Gene expression

Robust multi-scale clustering of large DNA microarray datasets with the consensus algorithm

Thomas Grotkjær¹,*, Ole Winther², Birgitte Regenberg¹, Jens Nielsen¹
and Lars Kai Hansen²

¹Center for Microbial Biotechnology, BioCentrum-DTU, Building 223 and ²Informatics and Mathematical Modelling, Building 321, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

Received on February 11, 2005; revised on October 13, 2005; accepted on October 25, 2005
Advance Access publication October 27 2005
Associate Editor: Joaquin Dopazo

ABSTRACT
Motivation: Hierarchical and relocation clustering (e.g. K-means and self-organizing maps) have been successful tools in the display and analysis of whole genome DNA microarray expression data. However, the results of hierarchical clustering are sensitive to outliers, and most relocation methods give results which are dependent on the initialization of the algorithm. Therefore, it is difficult to assess the significance of the results. We have developed a consensus clustering algorithm, where the final result is averaged over multiple clustering runs, giving a robust and reproducible clustering, capable of capturing small signal variations. The algorithm preserves valuable properties of hierarchical clustering, which is useful for visualization and interpretation of the results. 
Results: We show for the first time that one can take advantage of multiple clustering runs in DNA microarray analysis by collecting re-occurring clustering patterns in a co-occurrence matrix. The results show that consensus clustering obtained from clustering multiple times with Variational Bayes Mixtures of Gaussians or K-means significantly reduces the classification error rate for a simulated dataset. The method is flexible and it is possible to find consensus clusters from different clustering algorithms. Thus, the algorithm can be used as a framework to test in a quantitative manner the homogeneity of different clustering algorithms. We compare the method with a number of state-of-the-art clustering methods. It is shown that the method is robust and gives low classification error rates for a realistic, simulated dataset. The algorithm is also demonstrated for real datasets. It is shown that more biological meaningful transcriptional patterns can be found without conservative statistical or fold-change exclusion of data.
Availability: Matlab source code for the clustering algorithm ClusterLustre, and the simulated dataset for testing are available upon request from T.G. and O.W.
Contact: tg@biocentrum.dtu.dk and owi@imm.dtu.dk
Supplementary information: http://www.cmb.dtu.dk/

1 INTRODUCTION

The analysis of whole genome transcription data using clustering has been a very useful tool to display (Eisen et al., 1998) and to identify the functionality of genes (DeRisi et al., 1997; Gasch et al., 2000). However, it is well known that many relocation clustering algorithms such as K-means (Eisen et al., 1998), self-organizing maps (SOM) (Tarnayo et al., 1999), Mixtures of Gaussians (Mackay, 2003), etc. give results that depend upon the initialization of the clustering algorithm. This tendency is even more pronounced when the dataset increases in size and transcripts with more noisy profiles are included in the dataset. It is therefore common to make a substantial data reduction before applying clustering. This is acceptable when we expect only few genes to be affected in the experiment, but if thousands of genes are affected the data reduction will remove many informative genes. In a recent study it was also clearly demonstrated that small changes in the expression level were biological meaningful, when the yeast Saccharomyces cerevisiae was grown under well-controlled conditions (Jones et al., 2003).

Hence, with the emerging quantitative and integrative approaches to study biology there is a need to cluster larger transcription datasets, reduce the randomness of the clustering result and assess the statistical significance of the results (Grotkjær and Nielsen, 2004).

An alternative to relocation clustering is hierarchical clustering where transcripts are assembled into a dendrogram, but here the structure of the dendrogram is sensitive to outliers (Hastie et al., 2001). A practical approach to DNA microarray analysis is to run different clustering methods with different data reduction (filtering) schemes and manually look for reproducible patterns (Kaminski and Friedman, 2002). This strategy is reasonable because the clustering objective is really ill defined, i.e. the natural definition of distance or metric for the data is not known. Clustering methods vary in objective and metric, but the success of the practical approach shows that many objectives share traits that often make more biological sense than looking at the results of any methods alone.

A Bayesian model selection should be able to find the relative probability of the different clustering methods tested (MacKay, 2003). The main problem with the Bayesian approach is computational since for non-trivial models, it is always computationally intractable to perform the necessary averages over model parameters. Approximations such as Monte Carlo sampling (MacKay, 2003; Dubey et al., 2004) or variational Bayes (Attias, 2000) have to be employed instead. A proper model parameter average will give a clustering that is unique up to an arbitrary permutation of labels, i.e. the cluster numbering is allowed to change. Unfortunately approximate methods tend to give results that are non-unique. The randomness of an algorithm, approximate Bayesian or any other, can be interpreted as arising from partitionings of the data that are more or less equally likely, and the algorithm is stuck in a local
maxima of the objective. This is a practical problem, and global search methods such as Monte Carlo or genetic algorithms (Falkenauer and Marchand, 2003) have been devised to overcome this. However, we can also choose to take advantage of the randomness in the solutions to devise a robust consensus clustering which is the strategy taken here. Upon averaging over multiple runs with different algorithms and settings, common patterns will be amplified whereas non-reproducible features of the individual runs are suppressed.

The outline of the paper is as follows: first, we present the consensus clustering algorithm framework. The actual clustering method used variational Bayes (VB) Mixture of Gaussians (MoG) (Attias, 2000) as described in Supplementary material since VBMoG and its maximum likelihood counterpart are already well-established in the DNA microarray literature (McLachlan et al., 2002; Ghosh and Chinnaiyan, 2002; Pan et al., 2002). The developed framework is tested on a generative model for DNA microarray data, since it is crucial with a simulated dataset that reflects the underlying biological signal. Finally, we show how one can use the consensus clustering algorithm to group co-expressed genes in large real whole genome datasets. The results demonstrate that cluster-then-analyse is a good alternative to the commonly used filter-then-cluster approach.

2 CONSENSUS CLUSTERING

The consensus clustering method described in this paper is related to those more or less independently and recently proposed in Fred and Jain (2002, 2003); Strehl and Ghosh (2002); Monti et al. (2003), see also the discussion of related work in Section 5 of Strehl and Ghosh (2002). The main features of these methods are that they use only the cluster assignments (soft or hard) as input to form the consensus (and not e.g. cluster means), and the consensus clustering is able to identify clusters with more complex shapes than the input clusters. For instance, K-means is forming spherical clusters but non-spherical clusters can be identified with the consensus clustering algorithm if K-means is used as input (Fred and Jain, 2003). Here, we will motivate the introduction of the consensus method in DNA microarray analysis from a model averaging point of view as a way to make approximate Bayesian averaging when the marginal likelihoods coming out of the approximate Bayesian machinery cannot be trusted.

2.1 Soft and hard assignment clustering

In this section we briefly introduce the basic concepts of probabilistic clustering, for more details, see Supplementary material. The probabilistic (or soft) assignment is a vector \( p(x) = [p(1 | x_1), \ldots, p(K | x_N)]^T \) giving the probabilities of the \( k = 1, \ldots, K \) cluster labels for an object \( M \) experiment data vectors \( x = [x_1, \ldots, x_N]^T \). One way to model \( p(k | x) \) is through a mixture model

\[
p(k | x) = \frac{p(k)p(x | k)}{\sum_{k'=1}^{K} p(k')p(x | k')},
\]

Hard assignments are the degenerate case of the soft assignment where one component, say, \( pk(x) \) is one, but a hard assignment can also be obtained from \( a(x) = \text{argmax}_{k} p(k | x) \), i.e. a transcript is only assigned to the most probable cluster.

In practice we do not know the density model before the data arrive and we must learn it from the dataset: \( D_N \equiv \{x_1, \ldots, x_N\} \) of size \( N \) examples (number of transcripts). We therefore write the mixture model as an explicit function of the set of model parameters \( \theta \) and the model \( M \):

\[
p(x | \theta, M) = \sum_{k=1}^{K} p(k | \theta, M)p(x | k, \theta, M).
\]

The model \( M \) is shorthand for the clustering method used and the setting of parameters such as the number of clusters \( K \). Maximum likelihood and the Bayesian approach give two fundamentally different ways of dealing with the uncertainty of the model parameters \( \theta \).

In maximum likelihood the parameters are found by maximizing the likelihood of the parameters: \( \theta^{\text{ML}} = \text{argmax}_{\theta} p(D_N | \theta, M) \) with the assumption of independent examples

\[
p(D_N | \theta, M) = \prod_{n=1}^{N} p(x_n | \theta, M)
\]

and assignment probabilities are given by

\[
p(k | x, \theta^{\text{ML}}, M) \propto p(k | \theta^{\text{ML}}, M)p(x | k, \theta^{\text{ML}}, M).
\]

This naturally leads to a set of iterative expectation maximization (EM) updates which are guaranteed to converge to a local maximum of the likelihood. In the Bayesian approach we form the posterior distribution of the parameters \( p(\theta | D_N, M) = [p(D_N | \theta, M)p(\theta | M)] / p(D_N | M) \), where \( p(\theta | M) \) is the prior over model parameters and

\[
p(D_N | M) = \int d\theta p(D_N | \theta, M)p(\theta | M)
\]

is the likelihood of the model (marginal likelihood or evidence). We can find the cluster assignment probability for a data point \( x \) by averaging out the parameters using the posterior distribution

\[
p(k | x, D_N, M) = [p(k | x, D_N, M)p(D_N | M)] / p(D_N | M).
\]

Either way, we calculate the soft assignments and obtain an assignment matrix \( P = [p(x_1), \ldots, p(x_N)] \) of size \( K \times N \).

The marginal likelihood plays a special role because it can be used for model selection/averaging; i.e. we can assign a probability to each model \( M \propto p(M) / p(D_N | M) \), where \( p(M) \) is the prior probability of the model. For non-trivial models the evidence is computationally intractable although asymptotic expressions exist. The VB framework aims at approximating the average over the parameters, but it unfortunately underestimates the width of the posterior distribution of \( \theta \) (MacKay, 2003). As a consequence multiple modes of the approximate marginal likelihood exists for this flexible model. It means that depending upon the initialization, two runs \( r \) and \( r' \) give different estimates of the marginal likelihood, i.e.

\[
p_{\text{app}}(D_N | r, M) \neq p_{\text{app}}(D_N | r', M).
\]

This clearly indicates that the posterior averaging has not been performed correctly. However, the clustering we find in a run typically has many sensible features and can still be useful if we combine clusterings from multiple runs.

2.2 Averaging over the cluster ensemble

After partitioning the data \( R \) times we have a cluster ensemble of \( R \) soft assignment matrices \( [P_1, \ldots, P_R] \). We may also have posterior
probabilities for each run \( p(M_t | D_N) \propto p(D_N | M_t)p(M_t) \), where \( M_t \) is the model used in the \( r \) run. From the cluster ensemble we can get different average quantities of interest.

We will concentrate on measures that are invariant with respect to the labelling of the clusters and can be used to extract knowledge from runs with different number of clusters. The co-occurrence matrix \( C_{nt} \) is the probability that transcript \( n \) and \( t \) are in the same cluster

\[
C_{nt} = \sum_{r=1}^{R} \sum_{k=1}^{K} p(k | x_n, r)p(k | x_t, r)p(M_t | D_N) = \sum_{r=1}^{R} \left[ P_{rr}^{t}P_{lr}^{nt}p(M_t | D_N) \right].
\]

We can convert the co-occurrence matrix into a transcript–transcript distance matrix \( D_{nt} = 1 - C_{nt} \). This distance matrix can be used as input to a standard hierarchical clustering algorithm. In the chosen Ward algorithm (Ward, 1963), clusters which ‘do not increase the variation drastically’ are merged when the number of leaves (clusters) is decreased, see Section 4.1.

### 2.3 Mutual information

The normalized mutual information can be used to quantify the significance of the different clustering runs, i.e. how diverse are the partitionings. Strehl and Ghosh (2002) proposed the mutual information between the cluster ensemble and the single consensus clustering as the learning objective, and Monti et al. (2003) used the same basic method [apparently without being aware of the work of Fred and Jain (2002)] focusing their analysis on the stability of clustering towards perturbations. The mutual information between two runs, \( r \) and \( r' \), measures the similarity between the clustering solutions

\[
M_{rst} = \sum_{k,k'} p_{rs}(k,k') \log \frac{p_{rs}(k,k')}{p_r(k)p_{r'}(k')},
\]

where the joint probability of label \( k \) and \( k' \) in runs \( r \) and \( r' \) is calculated as

\[
p_{rs}(k,k') = \frac{1}{N} \sum_{n} p(k | x_n, r)p(k' | x_n, r')
\]

and the marginal probabilities as

\[
p_r(k) = \frac{1}{N} \sum_{n} p(k | x_n, r) = \sum_{k'} p_{rs}(k, k').
\]

We can also introduce a normalized version of this quantity:

\[
M_{rst}^{\text{norm}} = \frac{M_{rst}}{\max(M_r, M_{r'})} \in [0;1],
\]

where the entropy of the marginal distributions \( p_r(k) \) is given by

\[
M_r = -\sum_{k} p_r(k) \log p_r(k).
\]

Finding the consensus clustering by optimizing the mutual information directly is NP-hard and the method suggested above may be viewed as an approximation to do this (Strehl and Ghosh, 2002). The average mutual information

\[
\overline{M}^{\text{norm}} = \frac{2}{R(R-1)} \sum_{r,r',r>r'} M_{rst}^{\text{norm}}
\]

can be used as a yardstick for determining the sufficient number of repetitions. Clearly when \( \overline{M}^{\text{norm}} \) is small, the cluster ensemble is diverse, and more repetitions are needed. We can express the required number of repetitions as a function of \( \overline{M}^{\text{norm}} \) by assuming a simplistic randomization process: the observed cluster assignment is a noisy version of the true (unknown) clustering. This randomization both lowers the mutual information and introduces ‘false positive’ entries in the co-occurrence matrix. Requiring that the ‘true positive’ entries should be significantly larger than ‘false positive’ determines \( R \) in terms of \( \overline{M}^{\text{norm}} \). See Supplementary material for more detail.

### 3 GENERATIVE MODEL

In order to test the performance of the consensus clustering algorithm we developed an artificial dataset based on a statistical model of transcription data. Rocke and Durbin (2001) showed that data from spotted cDNA microarrays could be fitted to a two-component generative model. The model was also shown to be valid for oligonucleotide microarrays manufactured by Affymetrix GeneChip (Geller et al., 2003; Rocke and Durbin, 2003). Here we consider a slight generalization of this model by including a multiplicative gene effect \( \exp(y_n) \) on the ‘true’ transcript level \( \mu_{nm} \) of gene \( n = 1, \ldots, N \) in DNA microarray \( m = 1, \ldots, M \). The introduction of this factor is reflecting the fact that the transcript level of individual genes have different magnitude. The measured transcript level, \( y_{nm} \), is given by

\[
y_{nm} = \alpha_n + \mu_{nm} \exp(y_{nm}) + \sigma_{nm},
\]

where \( \alpha_n \) is the mean background noise of DNA microarray \( m \) and \( \eta \sim N(0, \sigma_{in}^2) \) and \( \epsilon \sim N(0, \sigma_{e}^2) \) are biological- and technical-dependent multiplicative and additive errors that follow Gaussian distributions \( N \) with mean 0, and variances \( \sigma_{in}^2 \) and \( \sigma_{e}^2 \), respectively. The parameters \( \alpha_n \) and \( \sigma_{e} \) can be estimated by considering the probe sets with lowest intensity (Rocke and Durbin, 2001).

The rather strong influence of transcript dependent multiplicative effect \( \exp(y) \) suggests that we should transform the data in order to at least partly remove it prior to clustering; otherwise we will mostly cluster the data according to the magnitude of the transcript level (Eisen et al., 1998; Gibbons and Roth, 2002). A Pearson distance is therefore used prior to clustering as

\[
x_{nm} = \left( \frac{y_{nm} - \bar{y}_n}{\max(\sigma_n, \sigma_0)} \right) \in [-1; 1]
\]

where \( \bar{y}_n \) and \( \sigma_{in}^2 \) are the average and variance of \( y_{nm} \) for the \( n \)-th transcript, respectively, and the max operation with \( \sigma_0 \) small is introduced to avoid amplifying noise for transcripts with constant expression. A soft version is also possible with \( \sqrt{\sigma_{in}^2 + \sigma_{e}^2} \) instead of the max.

The gene effect and multiplicative error for high transcript levels cannot be determined without DNA microarray replicates and thus \( \sigma_{in} = 0.14 \) was based on in-house transcription data (commercial oligonucleotide microarrays from Affymetrix). For modelling purposes it was assumed that \( y \) follows a Gaussian distribution \( \sim N(0, \sigma_{y}^2) \). Under this assumption, the mean of the true transcript level of gene \( \mu_{nm} \) was calculated to \( \mu = 280 \) and the transcript-dependent multiplicative effect to \( \sigma_{y} = 1.5 \) by fitting the same in-house expression data to Equation 13. Thus, we can simulate the influence of noise on the true transcript level for both high and low expression levels.
In this section we make consensus analysis of the simulated dataset and compare the classification error rate with different ‘single shot’ approaches. Furthermore, we use a very large dataset (spotted cDNA microarray) (Gasch et al., 2000) for biological validation and comparison. Finally, we use consensus clustering to re-analyse a DNA microarray dataset (Affymetrix oligonucleotide DNA microarray) (Bro et al., 2003).

4 RESULTS

In this section we make consensus analysis of the simulated dataset and compare the classification error rate with different ‘single shot’ approaches. Furthermore, we use a very large dataset (spotted cDNA microarray) (Gasch et al., 2000) for biological validation and comparison. Finally, we use consensus clustering to re-analyse a DNA microarray dataset (Affymetrix oligonucleotide DNA microarray) (Bro et al., 2003).

4.1 Complete consensus analysis of simulated data

We clustered the simulated and transformed dataset [Equation (14)], and show the properties and the results of the consensus clustering algorithm in Figure 2A–E. As mentioned earlier, we do not have any a priori knowledge of the true number of clusters. Thus, in practice we have to scan different clustering solutions in a user-defined interval. In the current case, we scanned cluster solutions with \( K = 5, \ldots, 20 \) clusters with 15 repetitions resulting in a total of 16.15 \( \times 240 \) runs. As seen in Figure 2A the normalized mutual information between all \( (240 - 1)240/2 = 28680 \) pairs is on average 0.53 indicating a high degree of uncertainty in the VBMoG clustering algorithm. Based on the 240 VBMoG clustering runs we constructed the co-occurrence matrix in Figure 2B weighing all runs equally, i.e. \( p(M_j \mid Z_n) = 1/R \) in Equation (7). We also tried to use the estimate of the marginal likelihood from VB as weights in Equation (7) but that led to a much less stable, close to winner-take-all ensemble, and always very high classification error rates, see also VBMoG ‘single shot’ clustering in Figure 4. This underlines that the VB is not accurate enough to be used for model averaging. For each repetition the most likely number of clusters was determined by the Bayesian Information Criteria (BIC) (see also Supplementary Material). The average of the most likely number of clusters based on the dataset it is desirable to eliminate noise, but in our case we used the simulated dataset to address the robustness of different clustering algorithms.

3.2 Classification error rate

Compared with previous studies (Fred and Jain, 2002, 2003; Strehl and Ghosh, 2002; Monti et al., 2003) the proposed, simulated dataset is difficult to cluster, and a high classification error rate is expected due to large overlap between clusters (Fig. 1). The perfect clustering would determine the number of clusters to 7 with the number of members as given in Table 1. We defined the classification error rate as follows: for a clustering result of the simulated dataset with a given clustering algorithm the correctly clustered transcripts in a single cluster was the maximum number of transcripts identified in one of the 7 clusters in the simulated dataset. We determined the total number of correctly clustered transcripts by summing over all clusters, and hence the classification error rate was determined as the difference between all transcripts (500) and the total number of correctly clustered transcripts divided by the total number of transcripts (500). An alternative to the classification error rate is simply to use the normalized mutual information between the simulated dataset and a given clustering, but the classification error rate is easier to interpret and has strong resemblance to the commonly used false discovery rate used in statistical analysis of DNA microarrays (Tusher et al., 2001; Reiner et al., 2003).

### Table 1. Model parameters for the simulated dataset

<table>
<thead>
<tr>
<th>( \alpha_{m,n} )</th>
<th>( K_1 )</th>
<th>( K_2 )</th>
<th>( K_3 )</th>
<th>( K_4 )</th>
<th>( K_5 )</th>
<th>( K_6 )</th>
<th>( K_7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>60</td>
<td>1</td>
<td>1.3</td>
<td>2.1</td>
<td>1.7</td>
<td>0.9</td>
<td>3.9</td>
<td>2.2</td>
</tr>
<tr>
<td>70</td>
<td>2</td>
<td>3.6</td>
<td>3.1</td>
<td>2.7</td>
<td>1.4</td>
<td>4.1</td>
<td>3.4</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>1.2</td>
<td>1.4</td>
<td>1.5</td>
<td>2.1</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>120</td>
<td>4</td>
<td>0.9</td>
<td>1.2</td>
<td>1.5</td>
<td>1.6</td>
<td>1.2</td>
<td>1.2</td>
</tr>
<tr>
<td>40</td>
<td>5</td>
<td>3.0</td>
<td>1.2</td>
<td>1.0</td>
<td>0.5</td>
<td>2.1</td>
<td>1.3</td>
</tr>
<tr>
<td>80</td>
<td>6</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

The tabulated signal values are given relative to \( \mu = 280 \), i.e. the true transcript level \( \mu_{true} \) is found by multiplying with \( \mu \). The parameter \( \alpha_{m,n} \) is the background noise level of DNA microarray \( m \) and \( K_k \) is the number of members in cluster \( k \).
corresponds to a high degree of co-occurrence of a number of transcripts, i.e. these transcripts are frequently found in the same clusters. As an example, it can be observed that transcripts 1–178 are frequently clustered together and forms a dark square. Within the dark square two new clusters can be observed indicating a possible subdivision of transcripts 1–178 into two new clusters. In turn, the white area outside the dark square indicates that there is a very low probability of finding any of the transcripts 179–500 in the cluster. In contrast to the first observation, transcripts 179–407 did not show a similarly clear pattern with a sharp borderline though transcripts 322–407 suggest two clusters. Finally, transcripts 408–500 indicate two clear clusters.

In the dendrogram in Figure 2C it was observed that the small clusters 4–8 compromising 131 transcripts in the dataset (Fig. 2D) are very similar with respect to the Ward distance. As mentioned earlier, the Ward distance metric is a measure of heterogeneity, and thus a low Ward distance indicated that the transcripts in one of the clusters 4–8 are almost just as likely to emerge in one of the other three clusters. Furthermore, the standard deviation within this cluster was much higher than for the remaining clusters. In Figure 3 it can be seen that merging clusters 4–8 result in a cluster without signal (cluster 4 in row 3), i.e. mean value of 0 in 8 experiments. Clusters 9 and 10 represent $K_2$ in Table 1. We can merge these two clusters based on the transcription profile in Figure 2E, and most importantly, biological validation of the clusters. Thus, the dendrogram can be used to discard and to merge clusters. If we decided to decrease the number of clusters to 7 by merging clusters, it is important that the transcript classification error rate is controlled. Indeed, in the current example a moderate decrease in the number of clusters from 12 to 7 resulted in an increase in classification error rate from 0.094 to 0.120 (47 to 60 classification errors per clustering).

### 4.2 Comparison of clustering approaches

In Figure 4 the classification error rate for some selected clustering algorithms was investigated and compared with the consensus clustering. The simple hierarchical clustering algorithms in Figure 4A had all high classification error rates, but the Ward algorithm was performing considerably better than the remaining algorithms. The classification error rate was 0.272 for 7 clusters decreasing to only 0.010 for 12–15 clusters. A large number of clusters results in many, relatively homogeneous clusters for the Ward algorithm (Karnvar et al., 2002) and consequently a low classification error rate for the proposed generative model for transcription data.

All four classical ‘single shot’ relocation clustering methods in Figure 4B also fail to cluster the simulated dataset correctly and result in very high classification error rates. The classification error rate was only weakly dependent on the number of clusters, and an increase in cluster size did not result in a much better separation and identification of the ground truth (Fig. 4). Most transcripts were always collected in a few major clusters, and hence extra clusters only resulted in the formation of clusters with few transcripts.

To further test the sensitivity of the clustering initialisation, we initialized in the 7 cluster centres, as defined in Table 1. The classification error rates decreased significantly: VBMoG 0.104, genMoG 0.096 and K-means 0.176 (calculations not shown) besides for MoG which ended up in a trivial solution (see Supplementary material). As expected, the probabilistic models VBMoG and genMoG are performing better than K-means when all algorithms
are initialized in the 7 true cluster centres. The more flexible probabilistic models are not limited to only capturing spherical clusters. However, it is worth noting that the average classification error rates are in sharp contrast to the values obtained with random initialization in Figure 4B. Our results suggested that the clusters obtained from ‘single shot’ clustering algorithms represented local maxima, and these maxima were far from the ground truth.

To test this hypothesis, transcripts with low maximum to average signal ratio were removed prior to clustering. For the relocation algorithms this fold change elimination only decreased the classification error rate slightly indicating that local maxima still play a significant role. For all the hierarchical clustering algorithms, on the other hand, classification error rates decreased significantly (see Supplementary material). The ‘single shot’ clustering—essentially maximum likelihood results—can be understood from a bias/variance consideration: less flexible models, in this case K-means, have a lower tendency to overfit data than more flexible models. However, they are also biased towards simpler and often less accurate explanations of data, here spherically shaped K-means clusters. To ensure that the results were not biased by the parameters in the generative model, we performed a sensitivity analysis of the parameters in Table 1. It was confirmed that all results were qualitative identical to the results in Figure 4 (see Supplementary material).

Consensus clustering significantly reduced the classification error rate for all algorithms taken as input to the consensus clustering (Fig. 4C). We confirm the results by Fred and Jain (2002) who showed that consensus clustering with K-means enabled the identification of more complex pattern than with K-means alone. The classification error rate was reduced from 0.176 to 0.142 in Figure 4C. A question not addressed by Fred and Jain (2002) is the formation of clusters based on interpretation of the co-occurrence matrix. The various agglomerative algorithms differ in the objection function (model) they use to determine the sequence of merges (Kamvar et al., 2002). We found that the Ward algorithm and the average linkage for hierarchical clustering were performing best while the single linkage was inferior (results not shown). These results were based on classification error rates of the simulated dataset and biological validation of the datasets in Section 4.3. A model-based interpretation of the results suggests that the co-occurrence matrix is reflecting a situation where the entries are uniformly generated on hyperspheres rather than branched random
Both algorithms group transcripts into unique and reproducible clusters, but they also identify unclassified transcripts, e.g. insignificant clusters and outliers. Clustering with ArrayMiner (default options and the number of clusters specified to 7, including a cluster capturing non-classified transcripts) resulted in a low classification error rate of only 0.096. If the number of clusters was increased to 11 the classification error rate decreased to 0.082. There seems to be a trend that consensus clustering (with VBMoG) outperforms ArrayMiner for larger number of clusters. CLICK correctly identified the number of clusters (default options) to 6 excluding a cluster with unclassified transcripts. In this case the classification error rate was 0.232. However, we found that the CLICK algorithm was more conservative than the other algorithms and resulted in a large cluster of 176 unclassified transcripts. Thus, with our definition of classification error rate the CLICK algorithm is not performing well.

4.3 Consensus clustering of real datasets

We next validated the different clustering algorithms on a real cDNA microarray dataset (Gasch et al., 2000). This dataset was produced by exposing the yeast *S.cerevisiae* to 11 environmental changes and detecting the transcriptional changes over 173 DNA microarrays. The subsequent 3-fold change exclusion showed that 2049 genes had altered transcript level in at least one of the 173 conditions.

This large dataset was analysed with consensus clustering of K-means, where cluster solutions with K = 10, 25 clusters and 25 repetitions leading to a total of 16.25 = 400 runs were scanned and used as input. The average mutual information between runs was 0.68. The result was compared with clustering with a number of classical and commercially available methods (Table 2). The performance of the different algorithms was validated by over-representation of the three Gene Ontology (GO) categories (Ashburner et al., 2000). For each cluster over-representation of GO categories was tested in the cumulative hypergeometric distribution, see e.g. Tavazoie et al. (1999); Smet et al. (2002). The total number of over-represented categories was determined by summing all clusters over all categories with an adjusted P-value below 0.01 (Table 2).

The rational behind this validation was that yeast genes with similar function mostly follow common regulatory mechanism and therefore have common transcript patterns (Eisen et al., 1998; Hughes et al., 2000). The GO describes the cellular process, function and component categories of a gene and the over-representation of a particular GO category in a cluster may thereby be used as a measure of successful clustering of co-regulated genes. The over-representation of different GO categories was tested in the cumulative hypergeometric distribution (Tavazoie et al., 1999; Smet et al., 2002). K-means consensus clustering performed better than other algorithms in all three test examples (Table 2; 10, 13 and 18 clusters). This result was opposed to the clustering of the simulated dataset where ArrayMiner and consensus VBMoG performed better than consensus K-means (Fig. 4C) and probably reflect the fact that the Gasch et al. dataset has a much larger dimensionality than the simulated dataset (2049 transcripts and 173 DNA microarrays compared to 500 transcripts and 8 DNA microarrays). K-means is a more robust method and therefore better suited for multi-dimensional datasets for the ‘single shot’ cases. ArrayMiner and consensus VBMoG, on the other hand, rely
on Mixtures of Gaussians and therefore possess the ability to describe data more sophisticated than K-means (Fig. 4). However, this characteristic of MoG is apparently a drawback when the dimensionality of the dataset increases. ‘Single shot’ VBMoG performed poorly on the Gasch et al. dataset with a mutual information between runs that was less than 0.05 (Table 2). Consensus clustering with VBMoG consequently requires a very large number of repetitions before a stable solution can be obtained (see Supplementary material). For low mutual information between runs it seems like a more prudent strategy to go for a local search method that the solutions are very different from run to run. In Figure 5A the variation in the number of optimal clusters reflect minimum difference between adjacent clusters (Bar-Joseph et al., 2003) with an adjusted P-value (Benjamini and Hochberg, 1995) below 0.01 were considered significant. The tabulated values are the number of significant categories summed over all clusters.

Another characteristic of the consensus clustering algorithms was the ability to cluster and exclude transcripts in the same step. Transcript datasets are often sorted prior to clustering either according to fold change or by a statistical method (Tusher et al., 2001), which may lead to exclusion of false negative data. We therefore re-analysed a time course experiment from yeast treated with lithium chloride (LiCl). The budding yeast S. cerevisiae was grown on galactose and exposed to a toxic concentration of LiCl at time 0, and the cells were harvested for transcription analysis at time 0, 20, 40, 60 and 140 min after the pulse (Bro et al., 2003). In the current analysis we used consensus clustering on all 5710 detectable transcripts without prior data exclusion. The data were clustered as illustrated with the simulated dataset in Section 4.1. The only exception was that we scanned cluster solutions with K = 10, . . . , 40 and 50 repetitions leading to a total of 31.50 × 10^3 solutions with K = 10, . . . , 40 and 50 repetitions leading to a total of 31.50 × 10^3. For each repetition the most likely number of clusters was determined by the BIC. The average of the most likely number of clusters based on the 50 repetitions was 22 with a standard deviation of 10. Once again, the result indicates that the posterior averaging has not been performed correctly; i.e. the variation in the number of optimal clusters reflect that the solutions are very different from run to run. In Figure 5A the co-occurrence matrix has been sorted according to the 22 clusters to reflect minimum difference between adjacent clusters (Bar-Joseph et al., 2001). The 22 clusters consisted of upregulated clusters.

Table 2. Clustering and biological validation

<table>
<thead>
<tr>
<th>Algorithm and settings</th>
<th>Clusters</th>
<th>Process</th>
<th>Function</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means consensus</td>
<td>10</td>
<td>536</td>
<td>229</td>
<td>141</td>
</tr>
<tr>
<td>ArrayMiner a,b</td>
<td>10</td>
<td>484</td>
<td>236</td>
<td>151</td>
</tr>
<tr>
<td>Hierarchical (Ward)</td>
<td>10</td>
<td>342</td>
<td>147</td>
<td>117</td>
</tr>
<tr>
<td>Click and Expander c</td>
<td>10</td>
<td>282</td>
<td>122</td>
<td>89</td>
</tr>
<tr>
<td>K-means (single shot)</td>
<td>10</td>
<td>275</td>
<td>101</td>
<td>113</td>
</tr>
<tr>
<td>VBMoG (single shot)</td>
<td>10</td>
<td>86</td>
<td>42</td>
<td>15</td>
</tr>
<tr>
<td>K-means consensus</td>
<td>13</td>
<td>561</td>
<td>259</td>
<td>158</td>
</tr>
<tr>
<td>K-means (single shot)</td>
<td>13</td>
<td>444</td>
<td>171</td>
<td>127</td>
</tr>
<tr>
<td>Hierarchical (Ward)</td>
<td>13</td>
<td>372</td>
<td>156</td>
<td>114</td>
</tr>
<tr>
<td>Adaptive quality-based a</td>
<td>13</td>
<td>260</td>
<td>110</td>
<td>101</td>
</tr>
<tr>
<td>VBMoG (single shot)</td>
<td>13</td>
<td>80</td>
<td>45</td>
<td>17</td>
</tr>
<tr>
<td>K-means consensus</td>
<td>18</td>
<td>595</td>
<td>274</td>
<td>180</td>
</tr>
<tr>
<td>K-means (single shot)</td>
<td>18</td>
<td>483</td>
<td>174</td>
<td>160</td>
</tr>
<tr>
<td>Hierarchical (Ward)</td>
<td>18</td>
<td>454</td>
<td>184</td>
<td>177</td>
</tr>
<tr>
<td>CAGED version 1.0d</td>
<td>18</td>
<td>426</td>
<td>163</td>
<td>136</td>
</tr>
<tr>
<td>VBMoG (single shot)</td>
<td>18</td>
<td>105</td>
<td>64</td>
<td>45</td>
</tr>
</tbody>
</table>

For each algorithm with a fixed number of clusters (Clusters) the over-represented Gene Ontology categories (Process, Function and Component) (Ashburner et al., 2000) with an adjusted P-value (Benjamini and Hochberg, 1995) below 0.01 were considered significant. The tabulated values are the number of significant categories summed over all clusters.

This algorithm is not assigning all genes to a cluster. Genes not classified are considered one cluster, and consequently the chosen number of clusters in the algorithm is chosen to be one less than the tabulated value.

Algorithm reference: Ramoni et al. (2002).

The yeast S. cerevisiae was treated with a toxic concentration of LiCl at time 0. (A) Co-occurrence matrix of the 5710 ORFs. The transcripts have been sorted with respect to the 22 clusters using optimal leaf ordering (Bar-Joseph et al., 2001). (B) dendrogram of the 22 clusters. (C) transcription profile for all 22 clusters shown as transformed values between −1 and 1, where 0 indicates the average expression level. The bars give the standard deviation with the clusters.
validation of single runs to validation of the consensus result, e.g. validation by human intervention is then moved from cumbersome different algorithms (Kaminski and Friedman, 2002). The result is a statistical formalisation of the practical clustering approach using namely conservative data exclusion.

Both up- and downregulated genes were further subdivided into clusters with immediate or delayed response to the lithium pulse, revealing a better resolution of the data than in the initial analysis (Bro et al., 2003). It was thereby clear that genes in the carbon metabolism are upregulated while genes involved in ribosome biogenesis are downregulated as an immediate response to the LiCl pulse (clusters 6–8 and 22). After 40 min genes in clusters 2 and 3 were upregulated, while those in cluster 20 started to be downregulated. Many of the genes in clusters 2 and 3 were involved in protein catabolism and transport through the secretory pathway, while genes involved in amino acid metabolism and replication were found in cluster 20. Finally, after 60–140 min genes involved in cell wall biosynthesis, invasive growth and autophagy in clusters 1, 4, 9 and 10 were upregulated. Hence, it was clear that there were functional differences between genes with immediate and delayed response and that this separation was greatly aided by consensus clustering.

The current data analysis suggested more than the original 1390 identified ORFs had altered expression in response to the chemical stress. In total 2106 genes were found in clusters of upregulated genes, 1232 in clusters of downregulated genes and 794 in clusters of genes with a transient response. This large discrepancy between the original data analysis and the current one was mostly owed to exclusion of transcripts without a three-fold change in expression. Fold-change exclusion did not appear to be necessary in the current analysis, and analysis of the complete dataset led to identification of more transcripts that were affected by LiCl. Consensus clustering thereby bypasses a major challenge in transcription analysis, namely conservative data exclusion.

5 DISCUSSION

A good clustering has predictive power: clues to the function of unknown genes can be obtained by associating the function of the known co-regulated genes. Thus, the chosen clustering algorithm must be reliable in order to distinguish between different effects when small changes in the transcript level are significant (Jones et al., 2003), and secondly the results must be presented in a form which makes biological interpretation and validation accessible.

We showed that classical and fast 'single shot' clustering produced poor cluster results for a realistic simulated dataset based on biological data. Initialization in the cluster centres and the success of ArrayMiner (Falkenauer and Marchand, 2003), which uses a genetic algorithm for optimizing the Mixture of Gaussians objective function, indicates that local minima is the main reason why single run relocation algorithm fails. Thus, the increased computation time for ArrayMiner is clearly beneficial for the clustering result. The consensus approach taken in this paper can be seen as a statistical formalisation of the practical clustering approach using different algorithms (Kaminski and Friedman, 2002). The result is a consensus clustering, where common traits over multiple runs are amplified and non-reproducible features suppressed. The biological validation by human intervention is then moved from cumbersome validation of single runs to validation of the consensus result, e.g. to choosing the clusters of interest in a hierarchical clustering.

Averaging over multiple clustering runs enables the clusters to capture more complicated shapes than any other single clustering algorithm (Fred and Jain, 2002, 2003) as shown in Figure 4 where the consensus of the K-means outperformed K-means initialized in the true cluster centres. Consensus clustering, taking any cluster ensemble as input, offers a very simple way to combine results from different methods and can thus be expected to a larger scope of validity of any single method. It is not likely that one method is capturing all biological information (Goldstein et al., 2002), and hence consensus clustering is a valuable tool for discovering ever emerging patterns in the data. The drawback of consensus clustering is the increased computation time, but the considerable amount of time investigated in biological interpretation justifies a longer computation time.

The consensus clustering algorithm does not determine the number of clusters unambiguously though optimality criteria exist (Fred and Jain, 2002, 2003), but the dendrogram is a useful and pragmatic tool for biological interpretation of the results (Eisen et al., 1998). In DNA microarray analysis the 'correct' number of clusters depends upon the questions asked. The advantage of the dendrogram representation is that the biological analyst can choose the scale and here the purpose of the consensus method is simply to provide a robust multi-scale clustering. For example, in Figure 3 (clusters 1 and 2) and Figure 5 (clusters 6 and 7) the clusters are very similar in shape, but only a biological validation can justify the existence of one or two clusters. As discussed in Falkenauer and Marchand (2003) standard hierarchical clustering is based on a 'bottom-up' approach where smaller clusters at the lower level are merged into bigger clusters. Thus, the dendrogram is constructed based on the local structure with no regard to the global structure of the expression data—in consensus clustering it is the other way around: the robust, local structure is emerging out of the global picture.

In conclusion, with consensus clustering we have achieved the 2-fold aim of a robust clustering, where gene expression data are divided into robust and reproducible clusters and at the same time attaining the advantages of hierarchical clustering. Clusters can be visualized in a dendrogram and analysed on multiple scales in a biological context.

ACKNOWLEDGEMENTS

The authors would like to thank Emanuel Falkenauer for providing a free version of the ArrayMiner software package. Thomas Grotkjær would like to acknowledge the Novozymes Bioprocess Academy for financial support.

Conflict of Interest: none declared.

REFERENCES