.

Figure 2 shows the state transition diagram of the proposed model. A state q of this model can be classified into two types, which we write F and G, called a *feature state* and a *control state*, respectively. The parameters μ_i and v_i at a control state are fixed and not trained, whereas the μ_j and v_j at a feature state are trained. Let M be the number of feature states in a given model.

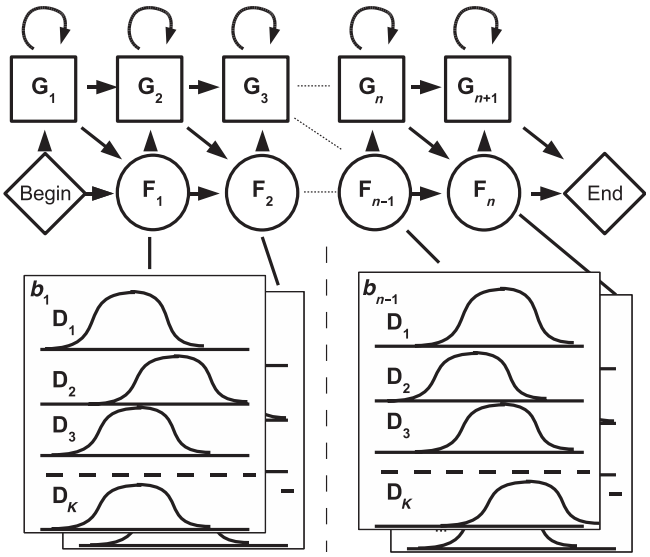

Fig. 2. A state transition diagram of our model, with n feature states and $n+1$ control states. A real-valued vector of size K is generated at each state, and each value of this vector is generated according to a normal distribution. This distribution is trained at a feature state, but not at a control state, where a prefixed normal distribution is used.

2.3 Likelihood computation

Given an example Y and a state transition Q, we can compute probability $P(Y, Q)$ as follows:

$$P(Y, Q) = a_{0,1} \prod_{t=1}^{|Q|} a_{q_t, q_{t+1}} b_{q_t}(y_t; \mu_{q_t}, v_{q_t}),$$

where q_0 and $q_{|Q|+1}$ are the begin and end states, respectively, where no values are generated.

The log-likelihood of an entire dataset $\mathbf{Y}$ is then given as follows:

$$L(\mathbf{Y}) = \sum_{Y \in \mathbf{Y}} \log P(Y) = \sum_{Y \in \mathbf{Y}} \log \sum_{Q} P(Y, Q) \quad (1)$$

2.4 Parameter estimation

A standard way to estimate the parameters of a probabilistic model is the maximum likelihood, i.e. to maximize the log-likelihood given in Equation (1). A popular approach of the maximum likelihood is a so-called EM (expectation-maximization) algorithm, by which it is guaranteed that we can find a local optimum. To estimate the parameters of our model, we use the EM algorithm, which iterates the following E- and M-steps alternately until some stopping condition is satisfied.

E-step: For each $Y \in \mathbf{Y}$, we compute two auxiliary probabilities, which we call *forward* and *backward probabilities*. The forward probability $\alpha_Y(j, t)$ is the probability that all expression values at time points 1 to t are already generated and the current state is j. Similarly, the backward probability $\beta_Y(i, t)$ is the probability that all expression values at time point t to the last time point of Y are already generated and the current state is i. The forward probabilities are computed recursively by increasing t from zero to the last time point, according to the following equation.

$$\alpha_Y(j, t+1) = b_j(y_{t+1}; \mu_j, v_j) \sum_i \alpha_Y(i, t) a_{i,j}$$

The backward probabilities are computed in the reverse order using the following equation as well.

$$\beta_Y(i, t) = \sum_j b_j(y_{t+1}; \mu_j, v_j) \beta_Y(j, t+1) a_{i,j}$$

M-step: Using the forward and backward probabilities computed in the E-step, we can update the three parameters in our model: $a_{i,j}$, μ_i and v_i, as follows:

$$\hat{a}_{i,j} = \frac{\sum_{Y \in \mathbf{Y}} \sum_t \alpha_Y(i, t) \beta_Y(j, t+1) a_{i,j} b_j(y_{t+1}; \mu_j, v_j)}{\sum_{Y \in \mathbf{Y}} \sum_t \alpha_Y(i, t) \beta_Y(i, t)}$$

$$\hat{\mu}_{i,k} = \frac{\sum_{Y \in \mathbf{Y}} \sum_t \alpha_Y(i, t) \beta_Y(i, t) y_{t,k}}{\sum_{Y \in \mathbf{Y}} \sum_t \alpha_Y(i, t) \beta_Y(i, t)}$$

$$\hat{v}_{i,k} = \frac{\sum_{Y \in \mathbf{Y}} \sum_t \alpha_Y(i, t) \beta_Y(i, t) (y_{t,k} - \mu_{i,k})^2}{\sum_{Y \in \mathbf{Y}} \sum_t \alpha_Y(i, t) \beta_Y(i, t)}$$

Initial values of the above iteration in our experiments are set up as follows: we first computed the variance of all expression values in a given dataset and then assigned it as initial values for $v_{i,k}$ at both control states and feature states. On the other hand, as initial values for $\mu_{i,k}$, we assigned zero for control states and some fixed large value, which is larger than zero, for each feature state.

2.5 Time and space complexities

Our state transition diagram in Figure 2 shows that the number of outgoing edges at a node is three at maximum. So, the number (space complexity) of $a_{i,j}$ can be almost linear in the number of states. Thus, in each iteration of the EM algorithm, the most time-consuming part is updating μ (or v) in the M-step. When we update each $\mu_{i,k}$ ($i = 1, \ldots, M, k = 1, \ldots, K$), we have to sum up $\alpha_Y(i, t)\beta_Y(i, t)y_{t,k}$ over all $Y \in \mathbf{Y}$ and t ($= 1, \ldots, N(Y)$). The maximum of t is N, and so the time complexity of our method is $O(M \cdot K \cdot |\mathbf{Y}| \cdot N)$. Our model generates a vector at each state, and so the above complexity is larger than that of a usual HMM by K, i.e. the size of a vector.

On the other hand, the space complexity of our method is kept the same as that by a standard HMM for real-valued sequences that have been used in some applications including speech recognition. That is, at first, a is almost linear in the number of states, and all μ, v, α and β stay at quadratic complexity (We note that we do not have to store α and β for each Y.). Thus, the space complexity of our method is $\max\{O(M \cdot K), O(M \cdot N)\}$.

2.6 Why the model works

Our model has two unique characteristics: First, our model generates a real-valued vector at each state, while a usual HMM generates only one real value (or symbol) at a state. Second, the probability b is trained at feature states only. It is not trained at control states where the parameters of b are fixed at some reasonable value like an average over all expression values in the given data.

Due to these two characteristics, our model works as follows: if the expression values at a time point are all rather normal (or average), they should be generated at a control state; otherwise they can be generated at a feature state. More concretely, if one or more expression values in the vector of a time point are very different from the average, this vector can be generated at a feature state. That is, our model attempts to

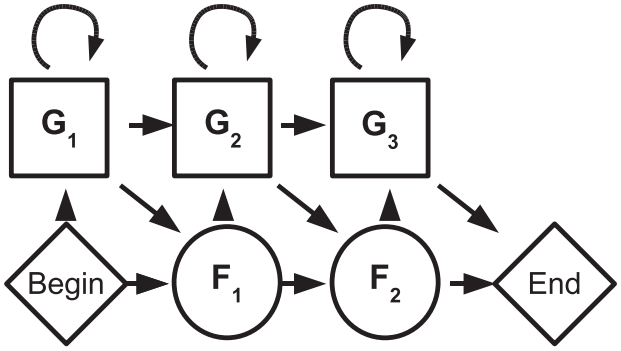

Fig. 3. A state transition diagram example of our model with only two feature states.

capture a time point with unusual (or unique) values, and these values will appear at feature states.

Figure 3 shows a simple example of the state transition diagram of our model with only two feature states. When we train this model by the data in Figure 1, the shaded areas (in Figure 1) with expression values of zero will be assigned to control states. On the other hand, the white squares in Figure 1 will be assigned to the two feature states, because the values in these squares are much higher than zero. More concretely, the case of a high value at Condition A and zero at Condition B will be assigned to F_1, and the reverse case will be assigned to F_2. This meets our purpose of finding the difference between a given set of pairwise (generally multiplewise) sequences. Finally, we note that by checking the parameter values of b at feature states of the trained model, we can easily see the difference between a given set of time series sequences.

3 TIME SERIES DATA

3.1 Synthetic data

We first evaluated our model using two types of synthetic datasets, which we call SD1 and SD2. Each dataset had 100 examples, and each example had a pair of real-valued sequences, assuming that two time series expression values are measured under two different experimental conditions, which we name Condition A and Condition B for further explanations. Thus, $|\mathbf{Y}| = 100$ and $K = 2$. The length of a sequence ranged from six to ten and was randomly chosen according to the uniform distribution. Of course, the length of two sequences in a pair was kept the same.

A time point of each sequence randomly takes a value from -1 to $+1$, except some time points that randomly take higher values ranging from $+1$ to $+4$. We note that this high value simulates that a gene is highly expressed at this time point. In SD1, only one randomly chosen time point of a sequence takes a high value, and this high value appears in the first half for Condition A and in the last half for Condition B. Figure 4a shows a schematic example of a dataset of SD1. On the other hand, randomly chosen three continuous time points take high values, and they start in the first half in Condition A and in the last half of Condition B. This is also shown in Figure 4b as a schematic example.

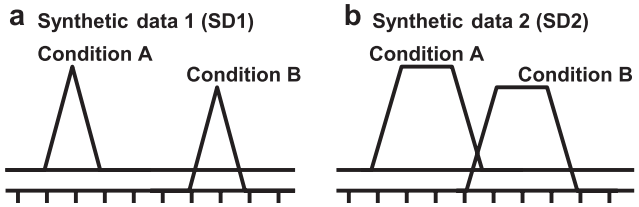

Fig. 4. Schematic figures of two synthetic datasets. (**a**) SD1: A randomly chosen high-valued point is randomly in the first half of a sequence for Condition A and randomly in the last half for Condition B. (**b**) SD2: Randomly chosen three continuous high values

Table 2. Yeast microarray dataset generated from GSE3406

Strains	*S.cerevisiae* (Sc), *S.paradoxus* (Sp), *S.mikatae* (Sm), *S.kudriavzevii* (Sk)
Stresses	37°C heat shock (Heat shock), 0.3 mM H_2O_2 (Oxidative stress), Glucose to glycerol (glycerol), 0.02% MMS (DNA damage)
Time points	10, 20, 30, 45, 60, 90 min

Table 3. A list of genes used for generating real microarray datasets in Table 2

YAL028W	YBL075C	YBR001C	YBR072W	YBR082C
YBR126C	YCR021C	YDL190C	YDR001C	YDR017C
YDR074W	YDR143C	YDR171W	YDR184C	YDR214W
YDR258C	YER011W	YER012W	YER103W	YFL053W
YFR019W	YGR088W	YGR234W	YGR253C	YHL028W
YHR043C	YHR104W	YIL033C	YIL101C	YJL001W
YJR032W	YKL062W	YKL201C	YLL010C	YLL026W
YLL039C	YLR019W	YLR266C	YML014W	YML070W
YML100W	YMR037C	YMR169C	YMR186W	YMR251W-A
YMR261C	YNL160W	YNL234W	YNL281W	YOL052C-A
YOL151W	YOR010C	YOR324C	YPL223C	YPL240C
YPR026W				

3.2 Real microarray data

We used GSE3406 (Tirosh *et al.*, 2006) of the Gene Expression Omnibus (GEO) database (http://www.ncbi.nlm.nih.gov/geo/) (Edgar *et al.*, 2002). This dataset included time series expression values of four different yeast strains measured under four different stresses. Table 2 shows a summary of this dataset. From GSE3406, we generated datasets with $K = 2$, each of which was a set of pairwise sequences under two different factors, i.e. stresses or stains, keeping other factors the same. The purpose of this experiment is to find the difference in gene expression under different stresses or strains. We then focused on genes, which are categorized into 'response to stress' in the Gene Ontology (Gene Ontology Consortium., 2006), and selected all of them from the SGD database (Christie *et al.*, 2004). Table 3 shows the list of 56 genes we used. For each pair of stresses or strains, we generated a dataset by using

only genes of which at least one expression value is higher than 1.0 in GSE3406. This means that $|\mathbf{Y}|$, i.e. the size of a dataset, varies.

4 RESULTS

4.1 Synthetic data

We used $M=2$ in this experiment, i.e. the state transition diagram in Figure 3. Table 4a and b show the parameter values obtained by applying our method to the two synthetic datasets, SD1 and SD2, respectively. We note that, in this table, a μ value is bolded if it is higher than that at the other feature state under the same condition. For example, the μ value at state F_1 for Condition A in SD1 is bolded, because it was 2.83, which is higher than 0.01, i.e. the μ value at state F_2 for Condition A, by more than 1.0. This indicates that this state captured a time point with a high value for Condition A (and a low value for Condition B). From the table, we can see that the μ of F_1 was high for Condition A and around zero for Condition B, and became almost zero for Condition A and high for Condition B. This indicates that our model captured the embedded pattern in SD1, i.e. that a high value appears first in Condition A and then in Condition B.

These results and observations were basically true of SD2. However, interestingly, the μ of state F_1 for Condition A was 3.83 and that of state F_2 for Condition B was 3.47, indicating that these values were higher than those in SD1. We think that this is from the following reason. At first, the model used in this experiment has only two feature states, and high values in SD2 appear first in Condition A and then in Condition B, indicating that one of the two feature states can be given to each condition. More concretely, one of the three high continuous values in a sequence of SD2 was assigned to one feature state of our model. Thus, the highest value of the three high values was assigned to a feature state, since the μ of a control state was fixed at around zero (i.e. a very low value). Finally, the μ of SD2 must be higher than that of SD1.

From these results on synthetic data, we can say that our method worked favorably in finding expression values that are different between given two sets of sequences.

Table 4. Estimated parameters from synthetic data

	(a) Synthetic Data 1 (SD1)		(b) Synthetic Data 2 (SD2)	
	F_1	F_2	F_1	F_2
Condition A				
μ	**2.83**	0.01	**3.83**	0.01
v	1.16	0.31	0.65	0.36
Condition B				
μ	−0.06	**2.94**	−0.04	**3.47**
v	0.33	1.31	0.39	1.03

4.2 Real microarray data: difference between yeast strains under a stress

Out of the four stresses, we focused on the 37°C heat shock. We first show the result obtained by $M=2$, corresponding to the transition diagram in Figure 3.

A same experiment is conducted twice for *Saccharomyces paradoxus* in GSE3406, meaning that we can have two datasets, which we call Sp and Sp2, for a set of sequences of *S.paradoxus*. We first examined the difference in gene expression between Sp and Sp2, which are obtained for the same strain under the same stress, to check the variability/stability of microarray expression values. Table 5 shows the trained parameter values at feature states in this combination. From the table, we can see that the μ of Sp was almost the same as that of Sp2 at F_1, whereas they were not always the same at F_2. In fact, the μ of Sp was larger than that of Sp2 by around 0.5, indicating that a difference in gene expression of 0.5 can happen even when we compare two datasets obtained by the same experiment. Thus we can say that when we compare two datasets obtained under different conditions, we have to focus on a larger difference, say 1.0 or more. We think that 1.0 would be a natural threshold to judge that a difference happens in gene expression between two different experimental factors (Speed *et al.*, 2003).

Table 5. The difference in gene expression between duplicated sets of *S.paradoxus*: Sp and Sp2, under the 37°C heat shock

		F_1	F_2
Sp	μ	1.56	1.97
	v	1.50	1.37
Sp2	μ	1.60	1.45
	v	2.26	1.11

Table 6. The difference in gene expression between different strains under the 37°C heat shock. Sc, Sp, Sm and Sk stand for *Saccharomyces cerevisiae*, *Saccharomyces paradoxus*, *Saccharomyces mikatae* and *Saccharomyces kudriavzevii*, respectively

		F_1	F_2			F_1	F_2
(a)				(b)			
Sc	μ	0.36	**1.92**	Sc	μ	1.02	**2.01**
	v	0.35	1.55		v	0.34	1.22
Sp	μ	1.81	0.98	Sm	μ	**2.02**	0.23
	v	1.33	1.16		v	0.68	0.38
(c)				(d)			
Sc	μ	0.98	1.85	Sp	μ	1.46	1.79
	v	0.21	1.38		v	1.69	1.21
Sk	μ	1.60	1.45	Sm	μ	2.14	1.42
	v	1.38	0.58		v	0.68	0.42
(e)				(f)			
Sp	μ	1.64	1.31	Sm	μ	**1.99**	0.90
	v	0.78	1.22		v	1.89	0.29
Sk	μ	1.72	1.70	Sk	μ	0.49	**2.12**
	v	1.09	0.31		v	0.20	0.98

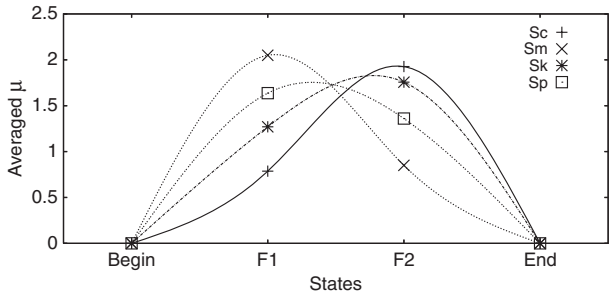

Fig. 5. A schematic picture of the timing in gene expression of four strains under the Heat shock stress.

We then focused on four strains under the 37°C heat shock. Table 6 shows the trained parameters at feature states in all possible six combinations of the four strains. We note that a μ value is bolded if it is higher than that at the other feature state under the same condition by more than around 1.0. In this table, (a) shows the comparison between *Saccharomyces cerevisiae* (Sc) and *S.paradoxus* (Sp). The μ for Sp was significantly high at F_1 but decreased to a low value at F_2, while that for Sc was first low at F_1 and then increased to a significantly high value at F_2. This indicates that 'gene expression tended to start earlier in Sp than in Sc under the heat shock stress'. We can see similar results in (b) and (e). The results in (c) and (d) were not so clear, but we would be able to say that gene expression in Sk seemed to tend to start earlier than in Sc. This is true of (d). On the other hand, the μ values in (e) imply that there are not such significant timing differences in gene expression between Sp and Sk. We can summarize these results as follows:

(a) Gene expression tended to start earlier in Sp than in Sc.
(b) Gene expression tended to start much earlier in Sm than in Sc.
(c) Gene expression tended to start a little earlier in Sk than in Sc.
(d) Gene expression tended to start a little earlier in Sm than in Sp.
(e) No significant timing difference in gene expression was found between Sp and Sk.
(f) Gene expression tended to start earlier in Sm than in Sk.

Figure 5 shows a summary picture of the obtained parameter values. To draw this picture, we first computed the average over the three μ values at F_1 (F_2) of a corresponding strain in Table 6. We then plotted the average μ values at F_1 and F_2 and drew the line connecting these two and zero at Begin and End states. Finally, this line was smoothed using spline interpolation. We note that Begin, F_1, F_2 and End are located at an equal interval in this figure. This figure confirms the above six observations, implying that gene expression tended to start in the timing ordering of Sm $\rightarrow$ Sp $\rightarrow$ Sk $\rightarrow$ Sc. We note that this ordering agrees with all the above six observations without any contradictions. We further checked a set of genes that clearly agrees with each tendency of (a) to (f) except (e), in which no significant difference was found.

Table 7. A list of genes clearly satisfying the tendencies of (a), (b), (c), (d) and (f) in the experiment of four strains under the 37°C heat shock

(a)				
YDR074W	YGR088W	YGR234W	YHR104W	YML070W
YML100W	YNL160W			
(b)				
YDR074W	YER012W	YGR088W	YGR253C	YHR043C
YHR104W	YJL001W	YML070W	YML100W	YMR251W-A
YMR261C	YNL160W			
(c)				
YDR074W	YIL033C	YML070W	YML100W	YNL160W
(d)				
YDR074W	YDR171W	YER012W	YGR253C	YHR104W
YJL001W	YML070W	YMR251W-A	YOL151W	
(f)				
YDR074W	YDR171W	YER012W	YGR088W	YGR253C
YHR104W	YIL101C	YJL001W	YML070W	YMR251W-A
YMR261C	YOL151W			

Table 7 shows the lists of these genes. When we selected each of these genes, we first checked the highest value of each of the two given sequences of a gene, and this gene was selected if the following two conditions were satisfied: (1) The two highest values are both larger than 1.0. (2) The two time points that provide these highest values are rightly ordered, like that the time point of Sp is earlier than that of Sc.

We then conducted experiments for $M=3$ using the same setting as done for $M=2$. The trained parameter values, which are in the supplementary information, show that the results were almost similar to the case of $M=2$. Thus, we skip the detailed explanation for $M=3$ due to space limitation.

4.3 Real microarray data: difference in gene expressions of *S.cerevisiae* under different stresses

We then focused on *S.cerevisiae* to check the difference between four types of stresses, i.e. 37°C heat shock (heat shock), H_2O_2 (Oxidative stress), a transfer of medium from glucose to glycerol (Glycerol) and 0.02% MMS (DNA damage). Table 8 shows the trained parameter values at features states of all six combinations of the above four different stresses. As in the case of different strains, this table shows the timing difference in gene expression between two different stresses. From (a), we can see that at F_1, the μ was high for Heat shock and low for Oxidative stress, while they were reversed at F_2. This result clearly indicates that gene expression under Heat shock tended to start earlier than that under Oxidative stress. A similar type of conversion in the timing of gene expression were slightly shown in (b) and (e), but they were not necessarily significant. Similarly, such a clear conversion was not found in (c), (d) and (f). Figure 6 shows a summary picture computed from the parameter values in Table 8. This picture was drawn in the same manner as done in drawing Figure 5. From the figure, we can see that gene expression under Heat shock tended to start earlier than that under other stresses, Oxidative stress in

Table 8. The difference in gene expression of *S.cerevisiae* between different stresses

		F_1	F_2			F_1	F_2
(a)				(b)			
Heat shock	μ	**1.98**	0.15	Heat shock	μ	**2.19**	0.89
	v	0.58	0.37		v	1.27	0.22
Oxidative stress	μ	0.92	**1.96**	Glycerol	μ	1.86	2.44
	v	0.56	1.48		v	1.33	1.33
(c)				(d)			
Heat shock	μ	2.26	1.59	Oxidative stress	μ	0.98	1.83
	v	0.74	0.67		v	0.15	1.54
DNA damage	μ	1.55	1.60	Glycerol	μ	2.31	2.33
	v	0.48	0.54		v	1.35	1.24
(e)				(f)			
Oxidative stress	μ	0.46	**2.13**	Glycerol	μ	2.59	2.36
	v	0.43	1.35		v	1.98	1.49
DNA damage	μ	1.57	1.42	DNA damage	μ	1.57	1.37
	v	0.34	0.45		v	0.36	0.37

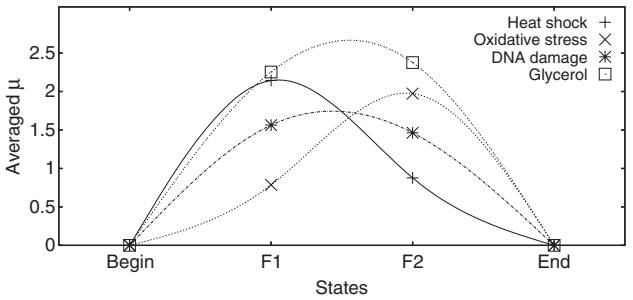

Fig. 6. A schematic picture of the timing in gene expression of *S.cerevisiae* under the four different stresses.

particular. In other words, gene expression tended to start earlier under Heat shock, later under Oxidative stress and normal under DNA damage and Glycerol. Another finding from this figure is that gene expression under Glycerol is always higher than that under the other three stresses. Overall, this result indicates that our method is useful in understanding the time series ordering in gene expression under different stresses/species.

We then performed an experiment for the case of $M=3$ to check the difference in gene expression of different stresses. The result of $M=3$, which are in the supplementary information, was almost the same as that of $M=2$, and so we skip the detailed explanation due to space limitation.

5 CONCLUSION AND DISCUSSION

We have developed a systematic approach to capture the differences in microarray time series sequences obtained by different factors, based on the learning of hidden Markov models. We emphasize that our model has the following two unique features. First, a real-valued vector, which corresponds to a set of expression values at a time point, is generated at each state. Second, our model has an unique state transition diagram, which is designed to identify a time point with distinctive expression values from an average value. Based on these two characteristics, our method allows to capture the differences in a given set of time series sequences.

We have conducted a series of experiments to evaluate the effectiveness of our method using synthetic datasets as well as real microarray datasets. In the synthetic datasets, one apparent timing difference was embedded in gene expression as a pattern. The parameters of the trained probabilistic model showed that our method clearly captured the embedded pattern in gene expression. The experiments using real microarray datasets showed that our method could identify the timing differences in gene expression, which are caused by external experimental factors. The typical two results are as follows: (1) Under the heat shock stress, gene expression tended to start in the ordering of in *S.mikatae*, *S.paradoxus*, *S.kudriavzevii* and *S.cerevisiae*. (2) Gene expression of *S.cerevisiae* tended to start earlier under heat shock stress than under oxidative stress. We stress that these findings are very useful for biologists who are conducting time series microarray experiments to compare the experimental conditions (or species) like environmental stresses.

In our experiments, the result of $M=3$ was almost the same as that of $M=2$. This is probably because the number of time points in our datasets is only six. If we use a larger number of time points, a larger M may be more useful to analyze the data. This would be possible future work. Another possible direction in future experiments is to use a larger number of sequences as one example. That is, by increasing K, a larger number of experimental conditions can be combined, and we can see the relations by two or more conditions at once. However, by increasing K, the number of timing differences between triplewise or more sequences will also increase, and they will not be captured by a small number of feature states easily. This problem was avoided by focusing on only a pair of sequences in this paper. Another plausible future direction is to deal with periodic patterns in time series gene expression, which already have often been found in time series microarray data. It would be an interesting research theme to design another state transition diagram that may be more useful in capturing the periodic timing difference in gene expression of different experimental factors.

ACKNOWLEDGEMENTS

The authors would like to thank Ichigaku Takigawa, Raymond Wan, Shanfeng Zhu and Motoki Shiga of Kyoto University, and Takayuki Onuma, Reina Matsumoto, Satoshi Yoshida and Osamu Kobayashi of Kirin Brewery for fruitful discussions and valuable comments.

Conflict of Interest: none declared.

REFERENCES

Aach,J. and Church,G.M. (2001) Aligning gene expression time series with time warping algorithm. *Bioinformatics*, **17**, 495–508.

Alter,O. *et al.* (2000) Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. USA.*, **97**, 10101–10106.

Arbeitman,M.N. *et al.* (2002) Gene expression during the life cycle of Drosophila melanogaster. *Science*, **297**, 2270–2275.

Bar-Joseph,Z. (2004) Analyzing time series gene expression data. *Bioinformatics*, **20**, 2493–2503.

Borgwardt,K. *et al.* (2006) Class prediction from time series gene expression profiles using dynamic systems kernels. *PSB*, **11**, 547–558.

Chen,D. *et al.* (2003) Global transcriptional responses of fission yeast to environmental stress. *Mol. Biol. Cell*, **14**, 214–229.

Christie,K.R. *et al.* (2004) Saccharomyces Genome Database (SGD) provides tools to identify and analyze sequences from Saccharomyces cerevisiae and related sequences from other organisms. *Nucl. Acids Res.*, **32**(Database issue), D311–D314.

Costa,I.G. *et al.* (2005) The Graphical Query Language: a tool for analysis of gene expression time-courses. *Bioinformatics.*, **21**, 2544–2545.

Durbin,R. *et al.* (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids,* Cambridge University Press, Cambridge, UK.

Edgar,R. *et al.* Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucl. Acids Res.*, 2002 Jan **30**, 207–210.

Ernst,J. *et al.* (2005) Clustering short time series gene expression data. *Bioinformatics*, **21**, i159–i168.

Ernst,J. and Bar-Joseph,Z. (2006) STEM: a tool for the analysis of short time series gene expression data. *BMC Bioinformatics*, **7**, 191.

Filkov,V. *et al.* (2002) Analysis techniques for microarray time-series data. *J. Comput. Biol.*, **9**, 317–330.

Gasch,A.P. *et al.* (2000) Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell*, **11**, 4241–4257.

Gene Ontology Consortium. (2006) The Gene Ontology (GO) project in 2006, *Nucl. Acids Res.*, **34**(Database issue), D322–D326.

Guillemin,K. *et al.* (2002) Cag pathogenicity island-specific responses of gastric epithelial cells to Helicobacter pylori infection. *Proc. Natl. Acad. Sci. USA*, **99**, 15136–15141.

Park,T. *et al.* (2003) Statistical tests for identifying differentially expressed genes in time-course microarray experiments. *Bioinformatics*, **19**, 694–703.

Rabiner,L.R. and Juang,B. (1986) An introduction to hidden Markov models. *ASSP Magazine of the IEEE*, **3**, 4–16.

Schliep,A. *et al.* (2003) Using hidden Markov models to analyze gene expression time course data. *Bioinformatics*, **19**, i255–i263.

Schliep,A. *et al.* (2004) Robust inference of groups in gene expression time-courses using mixtures of HMMs., *Bioinformatics*, **20**, i283–i289.

Speed,T. (2003) *Statistical Analysis of Gene Expression Microarray Data,* CRC Press.

Storey,J.D. *et al.* (2005) Significance analysis of time-course microarray experiments. *Proc. Natl. Acad. Sci. USA*, **102**, 12837–12842.

Tirosh,I. *et al.* (2006) A genetic signature of interspecies variations in gene expression. *Nat. Genet.*, **38**, 830–834.