

# Identifying Gene Clusters and Regulatory Themes using Time Course Expression Data, Hidden Markov Models and Transcription Factor Information

Karen Lees<sup>1\*</sup>, Jennifer Taylor<sup>2</sup>, Gerton Lunter<sup>2</sup>, Jotun Hein<sup>2</sup>

<sup>1</sup>LSI DTC, Peter Medawar Building, University of Oxford, 2 South Parks Road, Oxford OX1 3TG, <sup>2</sup>Department of Statistics, Oxford Centre for Gene Function, University of Oxford, 1 South Parks Road, Oxford OX1 3TG

## ABSTRACT

**Motivation:** The development of microarrays has allowed the gene expression of thousands of genes to be simultaneously observed. Methods are required that analyse this expression data in order to discover information about gene function and regulatory mechanisms. The regulatory information would be very useful for the understanding of biological processes and the changes to these processes associated with disease and development. Methods of clustering genes that show similar expression patterns using hidden Markov models have recently been developed for time series data. We extend this previous work by suggesting some improvements as well as developing a strategy to incorporate transcription factor information directly into the model.

**Results:** We applied the method to a large dataset containing 22283 genes and 5 time points. The results showed that expression patterns had been sensibly clustered, however many further improvements and analysis of the method need to be made.

**Contact:** karen.lees@linacre.ox.ac.uk, taylor@stats.ox.ac.uk

## 1 INTRODUCTION

### 1.1 Gene Expression Data

*1.1.1 Production of Expression Data* Gene expression measurements can be obtained for thousands of genes simultaneously and very quickly using microarray technology. In a cell, genes are transcribed into mRNA molecules which in turn can be translated into proteins, and it is these proteins that perform biological functions, give cellular structure and in turn regulate gene expression. For a gene to be expressed it means that it is being transcribed into mRNA. Therefore even

though cells may contain the same genes, it is their expression that dictates the proteins present in the cell and therefore the specialisation of a cell. In turn the presence or absence of certain proteins called transcription factors dictate where and when genes are expressed, so each gene is part of a regulatory network indirectly affecting the expression of itself and other genes at future time points.

In order to measure gene expression the levels of mRNA that correspond to each gene are observed using microarray methodology. This involves reverse transcribing the mRNA into cDNA molecules that have incorporated fluorescently tagged bases. A microarray consists of many thousands of short, single stranded sequences, each immobilised as individual elements on a solid support, that are complementary to the cDNA strand representing a single gene. The fluorescently labelled cDNA mixture is then exposed to the microarray so that the cDNA can bind (hybridise) to the strands attached to the microarray chip. The cDNA strands on the microarray are in vast excess to those in the fluorescently labelled mixture so that the amount of labelled DNA bound reflects the proportion in the mixture. Excess mixture that has not bound is then washed off, leaving only the strands the hybridised strands. Using a laser, the fluorescent molecules in the cDNA can be seen and the intensity for each gene can be measured. The greater the intensity the more mRNA is present that corresponds to a particular gene and therefore the greater the gene expression is implied.

*1.1.2 Purpose of Microarray Expression Data* There are two main experimental designs for microarrays. The first compares different samples, for example comparing healthy with diseased cells, different tissue types or cells under different conditions. The aim of this data is to discover how the gene expression differs between the samples and hence which gene behaviours can be associated with different conditions. The second produces time series data, which samples the same cell type over a series of time points, for example

\*to whom correspondence should be addressed

---

different time points during a cell cycle or at different stages of differentiation in a cell. In order to produce time series data the cells should be synchronised, such that the expression level given in the microarray represents the expression level at that point in a cell cycle and not a mixture of different points.

*1.1.3 Analysis of Expression Data* The analysis of microarray data aims to infer biological function of genes, regulatory networks, and identify genes that show similar expression patterns during a cell cycle or in response to a change in the cells environment. Identification of gene regulation and function should lead to a deeper understanding of genetic diseases, by discovering regulatory pathways and how they are perturbed. This is not easy due to the complexity of the regulation of the transcriptome. Large differences in gene expression can be observed from expression data analysis but these differences may really be a symptom of a diseased cell rather than the actual cause of the disease. Therefore understanding of regulatory pathways and transcription factor activity is vital to understand the causes of disease and how cell function can be effected by different environments and at different stages during it life.

## 1.2 Clustering Techniques for Gene Expression Data

The aim of clustering techniques is to analyse expression data by clustering genes that have similar expression profiles. These clusters can be then used to identify genes with similar changes and therefore possible similarities in the regulatory network.

Schliep *et al.* (2003) provide a good summary of clustering methods that have been used for analysing expression data. Two main approaches are distinguished:

- 1.Distance based approaches that have no dependencies between time points such as k-means clustering and singular value decomposition. These have been used successfully for distinguishing genes that differ between different samples. In these approaches distance measures are defined between objects and an objective function of these distance measure is minimised when the optimal clustering is found. However with time course data it is possible to change the ordering of the time points and get the same clustering results. Schliep *et al.* (2003) assert in their paper that taking account of the time dependencies should provide a higher quality of clustering for time course data.
- 2.Model based approaches that encode time dependencies and use statistical models to represent the clusters. These techniques calculate the likelihood of an expression profile given a cluster model and the aim is to find the assignment of profiles to clusters that maximises the likelihood.

In summary, model based approaches representative cluster models are defined and then the probability that the data fits a model is calculated, rather than just looking at similarities between data points. The main task of a model based approach is therefore finding a model that describes the data sufficiently well.

The clustering of time course data would ideally cluster genes that show similar expression patterns but be robust to errors and noise. Also it would be good to allow genes to be present in more than one cluster, as many genes do have multiple functions, so it makes more sense biologically to allow genes to appear in more than one cluster (Ji *et al.*, 2003).

There have been different approaches have been used to model time series data. These include modelling clusters as spline curves (Bar-Joseph *et al.*, 2002), none linear regression splines (Holmes *et al.*, 2004), autoregressive curves (Ramoni *et al.*, 2002), piece wise linear functions (Filkov *et al.*, 2002).

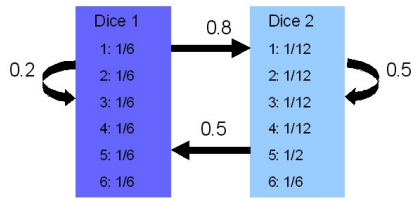
Model based approaches should be more robust to errors and noise and allow for some differences in the behaviour of a gene profile, whereas the distance based measures can be too sensitive to errors. Another advantage of the model based approaches is that they give a measure of likelihood that the data fits the model and therefore how much confidence we can have in the results.

## 1.3 Hidden Markov Models

Hidden Markov Models (HMMs) are statistical models that model a system being in a state at a particular time point and the transitions between states. Symbols can be emitted from each state with certain probabilities. HMMs are useful when sequences of emissions symbols are known and knowledge of the states a system can be in, but it would be useful to find information about the sequence of states that produced those emissions.

Durbin *et al.* (1998) uses the example of two different dice used in a casino. Each dice can be modelled as a hidden state in a Markov model, with transition probabilities to represent changing between dice. The number on the dice after each throw is represented by emission symbol from 1,2,3,4,5 or 6. It is known that one dice is fair, and so has equal emission probabilities from each number and one dice is biased, so that is it more likely to show some numbers than others. An example of this is given in figure 1 .Therefore given an sequence of emission symbols and the information about the dice it is possible to infer information about which dice(state) was used for each throw.

If the emission probabilities and the transition probabilities are known then the forward backward algorithm (see algorithm 1) can be used to calculate the probability,  $P(x)$ , that the model produced the sequence  $x$ . The forward backward



**Fig. 1.** A simple HMM with two states, one representing a fair dice and one for a biased dice. A HMM is represented by a set of states, transition probabilities and emission probabilities.

probabilities can also be used to calculate  $P(\pi = k|x)$ , the probability of being in state  $k$  when the  $i$ th emission was produced. A more detailed explanation of hidden Markov models and the forward backward algorithm can be found in Rabiner (1989) and Durbin *et al.* (1998).

This is fine when all the probabilities are known, but if some or all of them are unknown a method of estimating them is required. The Baum-Welch algorithm, an EM algorithm, is usually used for HMM parameter estimation. Baum-Welch initialises parameters and iteratively re-estimates them using the forward backward values. Counts  $A_{kl}$  and  $E_k(b)$  are estimated for each transition and emission probability equations (1) and (2). Once these have been summed over all sequences, the parameters  $a_{kl}$  and  $e_k(b)$  are re-estimated using equations (3) and (4), which standardise the probabilities so that they sum to one.

$$A_{kl} = \sum_j \frac{1}{P(x^j)} \sum_i F_k^j(i) a_{kl} e_l(x_{i+1}^j) B_l^j(i+1) \quad (1)$$

$$E_k(b) = \sum_j \frac{1}{P(x^j)} \sum_{i|x_i^j=b} F_k^j(i) B_k^j(i) \quad (2)$$

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \quad (3)$$

$$e_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b')} \quad (4)$$

### Forward Algorithm

$$\begin{aligned} \text{Initialisation (i=0):} \quad & F_S(0) = 1, F_k(0) = 0 \quad \text{for all } k \\ \text{Recursion (for i=1...L):} \quad & F_l(i) = e_l(x_i) \sum_k F_k(i-1) a_{kl} \\ \text{Termination:} \quad & P(x) = \sum_k F_k(L) a_{kE} \end{aligned}$$

### Backward Algorithm

$$\begin{aligned} \text{Initialisation (i=L):} \quad & B_k(L) = a_{kE} \quad \text{for all } k \\ \text{Recursion (for i=L-1,...,1):} \quad & B_k(i) = \sum_l a_{kl} e_l(x_{i+1}) B_l(i+1) \\ \text{Termination:} \quad & P(x) = \sum_l a_{Sl} e_l(x_1) B_l(1) \end{aligned}$$

### Probability that $i$ th state was $k$

$$P(\pi_i|x) = \frac{F_k(i) B_k(i)}{P(x)}$$

where:

$k, l$  = hidden states

$S$  = start state

$E$  = end state

$\pi$  = the state sequence

$L$  = the length of the emission sequence

$x_i$  = the  $i$ th symbol in the emission sequence

$a_{kl}$  = the transition probability from state  $k$  to  $l$

$e_k(b)$  = the emission probability of symbol  $b$  from state  $k$

$F_k(i)$  = the forward probability that state  $k$  emitted the  $i$ th symbol

$B_k(i)$  = the backward probability that state  $k$  emitted the  $i$ th symbol

### Algorithm 1: Forward Backward Algorithm

The re-estimation of parameters is repeated until the log likelihood (5) of the sequences given the model converges to a maximum value. This should happen when the parameters give the best model for the data or when a local maximum is reached. Therefore the Baum-Welch estimation process may need to be repeated several times to that the global maximum is found rather than a local maximum for the parameters.

$$l(x^1, \dots, x^n | \theta) = \sum_{j=1}^n \log P(x^j | \theta) \quad (5)$$

Microarray expression data can be thought of as being produced from several HMMs, each representing a particular expression profile or cluster of genes. It is unknown how many HMMs have produced the data and what the emission or transition probabilities are for any of the models, all of these need to be calculated and each gene needs to be allocated to a cluster that describes it best. The hidden information is there the cluster that each gene belongs to.

Schliep *et al.* (2003) used the sequences that were best described by a HMM for the re-estimation of that HMM's parameters. Ji *et al.* (2003) used a weighted Baum-Welch trainer to solve this problem, in which HMMs were separately trained to represent  $w$  expression patterns. This paper proposes a third way in which a larger HMM is built from  $w$  smaller HMMs that each represent a cluster. A path through the HMM can only go via one of the smaller ones, so the path that describes the sequence best gives the correct cluster for the sequence. These methods are described in more detail in later sections.

#### 1.4 Using Hidden Markov Models to analyse gene expression data

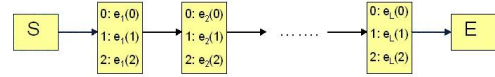
*1.4.1 Approach 1 - Schliep et al. (2003)* The algorithm represented each cluster by a HMM and randomly partitioned the sequence data between the clusters. Then standard iterative procedures such as Baum-Welch were used to adjust the HMM parameters until the maximum likelihood of the sequences given the models was found. At each step the parameters were re-estimated and the sequences re-assigned to the cluster for which was most likely.

In Schliep *et al.* (2003) partially supervised learning was used. The HMM parameters were initialised to model a set of expected biological clusters. Labelled profiles that represented these clusters were added and could be used at each re-estimation step to keep the models relatively stable as these profiles were fixed to a cluster rather than being re-assigned at each step. This is good as it will mean that meaningful clusters could be found but it involves having some prior knowledge of the biological data to be analysed and also is not fully automated.

Schliep *et al.* (2003) also had a strategy to automatically decide the best number of clusters. This strategy was to delete any clusters with few profiles and re-assign the profiles to remaining clusters. If there were clusters with too many profiles, a new cluster was created and then some of the profiles assigned to this new cluster.

*1.4.2 Approach 2 - Ji et al. (2003)* In the approach taken by Ji *et al.* (2003) expression sequences were converted to fluctuation sequences, then the fluctuation sequences were used in the training of a set of HMMs that represented clusters.

The expression profiles were first normalised and then transformed into fluctuation sequences. For an expression sequence of  $L$  time points a fluctuation sequence was produced of length  $L-1$  which contained only values 0,1 and 2. More formally for an expression level at time point  $i$ , denoted by  $E_i$ , the fluctuation level between time  $i$  and  $i+1$  is denoted by  $S_i$  and can be calculated by equation (6):



**Fig. 2.** A Markov chain representing a cluster.  $S$  is the start state,  $E$  is the end state. States 1 to  $L$  have only one transition but can emit symbols 0,1 or 2. State  $i$  emits the  $i$ th symbol in the fluctuation sequence.

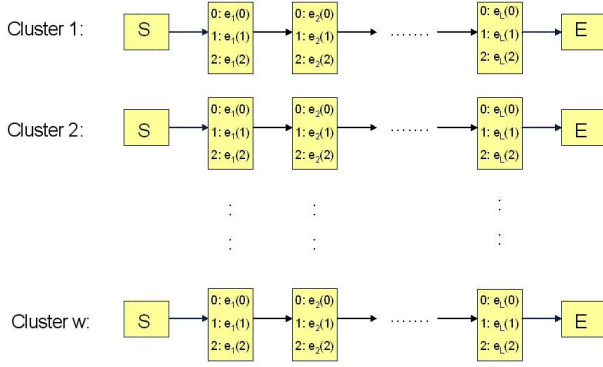
$$S_i = \begin{cases} 0 & \text{if } |E_i - E_{i+1}| < a \\ 1 & \text{if } E_{i+1} - E_i \geq a \\ 2 & \text{if } E_i - E_{i+1} \geq a \end{cases} \quad 1 \leq i \leq L \quad (6)$$

Therefore if the expression has been up regulated by more than a threshold value  $a$  between any time points this will be represented by a 1 in the fluctuation sequence and similarly a 2 represents down regulation by more than  $a$ . These fluctuations were then modelled as emissions sequences produced from a HMM.

Ji *et al.* (2003) chose a simple HMM topology, a Markov chain model, containing  $L$  states which can only be traversed in order and each state can emit a character from the alphabet  $\{0,1,2\}$  with certain probabilities to represent each cluster (see figure 2).

For  $w$  clusters,  $w$  Markov chains were randomly initialised with one chain representing one cluster (Figure 3). The parameters of each Markov chain were then trained independently using a weighted Baum-Welch algorithm on the entire set of fluctuation sequences. The weighting was necessary so that each model could specialise to represent a different cluster for a set of expression profiles. Without the weighting all Markov chains would converge to have the same parameters based on the entire set rather than be specialised to represent a gene cluster. The weights were proportional to the probability of a sequence being emitted from that chain, so those that fitted the chain best were given the most weight in re-estimating the parameters. Unlike Schliep *et al.* (2003) the learning was not partially supervised as the models were randomly initialised rather than being produced to model certain expression behaviours, and no labelled data was added as examples of each cluster. Also the weighted Baum-Welch algorithm was used rather than training each model on a subset of the sequences.

After training all of the models the probability of each sequence coming from each model was calculated. Then the sequences were allocated to clusters (represented by the models) based on these probabilities. The clustering algorithm was run 100 times to take account of local maxima problems of the Baum-Welch algorithm. This was then repeated with varying numbers of clusters from 2 to 50, as the cluster size was not known in advance.



**Fig. 3.** Multiple HMMs used to represent each multiple clusters. Each HMM has the same topology but initialised with different random parameters and will be trained to represent different expression patterns.

The method was used on three datasets and compared results to two other clustering algorithms, SOM and k-means. The datasets ranged in size from 74 genes to 613 genes and had either 17 or 18 time points. The results were then compared using two performance criteria.

The first of the performance criteria was the Rand index, equation (7), which used external data, a 'gold standard' partitioning of the dataset to compare results to. A Rand index of 1 means the partitions agree perfectly.

$$\text{Rand Index} = \frac{a + d}{a + b + c + d} \quad (7)$$

where:

- $a$  is the number of pairs of objects in the same class in both partitions.
- $b$  is the number of pairs of objects in the same class only in the external criterion.
- $c$  is the number of pairs of objects in the same class only in the clustering result.
- $d$  is the number of pairs of objects in different classes in both partitions.

The second performance criteria used by Ji *et al.* (2003) was the ratio figure of merit (FOM) from Yeung *et al.* (2001) which uses only the information within the dataset to assess the clustering results. It uses the idea that a good clustering technique would produce clusters that contain similar expression profiles within the cluster but be very different between clusters. The FOM is basically the ratio of within cluster differences to between cluster differences, see equation (8).

$$FOM_{ratio}(e, k) = \frac{\frac{1}{n} \sum_{i=1}^k \sum_{x \in C_i} |R(x, e) - \mu_{C_i}(e)|}{\frac{1}{k-1} (\mu_{C_i}^{max}(e) - \mu_{C_i}^{min}(e))} \quad (8)$$

where:

- $n$  is the number of genes
- $k$  is the number of clusters
- $R(x, e)$  is the expression level of gene  $x$  under condition  $e$
- $\mu_{C_i}(e)$  is the average expression level in condition  $e$  of genes in cluster  $C_i$

Therefore the best clustering techniques have a high between cluster difference, but low within cluster difference and producing a small ratio FOM value. In time course data the experimental conditions  $e$  correspond to the different time points. An estimate of the performance overall conditions for a given number of clusters (Yeung *et al.*, 2001) can be found with equation (9).

$$FOM_{ratio}(k) = \sum_{e=1}^m FOM_{ratio}(e, k) \quad (9)$$

When comparing the results of the different clustering methods Ji *et al.* (2003) found that the Rand and FOM ratio levels for different clusters were similar in all three methods, suggesting that HMMs are as good as k-means and SOM at clustering. Looking at the Rand indexes it can be seen that the HMM model suggested the correct number of clusters to when compared with the gold standard clustering. Furthermore the FOM index also correctly predicted the correct number clusters when compared with the gold standard. Therefore it could be possible to use this technique not only to allocate genes to clusters but also to predict the correct number of clusters for a dataset only using the information within the dataset. However the other k-means and SOM could not do this and so the HMM method of Ji *et al.* (2003) is in this way superior.

## 1.5 Aims of this research

The aim of this research was to build on the work of Ji *et al.* (2003) to use their Markov model clustering technique and to investigate if improvements could be made. The advantages of this type of method is that the optimal number of clusters can be found automatically, the model should be more robust to noise, genes can be allocated to more than one cluster and a measure of the probability that the clustering fits the data can be calculated

Our research used the same basic Markov chain model but instead integrated many chains together to form a larger hidden Markov model. With this new topology we no longer needed to use a weighted Baum-Welch strategy and used an alternative method to FOM for finding the optimal number of clusters. We used large, entire datasets rather than subsets of genes to see if this affected the quality of clustering. Therefore if the strategy was successful it should provide a fully automated clustering technique for time series data that could be used with entire genes sets to imply gene function without any human input. Finally we developed a strategy for adding transcriptions factor information into the model to allow regulatory information to be implied from more easily from the clustering.

## 2 APPROACH

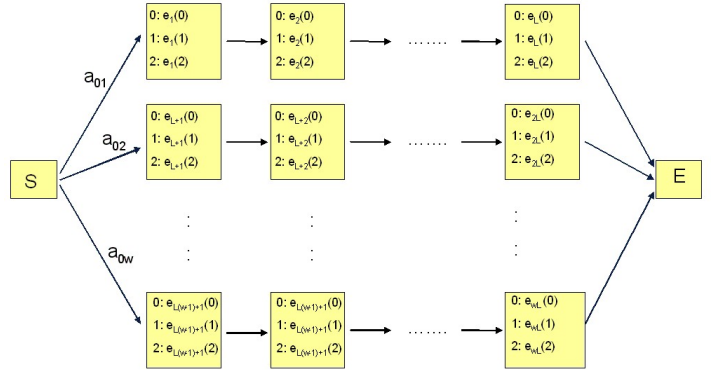
### 2.1 Datasets

A dataset from the Cookson laboratory at the Wellcome trust which had not previously been analysed was chosen. It contained affymetrix data of 22283 genes at 5 time points recording keratinocyte differentiation at 0,1,2,4 and 7 days. Each time point had been repeated experimentally 3 times and these repeats were averaged. The purpose of this dataset was to look at the gene expression during different stages of differentiation and it was expected that some genes would show large changes in expression levels. Eventually this dataset will be used in conjunction with other datasets produced from the Cookson laboratory to investigate asthma. This dataset was much bigger than the datasets analysed by Ji *et al.* (2003) using hidden Markov models as we wanted to investigate how the model performs using entire gene sets rather than a small predetermined subset that alternative clustering methods use.

It was hoped that a second more well known dataset would be analysed as well, so the results could be compared to previous clustering results. Unfortunately due to time constraints this was not possible.

### 2.2 Normalisation of datasets

The data was normalised using standard pre-processing methods for affymetrix data described in Bolstad (2004). First a *mas* background correction was made, then normalised using the quantiles method which is further discussed in Bolstad *et al.* (2003). The log mean of the channel was then subtracted from the log signal value to obtain a normalised log ratio for the dataset.



**Fig. 4.** Larger HMM representing  $w$  clusters as  $w$  separate paths from the start state to the end state. All transition probabilities apart from the initial transitions are set to 1.0 or 0.0.

### 2.3 Quantisation

After normalisation the datasets were quantised in the same way that Ji *et al.* (2003) used. A threshold of 1.5 was chosen to quantise the keratinocyte dataset, as it would be expected that large changes in regulation would occur between the time points. Therefore each gene was represented by a fluctuation sequence of length 5, comprising of a zero followed by  $N-1$  numbers representing the up-regulation, down-regulation or no change between time points.

### 2.4 Design of the HMM

Instead of having one HMM to represent each cluster, a similar larger topology was proposed. A cluster was represented as a path through the HMM, so if the model represented  $w$  clusters, there were  $w$  distinct paths through the states in the model. The structure of the HMM is shown in figure 4.

In this HMM, transitions between states in a path have transition probabilities of 1.0 and the transition probabilities for moving between states on different paths is 0.0. Initial transitions from the start state to a choice of  $w$  states represented the weight that path has in the model.

Therefore the number of states in the model is proportional to the number of clusters multiplied by the sequence length. Each state could emit one of three symbols, 0,1 or 2. Adding an extra path to the model increases the number of states by the sequence length, and number of emission probabilities by 3 times the sequence length. Modelling a dataset with more time points would increase the number of states by  $w$  and emissions by  $3w$  for each extra time point added. Adding more genes to the dataset would not increase the size of the clustering but would increase the time required for the trainer to find the maximum likelihood parameters for the model.

## 2.5 Programming the HMM Trainer

A Baum-Welch Trainer was programmed in order to train the parameters of the HMM. The trainer followed the algorithm described in Durbin *et al.* (1998) but with a random initiation of only the unknown parameters (see algorithm 2). Therefore using our HMM topology only the initial transitions and emission probabilities were recalculated.

### Baum-Welch

#### Initialise:

Set the values for the threshold and maximum iterations  
Randomly initialise the unknown parameters

#### Iterate:

Set the  $A_{kl}$  and  $E_k(b)$  variables to zero.  
For each sequence  $j=1\dots n$ ,  
Calculate  $F_k(i)$  and  $B_k(i)$  using algorithm 1  
Add the sequence contribution to  $A_{kl}$  and  $E_k(b)$  using equations (1) and (2)  
Calculate the new parameters using equations (3) and (4)  
Calculate the new log-likelihood for the model equation (5)  
Calculate the change in log-likelihood

#### Stop:

When the change in log likelihood is less than a threshold value or when the maximum number of iterations is reached

### Algorithm 2: Baum-Welch Training

As the Baum-Welch algorithms can get trapped in a local maxima, it was necessary to repeat the algorithm several times to find the global maximum giving the best estimate of the parameters. The global maximum was taken as the maximum value of all the local maximum calculated in the repeats. This repetition made the method very slow. Therefore the algorithm was optimised for this particular HMM topology rather than using a method general to all HMM topologies. The first optimisation was that only the unknown parameters were recalculated. Secondly the equations for  $F_k(i)$  and  $B_k(i)$  were reduced as each state other than the start state has only one state transition from it. Thus the forward backward equations (see algorithm 1) were reduced to (10) and (11), where  $a_{Sv}$  is the transition from the start state to the start of the path representing the cluster  $v$  which contains state  $k$ .

$$F_k(i) = a_{Sv}e_{k-i}(x_1)e_{k+1-i}(x_2)\dots e_k(x_i) \quad (10)$$

$$B_k(i) = e_k(x_i)e_{k+1}(x_{i+1})\dots e_{k+L-i}(x_L) \quad (11)$$

Furthermore for all states  $k$  in a particular path (or cluster)  $v$ , the multiplication of forward and backward probabilities were the same and could be denoted as  $FB_v$ . The proof is given below in proof 1.0,  $i'$  was the position of symbol in the sequence that can be emitted from state  $k$ .

PROOF 1.0 .

$$\begin{aligned} & \sum_i F_k(i)a_{kl}e_l(x_{i+1})B_l(i+1) \\ &= F_k(i')a_{kl}e_l(x_{i'+1})B_l(i'+1) \\ &= F_k(i')B_k(i') \\ &= a_{Sv}e_{(v-1)L+1}(x_1)e_{(v-1)L+2}(x_2)\dots e_{vL}(x_L) \text{ from (10 \& 11)} \\ &= FB_v \end{aligned}$$

Therefore it was only necessary to calculate and store the value of  $FB_v$  for  $v=1..w$ , in order to re-estimate the parameters rather than the  $F_k(i)$  and  $B_k(i)$  for  $i=1..L$  and  $k=1..wL$ . The equations for  $A_{kl}$  and  $E_k(b)$  reduced to:

$$A_{kl} = \sum_j \frac{1}{P(x^j)} FB_v \quad (12)$$

$$\begin{aligned} E_k(b) &= \sum_j \frac{1}{P(x^j)} \sum_{i|x_i^j=b} FB_v \\ &= \begin{cases} FB_v & \text{if } x_i^j = b \text{ and state } k \text{ emits the } i\text{th symbol} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (13)$$

Using these optimised equations the program was much quicker. However as the number of clusters was not known, the Baum-Welch had to be repeated using different number of clusters (different sized HMMs), the larger HMMs took longer to run as they had more parameters and took longer to converge.

## 2.6 Calculating the Optimal Number of Clusters

The HMMs for different numbers of clusters are hierarchically nested models. This means by adding a cluster we have the same model with more parameters added. As the models use this structure the likelihood ratio test static (LRT) can be used to calculate the optimal number of clusters for a model. Adding additional parameters to a model increases the likelihood that the model fits the dataset but after a certain point adding extra parameters will not significantly increase the likelihood. The LRT allows this critical point to be determined.

First the likelihood ratio,  $LR$ , is calculated, where  $LR = 2(LnL1 - LnL2)$ ,  $LnL1$  is the log-likelihood of the more complex model,  $LnL2$  is the log likelihood of the simpler model. This LRT approximately follows a chi-squared distribution, so a critical value can be calculated from statistical tables of the chi-squared distribution, using the number of new parameters and a chosen  $p$  value. For this HMM topology adding a cluster would result in  $L(Z - 1) + 1$  new

---

parameters, where  $L$  is the length of the sequence and  $Z$  is the number of emission symbols. As the emission probabilities sum to one for each state, knowing  $Z-1$  probabilities would allow the  $Z$ th probability to be known as well, thus the number of extra emission probability parameters was  $L(Z-1)$ . There was also one extra transition probability added from the start state to the new cluster. Summing the number of new emission parameters and transition parameters therefore gives an increase as  $L(Z-1) + 1$ .

If CR is the value of the chi-squared distribution with the number of extra parameters at a certain p-value, it can be compared with the LR value. If LR is greater than the CR then it is worthwhile adding new parameters, if it is smaller then the increase in log-likelihood is not sufficiently significant to warrant an increase in complexity.

Using this statistic it is possible to automatically calculate whether increasing the complexity is worthwhile. So the algorithm could start at a given minimum cluster size and increase the number of clusters until the LRT calculates the complex models no longer need to be considered. This is significant because current methods such as k-means clustering do not have an automated way of determining the number of clusters but rely on subjective assessment.

## 2.7 Adding Transcription Factor Data

To reduce the effects of noise it would be useful to incorporate transcription factor information into the model so that clusters were slightly biased towards genes with similar regulatory patterns. Transcription factor could be incorporated into the model using additional states. There are two possible strategies for this which are illustrated in figure 5:

1. Model each transcription factor as a state with emission symbols 0 to represent no association with the gene and 1 to represent an association with the gene.
2. Add a certain number of states to each path through the model that emit symbols representing the transcription factors. These states would need to have tied probabilities such the emission probabilities for each TF state would be the same, such that the ordering of the transcription factors were not important.

A transcription factor sequence would be appended to the beginning of each of the fluctuation sequence.

In the first strategy this would be a series of 0's or 1's with one symbol for each of the transcription factors being modelled. This would dramatically increase the length of the emission sequences, and if the model is supposed to predict on emission sequences as well as fluctuation sequences care must be taken to ensure the HMM does not have too much bias towards TF sequence in clustering genes. As the majority of TF values would be 0 it was thought that the bias towards

transcription factor sequences should not be too high and therefore this strategy would be reasonable. In order to assess the influence of the transcription factor information the model could be run without the transcription factor information and the results compared.

In the second strategy a maximum number of transcription factors is chosen to associate with each gene then add them as symbols to the beginning of the sequence. The transcription factors chosen for each gene would be the ones calculated as most significant for each gene. If fewer transcription factors were associated with the gene then a null symbol would be required to represent no TF. So that the ordering of the transcription factors do not affect the emission probabilities the re-estimation for the emission parameters should be summed from all of the states on each path and then set to the same value for each state. Effectively this means that the system remains in the same state for a set number of time points. However this can not be modelled directly as a state with a transition to itself would allow the state to emit any number between 1 and an infinite number of symbols rather than the set number as required.

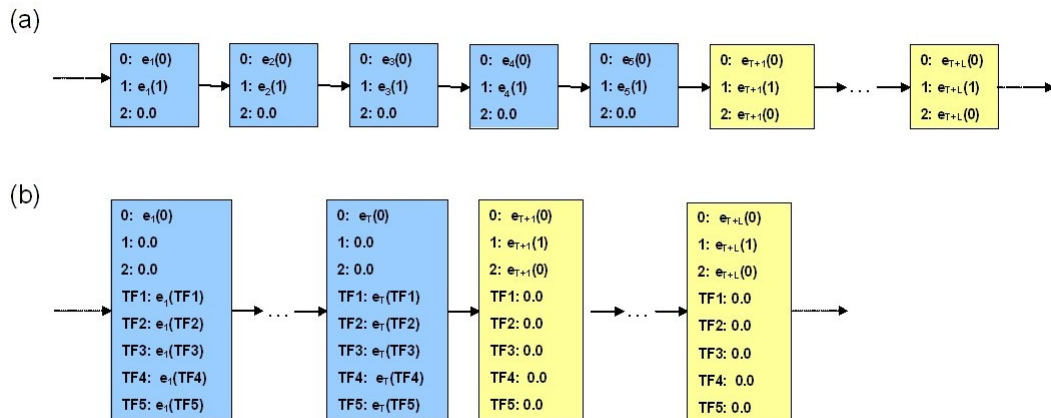
Transcription factor information could be found by taking sequences 1Kb upstream of the transcription start site from Ensembl using EnSmart. MSCAN (Wynand *et al.*, 2003) can then be used to find binding sites in the underlying sequences. For the keratinocyte dataset 140 transcription factors were found that satisfied the criteria of Rahmann *et al.* (2003) with a 0.9 specificity and 0.9 sensitivity.

## 3 RESULTS AND ANALYSIS

The keratinocyte dataset was clustered using the HMM model representing 2 to 25 clusters. Any fluctuation sequences that consisted of only zeros were removed before training which left 20040 genes.

### 3.1 Allocating Gene Profiles to Clusters

A simple approach was used which calculated the probability of each sequence being produced from each cluster, then allocating the sequence to the cluster that gave the highest probability. However this simple approach could be easily extended such that a sequence could be allocated to more than one cluster, which could give more biologically meaningful results. This could be done by comparing the probabilities for all clusters and if are two or more clusters give a similar probability to the maximum probability allow all these clusters to contain the sequence. This would allow genes with multiple functions to be modelled.



**Fig. 5.** (a) A state is added for each transcription factor to the beginning of each Markov path, which can emit a 0 or 1. In the figure only five transcription factors are used, but this number would be much larger in practice. (b) A set number of states are added to represent the most significant transcription factors, emission symbols represent the transcription factors. Again only a small example of five transcription factors has been illustrated. The emission probabilities for the transcription factor states on the same path should be the same for each symbol. The symbol 0 can be used to represent no transcription factor.

### 3.2 Parameters Used

The following parameters were used to produce the clustering results for a dataset quantised for 1.5 fold regulation changes.

#### *Baum-Welch parameters*

Number of iterations = 200

Log change threshold = 0.0001

Number of repetitions = 100

Number of sequences,  $n = 20040$

#### *HMM parameters*

Number of cluster paths,  $w = 2$  to 25

Length of sequences,  $L = 5$

Number of emissions per state,  $Z = 3$

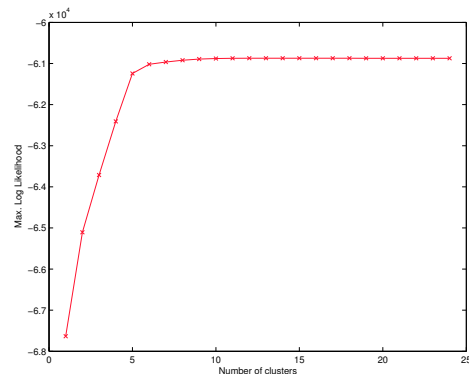
Emission symbol alphabet =  $\{1,2,3\}$

Number of hidden states =  $wL$

### 3.3 Optimal Number of Clusters

The optimal number of clusters found from the LRT statistic was 11. This was using an added 11 degrees of freedom at each time step, with a p value of 0.05, giving a critical value,  $CR$ , of 19.675.

Figure 6 shows how the log likelihood of the model increased with the number of clusters, showing an expected steep increase as the cluster size increased which slowed down when adding extra parameters was no longer worthwhile. Table 1 shows the LRT calculations which gave the optimal number of clusters as 11. A positive  $LR - CR$  value means that



**Fig. 6.** Log likelihoods for different numbers of clusters. The log likelihood values increase with the number of clusters sizes. After 11 clusters there is not a significant increase the the log likelihood.

the increase in degrees of freedom is worthwhile, whereas a negative value suggests the increase in parameters does not give a better model. It was expected that the log likelihood should always increase as the number of clusters increased, however the results show a very slight decrease as the number of clusters increase beyond 15. This may be because the algorithm was not given a sufficient number of iterations to reach the maximum value in the more complex models.

The results set was run with numbers of cluster varying from 2 to 25 in order to investigate how the log likelihood and clustering results varied. However if only the optimal clustering

**Table 1.** Table showing LRT calculations. The LR-CR values becomes negative after 11 clusters giving the optimal number as 11

Num Clusters	Log Likelihood	LR	LR-CR
2	-67634.2		
3	-65104.6	5059.321	5039.646
4	-63711.8	2785.547	2765.872
5	-62409.0	2605.597	2585.922
6	-61242.8	2315.132	2312.732
7	-61019.2	47.1285	427.4535
8	-60964.7	109.1591	89.48406
9	-60920.0	89.40004	69.72504
10	-60892.6	54.63802	34.96302
11	-60878.8	27.71818	8.043184
12	-60874.4	8.834064	-10.8409
13	-60872.3	4.040909	-15.6341
14	-60871.3	2.003527	-17.6715
15	-60871.2	0.374642	-19.3004
16	-60872.2	-2.05058	-21.7256
17	-60872.4	-0.34718	-20.0222
18	-60872.4	-0.0779	-19.7529
19	-60872.5	-0.17225	-19.8472
20	-60872.5	-0.07449	-19.7495
21	-60872.5	-0.05543	-19.7304
22	-60872.6	-0.03297	-19.708
23	-60872.7	-0.22612	-19.9011
24	-60872.7	0.015438	-19.6596
25	-60872.8	-0.18055	-19.8556

result is required then the algorithm would have only needed to calculate up to 12 clusters and could then automatically stop using the LRT statistic.

### 3.4 Comparison with K-means Clustering

The dataset was also clustered using k-means clustering. However k-means clustering could not use the entire gene set and so a subset of 1126 genes was used. Looking at the quality of k-means clustering by eye it was decided that a set of 15 clusters gave the best result. The k-means clustering on the subset of genes was then compared using the Rand index (equation 7) to the optimal HMM clustering result to see what proportion of gene pairs had been clustered similarly by both methods. The Rand index was calculated to be 0.702 which showed that there was a reasonable but not perfect correspondence between the two clustering methods. The reasons for this are probably that the HMM was clustered using an extra 18914 genes which could bias some profiles towards different clusters.

### 3.5 Profiles in Each Cluster

The normalised expression data for the genes in each cluster was then plotted to visualise which profiles had been clustered together (see Figures 7 and 8). The cluster sizes ranged from 541 to 4090, each corresponding to a set of fluctuation sequences.

Cluster 5 showed a very clear expression pattern for all the 541 genes, which was up-regulation, down-regulation and up-regulation. This cluster therefore shows the genes which experienced a greater than 1.5 fold regulation change of up, down, up for the first 3 changes. Cluster 7 had similar results showing a down, up, down regulation pattern.

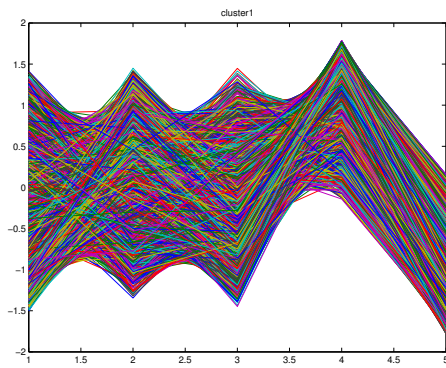
Patterns can be seen in the other clusters, however cluster 3 for example shows a definite up regulation between time point 1 and 2 for but the pattern after time point 2 is not as clear. Looking at the fluctuation sequences that this cluster contains, the majority are 01000 and so most of the profiles have a less than 1.5 fold change for between the final three time points. The HMM has correctly clustered on the fluctuation sequences but regulation changes of less than 1.5 fold up or down for thousands of genes can produce a wide spread of intensity values. If it was certain that less than a 1.5 fold change was biologically meaningless for the dataset then the clustering could be useful. However if finer changes need to be found then a smaller threshold could be used. Alternatively a better way of quantising the data could be used, as quantising to only 3 possible values loses a great deal of information that was contained in the intensity values. There was not time to implement better methods but some possible improvements are given in the discussion section of this report.

### 3.6 Changing the Quantisation Threshold

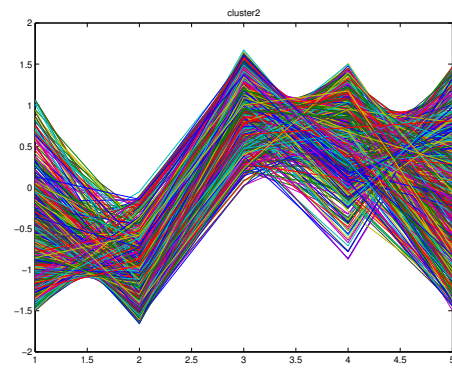
To evaluate if clearer patterns could be found the clustering was run again using a quantisation threshold of 1.0, the hope was to see smaller spreads of intensity values when the fluctuation change was 0. This meant that the fluctuation sequences would contain fewer 0s.

The clustering used the same parameters as before, except the maximum iterations of the Baum Welch trainer was increased to 300 to allow it to get closer to the maximum likelihood value for larger numbers of clusters. As the fluctuation sequences changed, fewer were all zeros, so the training was run with 22099 sequences.

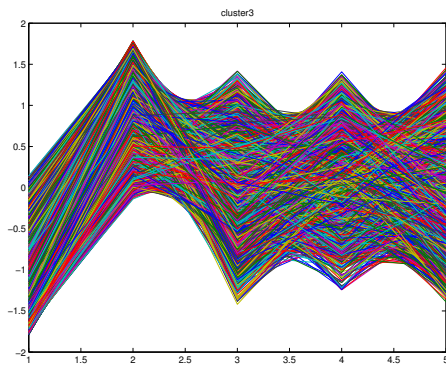
The LRT calculations give the optimal number of clusters to be 12. This clustering result was compared with the k-means clustering giving a higher Rand index than the previous results, of 0.780. Therefore clustering using a threshold fold change of 1.0 gives a closer result to the k-means than a threshold of 1.5.



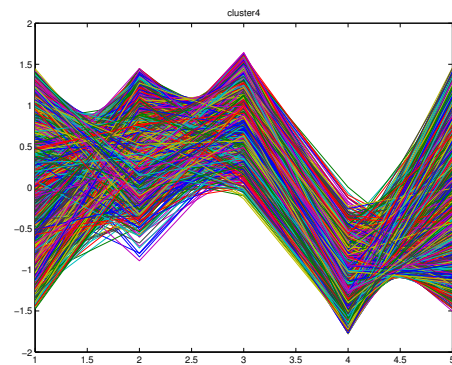
**Cluster 1, 2946 genes.** Represents sequences that have a greater than 1.5 down regulation between time points 4 and 5



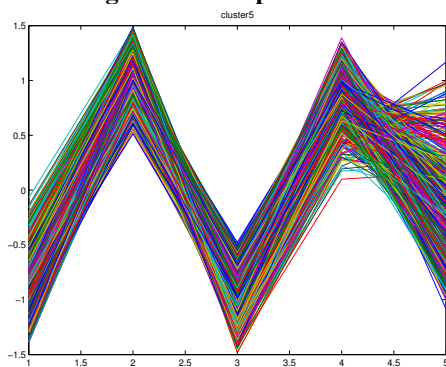
**Cluster 2, 666 genes.** Represents sequences with an up-regulation between time points 2 and 3.



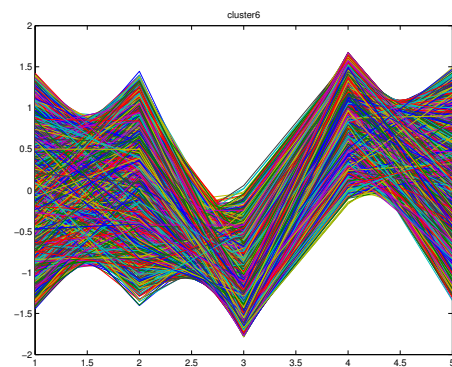
**Cluster 3, 3757 genes.** The majority of genes are up-regulated by more than 1.5 fold between time points 1 and 2. Allow there is a wide spread of intensity after time 2 the individual gene regulation changes by less than 1.5 fold between time points. Therefore the cluster shows genes that do not have a significant regulation change after time point 2.



**Cluster 4, 1365 genes.** This shows changes that do not have a significant regulation change apart from an up regulation between time points 3 and 4.

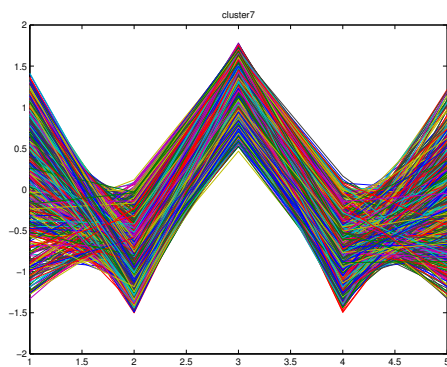


**Cluster 5, 541 genes.** This relatively small cluster showing a clear pattern of up, down, up regulation.

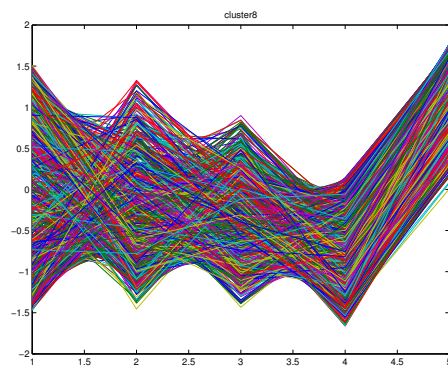


**Cluster 6, 2710 genes.**

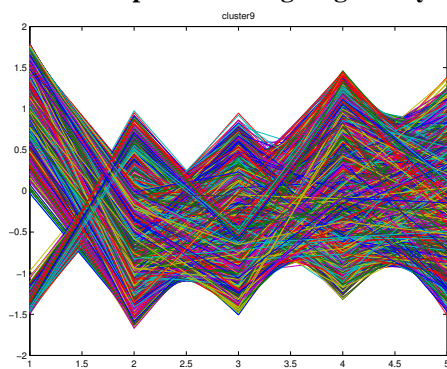
**Fig. 7.** Clusters 1 to 6.



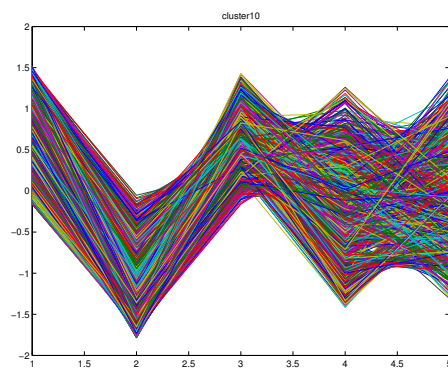
**Cluster 7, 843 genes. Another relatively small cluster that shows a clear pattern of large regulatory changes.**



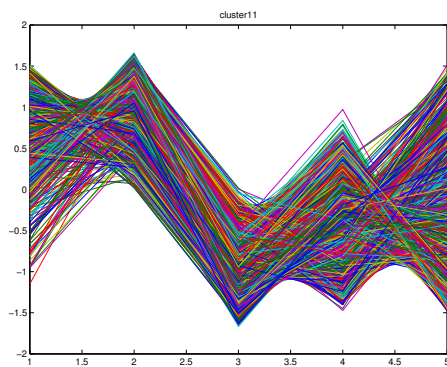
**Cluster 8, 936 genes.**



**Cluster 9, 4090 genes**



**Cluster 10, 1225 genes.**



**Cluster 11, 961 genes**

**Fig. 8.** Clusters 7 to 11.

The gene expression intensities were then plotted for the genes in each cluster. However expression patterns for the clusters seemed to be more clearly defined in the original results with a threshold of 1.5 than with a threshold of 1.0.

### 3.7 Biological Meaning of Clusters

The clusters highlight changes in expression between time points. Analysing the purpose of these gene clusters suggests that cell proliferation and transcription factor activation is high in the first stages. Between 1 and 4 days the protein bio-synthesis increases suggesting the cells are starting

to specialise. By the 7th day the transcription factor activation and cell proliferation has slowed down and instead the genes associated with cell growth, maintenance, signalling and transport are up regulated, suggesting that the cells are specialised by this stage.

## 4 DISCUSSION AND FUTURE WORK

We presented some improvements to a hidden Markov model clustering method. These included changing the Markov model topology such the calculations of the weights each sequence had in the re-estimation of a model was no longer necessary, this was instead effectively modelled using initial transition probabilities to a set of simpler models. A computationally easier method for calculating the optimal number of clusters was suggested and we managed to run our model on a larger dataset. In order to verify that this method could produce biologically meaningful results it would be necessary to run the method on further datasets that vary in the number of genes and time points. Due to the time constraints of the project and the slow speed of the algorithm this has not been possible so far.

A potential problem of the model is that it tends to fit expression patterns that are the most frequent, so effectively the most common sequences are given the highest probabilities under the model. If a dataset contained a relatively small number of genes that had a biologically interesting expression pattern these could be blurred behind a large number of uninteresting expression profiles. However this may not be such a big problem if the dataset has a larger number of time points, where there is likely to be more variation in the fluctuation sequences but improvements could be made to the quantisation method in order to improve this.

Another criticism of the technique is that it is very slow to run due to the number of training repetitions that need to be done. The likelihood ratio test should allow the algorithm to stop once the optimal number of clusters has been found which speeds up the clustering particularly for datasets with only a small number of clusters. Adding more states, more emission symbols or more sequences all slow the algorithm down considerably. However as all the repetitions of the training are done independently and then the maximum likelihood taken it would be very easy to distribute the computation between multiple processors in order to find the solution faster.

There are many more improvements to the method that there was unfortunately no time to investigate. Rather than lose a huge amount of information during the quantisation step it would be better to use the actual intensity values with a continuous probability distribution for the emission probabilities

rather than allowing only a few discrete emission probabilities. If it is more interesting to look at the changes in intensity between time points then the difference in intensity values could be used instead. It would be interesting to incorporate the transcription factor information into the model to see if any clustering improvements could be made. Different methods for allocating genes to clusters could be investigated to allow genes to be allocated to more than one cluster if it fits both clusters well. It may also be useful to try to define a different topology with fewer states such that the speed of the algorithm would not decrease so dramatically as the number of clusters and time points being modelled increases.

## ACKNOWLEDGEMENTS

## REFERENCES

- Bar-Joseph,Z., Gerber,G., Gifford,D.K., Jaakkola,T.S. (2002) A new approach to analyzing gene expression time series data. *6th Annual International Conference on Research in Computational Molecular Biology*.
- Bolstad,B. (2004) affy:Built-in Processing Methods, <http://www.bioconductor.org/repository/devel/vignette/builtinMethods.pdf>.
- Bolstad,B.M., Irizarry,R.A., Åstrand,M., Speed,T.P. (2003) A comparison of normalization methods for high density oligonucleotide array data based on variance and bias, *Bioinformatics*, **vol.19(2)**, 185-193.
- Durbin,R., Eddy,S., Krogh,A., Mitchison,G. (1998) Biological Sequence Analysis: Probabilistic models of proteins and nucleic acids, *Cambridge University Press*.
- Filkov,V., Skiena,S., and Zhi,J. (2002) Analysis techniques for microarray time series data, *J.Comput.Biol.*, **9**, 317-330.
- Ji,X., Li-Ling,J., Sun,Z. (2003) Mining gene expression data using a novel approach based on hidden Markov models, *FEBS Lett.*, **2003 May 8; 542(1-3)**, 125-31.
- Holmes,C.(2004) A quantitative study of gene regulation involved in the immune response of anopheline mosquitoes: an application of Bayesian hierarchical clustering of curves (*Submitted*).
- Rabiner,L.R.(1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc.IEEE*, **vol.77(2)**, 257-285.
- Rahmann,S., Mueller,T., Vingron,M. (2003) On the power of profiles for transcription factor binding site detection, *Stat.Appl.Genet.Mol.Biol.*, **2,7**.
- Ramoni,M.F., Sebastiani,P., and Cohen,P.R. (2002a) Bayesian clustering by dynamics, *Mach. Learn.*, **47**, 91-121.
- Schliep,A., Schönhuth,A., Steinhoff,C. (2003) Using hidden Markov models to analyse gene expression time course data, *Bioinformatics*, **vol.19(1)**, i255-i263.
- Wynand,B.L., Öjvind,J., Lagergren,J.,Wasserman,W.W. (2004) MSCAN: identification of functional clusters of transcription factor binding sites, *Nucleic Acids Research*, **vol.32**, W195-W198.
- Yeung,K.Y., Haynor,D.R.,Ruzzo,W.L. (2001) Validating clustering for gene expression data, *Bioinformatics*, **vol.17(4)**, 309-318.