BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–11

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

Thomas Grotkjær [a],[*] Ole Winther [b], Birgitte Regenberg [a], Jens Nielsen [a] and Lars Kai Hansen [b]

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

ABSTRACT

Motivation: Hierarchical and relocation clustering (e.g. K-means and self-organising 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 that are dependent on the initialisation 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 visualisation 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.

Contact: tg@biocentrum.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 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) (Tamayo *et al.*, 1999), Mixtures of Gaussians (MacKay, 2003), etc. give results that depend upon the initialialisation 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 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 & 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 & 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

[*]to whom correspondence should be addressed

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 a Monte Carlo or genetic algorithms (Falkenauer & 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) is described in Suppl. Material since VBMoG and its maximum likelihood counterpart are already well-established in the DNA microarray literature (McLachlan *et al.*, 2002; Ghosh & 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 & Jain (2002, 2003); Strehl & Ghosh (2002); Monti *et al.* (2003), see also the discussion of related work in section 5 of Strehl & 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 & 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 Suppl. Material. The probabilistic (or soft) assignment is a vector $\mathbf{p}(\mathbf{x}) = [p(1|\mathbf{x}_n), \ldots, p(K|\mathbf{x}_n)]^T$ giving the probabilities of the $k = 1, \ldots, K$ cluster labels for an object (M experiment data vectors) $\mathbf{x} = [x_1, \ldots, x_M]^T$. One way to model $p(k|\mathbf{x})$ is through a mixture model

$$p(k|\mathbf{x}) = \frac{p(k)p(\mathbf{x}|k)}{\sum_{k'=1}^{K} p(k')p(\mathbf{x}|k')} \; . \qquad (1)$$

Hard assignments are the degenerate case of the soft assignment where one component, say, $p(k|\mathbf{x})$ is one, but a hard assignment can also be obtained from $a(\mathbf{x}) = \mathrm{argmax}_k\, p(k|\mathbf{x})$, 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: $\mathcal{D}_N \equiv \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of size N examples (number of transcripts). We therefore write the mixture model as an explicit function of the set of model parameters $\boldsymbol{\theta}$ and the model $\mathcal{M}$:

$$p(\mathbf{x}|\boldsymbol{\theta}, \mathcal{M}) = \sum_{k=1}^{K} p(k|\boldsymbol{\theta}, \mathcal{M})p(\mathbf{x}|k, \boldsymbol{\theta}, \mathcal{M}) \; . \qquad (2)$$

The model $\mathcal{M}$ 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 $\boldsymbol{\theta}$.

In *maximum likelihood* the parameters are found by maximising the likelihood of the parameters: $\boldsymbol{\theta}^{\mathrm{ML}} = \mathrm{argmax}_{\boldsymbol{\theta}}\, p(\mathcal{D}_N|\boldsymbol{\theta}, \mathcal{M})$ with the assumption of independent examples $p(\mathcal{D}_N|\boldsymbol{\theta}, \mathcal{M}) = \prod_{n=1}^{N} p(\mathbf{x}_n|\boldsymbol{\theta}, \mathcal{M})$ and assignment probabilities are given by $p(k|\mathbf{x}, \boldsymbol{\theta}^{\mathrm{ML}}, \mathcal{M}) \propto p(k|\boldsymbol{\theta}^{\mathrm{ML}}, \mathcal{M})p(\mathbf{x}|k, \boldsymbol{\theta}^{\mathrm{ML}}, \mathcal{M})$. This naturally leads to a set of iterative expectation maximisation (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 $p(\boldsymbol{\theta}|\mathcal{D}_N, \mathcal{M}) = \frac{p(\mathcal{D}_N|\boldsymbol{\theta}, \mathcal{M})p(\boldsymbol{\theta}|\mathcal{M})}{p(\mathcal{D}_N|\mathcal{M})}$, where $p(\boldsymbol{\theta}|\mathcal{M})$ is the prior over model parameters and

$$p(\mathcal{D}_N|\mathcal{M}) = \int d\boldsymbol{\theta}\, p(\mathcal{D}_N|\boldsymbol{\theta}, \mathcal{M})p(\boldsymbol{\theta}|\mathcal{M}) \qquad (3)$$

is the likelihood of the model (marginal likelihood or evidence). We can find the cluster assignment probability for a data point $\mathbf{x}$ by averaging out the parameters using the posterior distribution $p(k|\mathbf{x}, \mathcal{D}_N, \mathcal{M}) = \frac{p(k,\mathbf{x}|\mathcal{D}_N,\mathcal{M})}{p(\mathbf{x}|\mathcal{D}_N,\mathcal{M})}$. Either way, we calculate the soft assignments and obtain an assignment matrix $\mathbf{P} = [\mathbf{p}(\mathbf{x}_1), \ldots, \mathbf{p}(\mathbf{x}_N)]$ 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 $\mathcal{M} \propto p(\mathcal{M})p(\mathcal{D}_N|\mathcal{M})$, where $p(\mathcal{M})$ is the prior probability of the model. For non-trivial models the evidence is computationally intractable although asymptotic expressions exist. The variational Bayes (VB) framework aims at approximating the average over the parameters, but it unfortunately underestimates the width of the posterior distribution of $\boldsymbol{\theta}$ (MacKay, 2003). As a consequence multiple modes of the approximate marginal likelihood exists for this flexible model. It means that depending upon the initialisation, two runs r and r' give different estimates of the marginal likelihood $p_{\text{app}}(\mathcal{D}_N|r, \mathcal{M}) \neq p_{\text{app}}(\mathcal{D}_N|r', \mathcal{M})$. 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 $[\mathbf{P}_1, \ldots, \mathbf{P}_R]$. We may also have posterior probabilities for each run $p(\mathcal{M}_r|\mathcal{D}_N) \propto p(\mathcal{D}_N|\mathcal{M}_r)p(\mathcal{M}_r)$, where $\mathcal{M}_r$ 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* $C_{nn'}$ is the probability that transcript n and n' are in the same cluster

$$\begin{aligned} C_{nn'} &= \sum_{r=1}^{R} \sum_{k=1}^{K_r} p(k|\mathbf{x}_n, r)p(k|\mathbf{x}_{n'}, r)p(\mathcal{M}_r|\mathcal{D}_N) \\ &= \sum_{r=1}^{R} [\mathbf{P}_r^T \mathbf{P}_r]_{nn'} p(\mathcal{M}_r|\mathcal{D}_N) \ . \end{aligned} \tag{4}$$

We can convert the co-occurrence matrix into a *transcript-transcript distance matrix* $D_{nn'} = 1 - C_{nn'}$. 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 normalised mutual information can be used to quantify the significance of the different clustering runs, i.e. how diverse are the partitionings. Strehl & 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 & 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

$$M_{rr'} = \sum_{kk'} p_{rr'}(k, k') \log \frac{p_{rr'}(k, k')}{p_r(k)p_{r'}(k')} \ , \tag{5}$$

where the joint probability of label k and k' in runs r and r' is calculated as $p_{rr'}(k, k') = \frac{1}{N} \sum_n p(k|\mathbf{x}_n, r)p(k'|\mathbf{x}_n, r')$ and the marginal probabilities as $p_r(k) = \frac{1}{N} \sum_n p(k|\mathbf{x}_n, r) = \sum_{k'} p_{rr'}(k, k')$. We can also introduce a normalised version of this quantity:

$$M_{rr'}^{\text{norm}} = \frac{M_{rr'}}{\max(M_r, M_{r'})} \in [-1; 1] \ , \tag{6}$$

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 optimising the mutual information directly is NP-hard and the method suggested above may be viewed as an approximation to do this (Strehl & Ghosh, 2002).

The average mutual information

$$\overline{M}^{\text{norm}} = \frac{2}{R(R-1)} \sum_{r,r',r>r'} M_{rr'}^{\text{norm}} \tag{7}$$

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 repetions 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-occurence matrix. Requiring that the 'true positive' entries should be significantly larger than 'false positive' determines R in terms of $\overline{M}^{\text{norm}}$. See Suppl. 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 & 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 & Durbin, 2003). Here we consider a slight generalisation of this model by including a multiplicative gene effect $\exp(\gamma_n)$ on the 'true' transcript level μ_{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

$$y_{nm} = \alpha_m + \mu_{nm} \exp(\gamma_n + \eta_{nm}) + \varepsilon_{nm} \ , \qquad (8)$$

where α_m is the mean background noise of DNA microarray m and $\eta \sim \mathcal{N}(0, \sigma_\eta^2)$ and $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ are biological and technical dependent multiplicative and additive errors that follow Gaussian distributions $\mathcal{N}$ with mean 0, and variance σ_η^2 and σ_ε^2, respectively.

The parameters α_m and σ_ε can be estimated by considering the probe sets with lowest intensity (Rocke & Durbin, 2001). The rather strong influence of transcript dependent multiplicative effect $\exp(\gamma)$ 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 & Roth, 2002). A Pearson distance is therefore used prior to clustering as

$$x_{nm} = \frac{y_{nm} - \bar{y}_n}{\max(\sigma_n, \sigma_0)} \in [-1; 1] \qquad (9)$$

where $\bar{y}_n$ and σ_n^2 are the average and variance of y_{nm} for the nth transcript, respectively, and the max operation with σ_0 small is introduced to avoid amplifying noise for transcripts with constant expression. A soft version is also possible with $\sqrt{\sigma_n^2 + \sigma_0^2}$ instead of the max.

The gene effect and multiplicative error for high transcript levels cannot be determined without DNA microarray replicates and thus $\sigma_\eta = 0.14$ was based on in-house transcription data (commercial oligonucleotide microarrays from Affymetrix). For modelling purposes it was assumed that γ follows a Gaussian distribution $\sim \ \mathcal{N}(0, \sigma_\gamma^2)$. Under this assumption, the mean of the true transcript level of gene μ_{nm} was calculated to $\bar{\mu} = 280$ and the transcript dependent multiplicative effect to $\sigma_\gamma = 1.5$ by fitting the same in-house expression data to Eq. 8. Thus, we can simulate the influence of noise on the true transcript level for both high and low expression levels.

3.1 Simulated dataset

A simulated dataset was generated by using the generative model in Eq. (8) followed by transformation according to Eq. (9). The parameters for the true transcript level in the simulated dataset with 500 transcripts and 8 DNA microarrays are given in Table 1 and plotted as clusters with means and deviations in Figure 1. The transcript level of transcript $n = 1, \ldots, 400$ was divided into 6 true clusters with K_k transcripts and a relative transcript level μ_{nm} as shown in Table 1. For the transcripts $n = 401, \ldots, 500$ we used a mean true transcript level, μ, of 280. For cluster 7 there was no true change in transcript level and variance in the transcript level was only due to noise imposed by the model. Clearly, before using any clustering algorithm on a dataset it is desirable to eliminate noise, but in our case we used the simulated dataset to address the robustness of different clustering algorithms.

Table 1. Model parameters for the simulated dataset y_{nm}. The parameter α_m is the background noise level of DNA microarray m and K_k is the number of members in cluster k.

	α_m	39	35	33	35	34	34	34	31
K_k	k/m	1	2	3	4	5	6	7	8
60	1	1.3	2.1	1.7	0.9	3.9	2.2	1.9	1.4
70	2	3.6	3.1	2.7	1.4	4.1	3.4	3.1	2.6
30	3	1.2	1.4	1.5	2.1	1.0	1.0	1.1	1.1
120	4	0.9	1.2	1.5	1.6	1.2	1.2	1.3	3.3
40	5	3.0	1.2	1.0	0.5	2.1	1.3	1.1	1.1
80	6	0.4	0.4	0.4	0.4	0.5	0.5	0.5	2.5

The tabulated signal values are given relative to $\bar{\mu} = 280$, i.e. the true transcript level μ_{nm} is found by multiplying with $\bar{\mu}$.

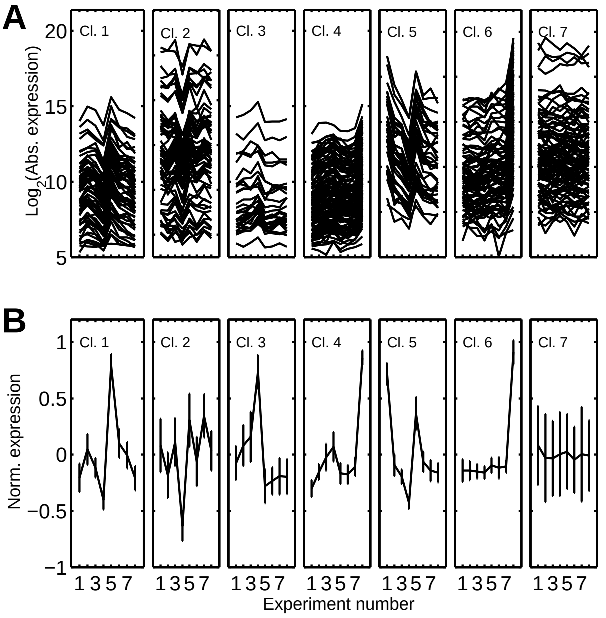

Fig. 1. Simulated dataset with 500 transcripts and 8 DNA microarrays divided into the 6 true clusters and a cluster without signal, i.e. pure noise (cluster 7). Note, only odd numbers are shown on the x-axis. **a**. Log$_2$ transformed transcription profile in each of the 7 clusters (Eq. 8). **b**. Means and deviation of the transformed dataset (Eq. 9).

3.2 Classification error rate

Compared to previous studies (Fred & Jain, 2002; Strehl & Ghosh, 2002; Fred & Jain, 2003; 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 (Figure 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 normalised 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).

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 (Eq. 9), and show the properties and the results of the consensus clustering algorithm in Figure 2a–e.

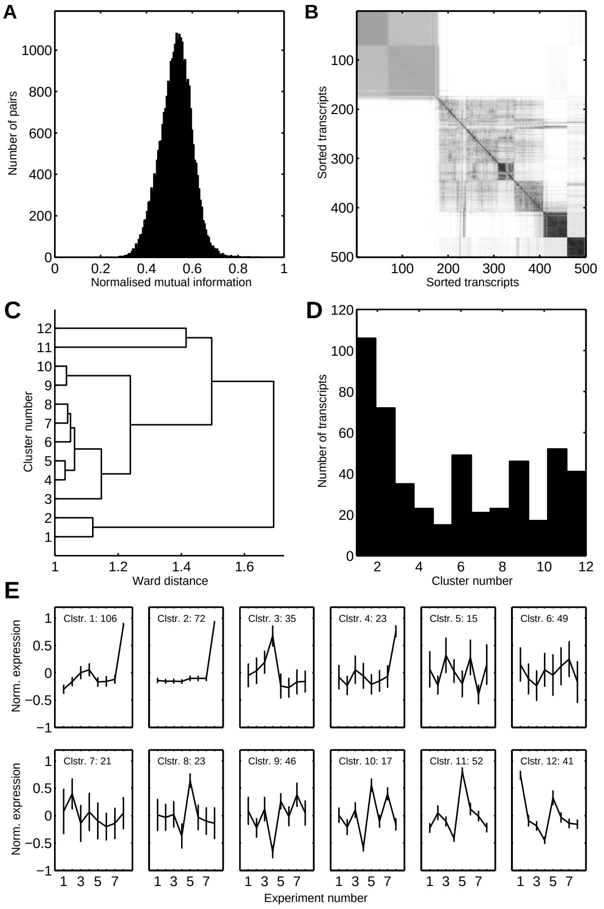

Fig. 2. Overview of the consensus clustering mechanism of a simulated dataset with 500 transcripts and 6 true clusters, including a cluster with pure noise. The consensus clustering was based on 240 VBMoG 'single shot' clustering runs. See text for additional details. **a**. Normalised mutual information between the 240 clustering runs. **b**. Co-occurrence matrix of the sorted transcripts using optimal leaf ordering (Bar-Joseph *et al.*, 2001). A black area corresponds to a high degree of co-occurrence, i.e. these transcripts tend to cluster in all clustering runs. The white area indicates that these transcripts never cluster together (see text for more details). **c**. The co-occurrence matrix is assembled into a dendrogram with 12 leaves, or clusters using the Ward distance. **d**. Histogram of the cluster size. **e**. Normalised transcription profile for all 12 clusters shown as normalised values between -1 and 1, where 0 indicates the average expression level. The bars give the standard deviation within the clusters. Note the high standard deviation within noisy clusters 4–8.

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 $16 \cdot 15 = 240$ runs. As seen in Figure 2a the normalised mutual information between all $(240-1)240/2 = 28,680$ 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. $p(\mathcal{M}_r|\mathcal{D}_N) = 1/R$ in Eq. 4. We also tried to use the estimate of the marginal likelihood from VB as weights in Eq. 4, 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 Suppl. Material). The average of the most likely number of clusters based on the 15 repetitions was 12 with a standard deviation of 3. This result also indicates that that the posterior averaging has not been performed correctly, and hence, 12 clusters is only considered a conservative and pragmatic starting point for further biological validation. For a real, biological dataset the problem becomes even worse (see section 4.3).

For improved visualisation we sorted the co-occurrence matrix with the optimal leaf ordering algorithm (Bar-Joseph *et al.*, 2001) implemented in `Matlab` (Venet, 2003). In Figure 2b a dark square 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 (Figure 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 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 results 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 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 were investigated and compared to 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 (Kamvar *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 results 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 (Figure 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 initialised 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 Suppl. Material). As expected, the probabilistic models VBMoG and genMoG are performing better than K-means when all algorithms are initialised 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 values are in sharp contrast to the average classification error rates obtained with random initialisation 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. 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 Suppl. material).

Consensus clustering significantly reduced the classification error rate for all algorithms taken as input to the consensus clustering (Figure 4c). We confirm the results by Fred & 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. The simulated dataset was also clustered with the `ArrayMiner` (Falkenauer & Marchand, 2003) (see also `http://www.optimaldesign.com`) and `CLICK` (Sharan *et al.*, 2003), clustering algorithms especially designed for analysis of DNA microarray data. 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

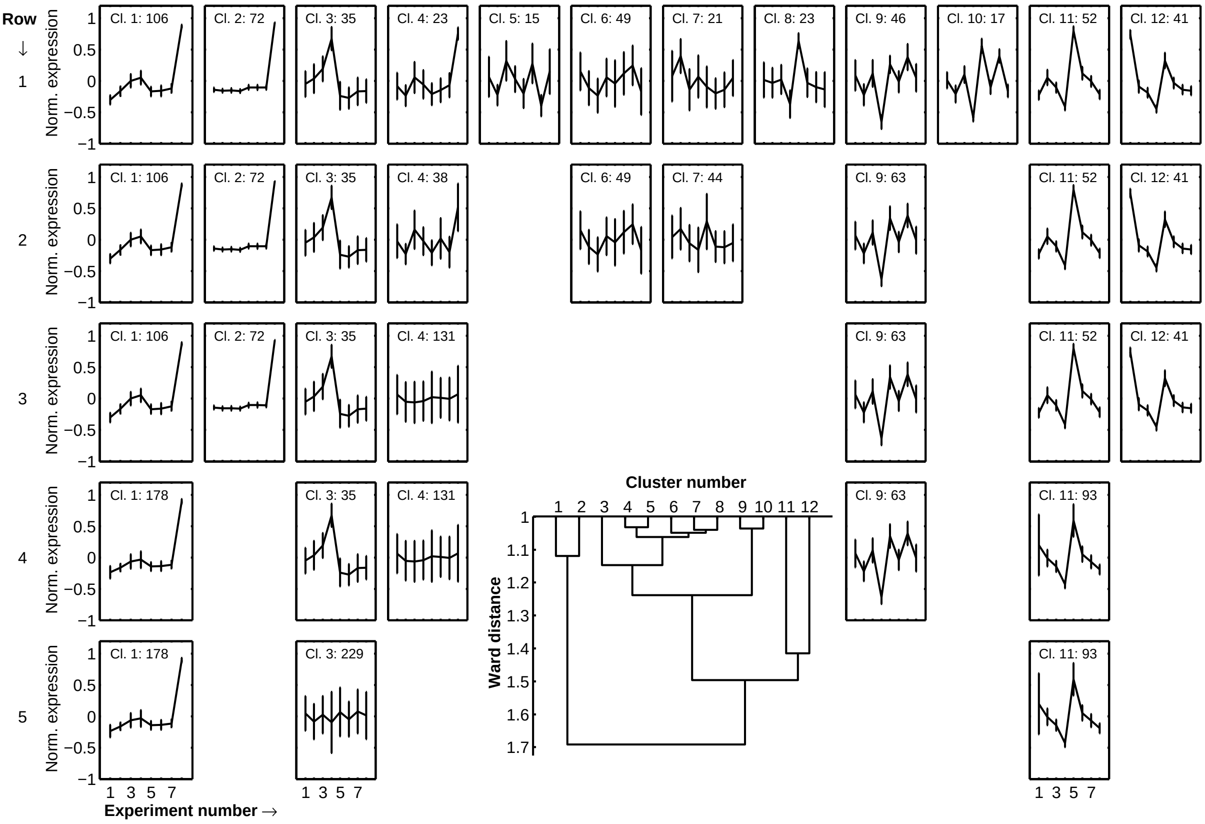

Fig. 3. Effect of merging clusters from Figure 2. The 12 initial clusters are merged to three clusters in five steps, indicated with rows to the left. In rows number 1–5 the number of clusters is 12, 9, 7, 5 and 3, respectively. When two or more clusters are merged, the lowest cluster label is preserved, e.g. clusters 9 and 10 in row 1 are merged into cluster 9 in row 2 with 46+17=63 transcripts, and clusters 4, 6 and 7 in row 2 into cluster 4 in row 3 with 38+49+44=131 transcripts. Clusters which are not merged are transferred horizontally from one row to the row below. Cluster 4 in rows 3 and 4 is composed of the noisy clusters 4–8. It is observed that the average normalised expression value is approximately 0 with a large standard deviation.

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 2,049 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 $K = 10, \ldots, 25$ clusters and 25 repetitions leading to a total of $26 \cdot 25 = 650$ runs were scanned and used as input. The average mutual information between runs was 0.68. The result was compared to clustering with a number of classical and commercially available methods (Table 2). The performance of the different algorithms was validated by the number of over-represented Gene Ontology (GO) categories (Ashburner *et al.*, 2000) in each cluster. The rational behind this validation was that yeast genes with similar function mostly obey 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

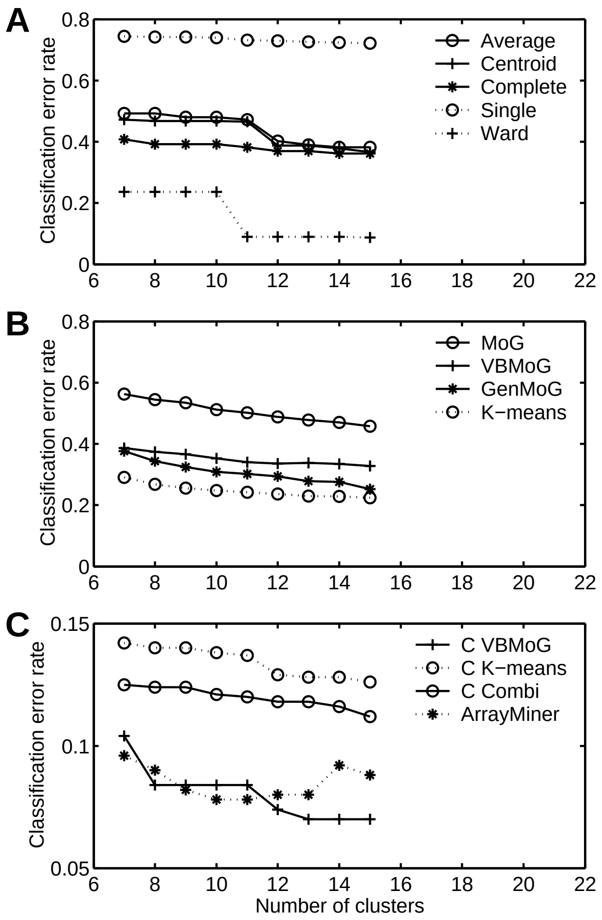

Fig. 4. Classification error rate as a function of number of clusters for selected clustering methods. **a**. Five hierarchical clustering methods. All standard algorithms, except from the Ward algorithm, have a tendency to form one large cluster and a number of small clusters resulting in high classification error rates (see also text). **b**. Four relocation 'single shot' clustering methods with fixed number of clusters. MoG is standard Mixture of Gaussians and GenMog is the generalised Mixture of Gaussian algorithm (Hansen *et al.*, 2000). The classification error rate was calculated as the mean value of 300 clustering runs. **c**. Consensus clustering (denoted with C) of VBMoG, K-means and Combi (inputs from both the VBMoG and K-means algorithms). Each consensus solution was based on scanning with $K = 5, \ldots, 20$ clusters with 15 repetitions, and the classification error rate was calculated as the mean value of 50 clustering runs. The classification error rate is compared with the `ArrayMiner` (Falkenauer & Marchand, 2003) where unclassified genes in the output have been collected in one single cluster. Note, there are much smaller classification error rates in C (y-axis scale changed) compared to the algorithms in **a** and **b**.

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 that 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 (Figure 4c) and probably reflect the fact that the Gasch *et al.* dataset has a much larger dimensionality than the simulated dataset (2,049 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 (Figure 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 repetition before a stable solution can be obtained (see Supp. Material). For low mutual information between runs it seems like a more prudent strategy to go for a local search method as in `ArrayMiner` compared to the consensus strategy. The advantage of K-means for analysis of this large dataset was also evident in the 'single shot' analysis of the Gasch *et al.* data where K-means improved the number of over-represented GO categories compared to 'single shot' VBMoG (Table 2).

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 minutes after the pulse (Bro *et al.*, 2003).

In the original dataset 1,390 open reading frames (ORFs) were found to to have altered expression in response to LiCl, of which 664 were found to be down-regulated and 725 up-regulated (Bro *et al.*, 2003). In the current analysis we used consensus clustering on all 5,710 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 $K = 10, \ldots, 40$ and 50 repetitions leading to a total of $31 \cdot 50 = 1,550$ runs. 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

Table 2. Clustering and biological validation. For each algorithm with a fixed number of clusters (Clusters) the over-represented Gene Ontology categories (Process, Function and Component) (Ashburner *et al.*, 2001) with a P-value below 0.01 were considered significant. The tabulated values are the number of significant categories summed over all clusters.

Algorithm and settings	Clusters	Process	Function	Component
K-means consensus	10	536	229	141
ArrayMiner[1,2]	10	484	236	151
Hierarchical (Ward)	10	342	147	117
Click and Expander[1,3]	10	282	122	89
K-means (single shot)	10	275	101	113
VBMoG (single shot)	10	86	42	15
K-means consensus	13	561	259	158
K-means (single shot)	13	444	171	127
Hierarchical (Ward)	13	372	156	114
Adaptive quality-based[1]	13	260	110	101
VBMoG (single shot)	13	80	45	17
K-means consensus	18	595	274	180
K-means (single shot)	18	483	174	160
Hierarchical (Ward)	18	454	184	177
CAGED version 1.0[4]	18	426	163	136
VBMoG (single shot)	18	105	64	45

[1] 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.
[2] Algorithm reference: Falkenauer & Marchand (2003).
[3] Algorithm reference: Sharan *et al.* (2003).
[4] Algorithm reference: Ramoni *et al.* (2002).

indicates that that the posterior averaging has not been performed correctly; that is, 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 up-regulated clusters (Figure 5b and Figure 5c, clusters 1–4 and 7–10), three down-regulated clusters (Figure 5b, clusters 20–22) plus a set of clusters with ORFs that had a transient response to LiCl (Figure 5c, clusters 6 and 11–13). The remaining seven clusters did not have a clear profile and were therefore considered as noise (Figure 5c, clusters 5 and 14–19).

Both up- and down-regulated 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 up-regulated while genes involved in ribosome biogenesis are down-regulated as an immediate response to the LiCl pulse (clusters 6–8 and 22). After 40 minutes genes in clusters 2 and 3 were up-regulated, while those in cluster 20 started to be down-regulated. 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 to 140 minutes genes involved in cell wall biosynthesis, invasive growth and autophagy in clusters 1, 4, 9 and 10 were up-regulated. 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 1,390 identified ORFs had altered expression in response to the chemical stress. In total 2,106 genes were found in clusters of up-regulated genes, 1,169 in clusters of down-regulated 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 more ORFs were found to improve the analysis. Consensus clustering thereby bypass 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. Initialisation in the cluster centres and the success of ArrayMiner (Falkenauer & Marchand, 2003), which uses a genetic algorithm for optimising 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 & 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 & Jain, 2002, 2003) as shown in Figure 4 where the consensus of the K-means outperformed K-means initialised in the true cluster centres. Consensus clustering, taking any cluster ensemble as input, offers a very

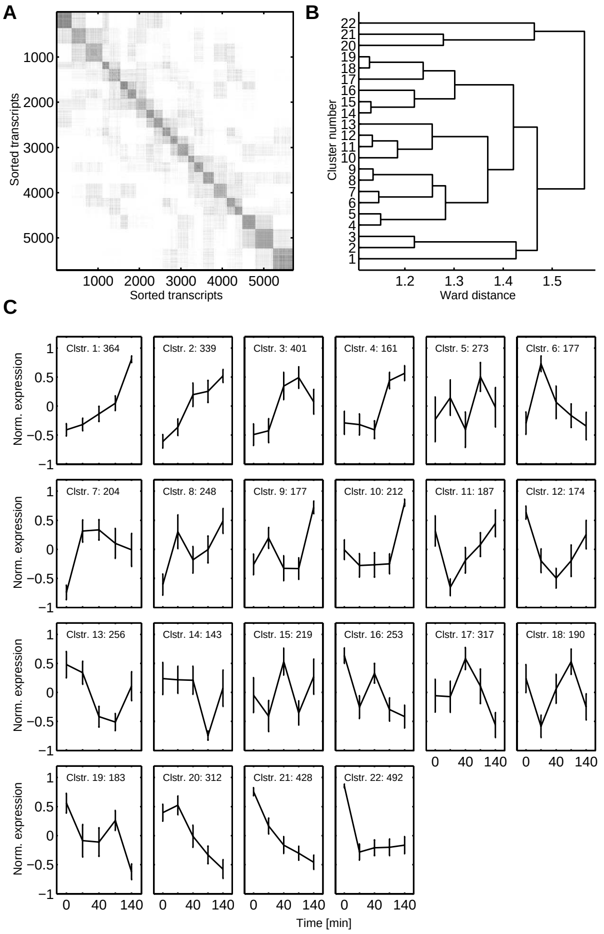

Fig. 5. Overview of a real whole genome consensus clustering result. The yeast *S. cerevisiae* was treated with a toxic concentration of LiCl at time 0. **a**. Co-occurrence matrix of the 5,710 ORFs. The transcripts have been sorted with respect to the 22 clusters using optimal leaf ordering (Bar-Joseph *et al.*, 2001). **b**. Dendrogram of the 22 clusters. **c**. Normalised transcription profile for all 22 clusters shown as normalised values between -1 and 1, where 0 indicates the average expression level. The bars give the standard deviation with the clusters.

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 & 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 & 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 two-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 visualised in a dendrogram and analysed on multiple scales in a biological context.

ACKNOWLEDGEMENT

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

REFERENCES

Ashburner, M., Ball, C. A. & Blake, J. A. (2001) Creating the gene ontology resource: design and implementation - the gene ontology consortium. *Genome Res.,* **11** (8), 1425–1433.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M. & Sherlock, G. (2000) Gene Ontology: tool for the unification of biology. the gene ontology consortium. *Nat. Genet.,* **25** (1), 25–29.

Attias, H. (2000) A variational Bayesian framework for graphical models. Adv. Neur. Info. Proc. Sys. MIT Press.

Bar-Joseph, Z., Gifford, D. K. & Jaakkola, T. S. (2001) Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics,* **17 Suppl 1**, S22–S29.

Bro, C., Regenberg, B., Lagniel, G., Labarre, J., Montero-Lomeli, M. & Nielsen, J. (2003) Transcriptional, proteomic, and metabolic responses to lithium in galactose-grown yeast cells. *J. Biol. Chem.,* **278** (34), 32141–32149.

DeRisi, J. L., Iyer, V. R. & Brown, P. O. (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science,* **278** (5338), 680–686.

Dubey, A., Hwang, S., Rangel, C., Rasmussen, C. E., Ghahramani, Z. & Wild, D. L. (2004) Clustering protein sequence and structure space with infinite Gaussian mixture models. Pacific Symposium on Biocomputing 2004 pp. 399–410 World Scientific Publishing.

Eisen, M. B., Spellman, P. T., Brown, P. O. & Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA,* **95** (25), 14863–14868.

Falkenauer, E. & Marchand, A. (2003) Clustering microarray data with evolutionary algorithms. In *Evolutionary computation in bioinformatics*, (Fogel, G. B. & Corne, D. W., eds), Evolutionary Computation. Morgan Kaufmann 1 edition, pp. 219–230.

Fred, A. & Jain, A. K. (2002) Data clustering using evidence accumulation. In *Proc. of the 16th Int'l Conference on Pattern Recognition* pp. 276–280.

Fred, A. & Jain, A. K. (2003) Robust data clustering. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* pp. 128–133.

Gasch, A. P., Spellman, P. T., Kao, C. M., Carmel-Harel, O., Eisen, M. B., Storz, G., Botstein, D. & Brown, P. O. (2000) Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell,* **11** (12), 4241–4257.

Geller, S. C., Gregg, J. P., Hagerman, P. & Rocke, D. M. (2003) Transformation and normalization of oligonucleotide microarray data. *Bioinformatics,* **19** (14), 1817–1823.

Ghosh, D. & Chinnaiyan, A. M. (2002) Mixture modelling of gene expression data from microarray experiments. *Bioinformatics,* **18** (2), 275–286.

Gibbons, F. D. & Roth, F. P. (2002) Judging the quality of gene expression-based clustering methods using gene annotation. *Genome Res.,* **12** (10), 1574–1581.

Goldstein, D. R., Ghosh, D. & Conlon, E. M. (2002) Statistical issues in the clustering of gene expression data. *Statistica Sinica,* **12** (1), 219–240.

Grotkjær, T. & Nielsen, J. (2004) Enhancing yeast transcription analysis through integration of heterogenous data. *Current Genomics,* **4** (8), 673–686.

Hansen, L. K., Sigurdsson, S., Kolenda, T., Nielsen, F. A., Kjems, U. & Larsen, J. (2000) Modeling text with generalizable Gaussian mixtures. vol. 4, of *International conference on acoustics, speech and signal processing* pp. 3494–3497.

Hastie, T., Tibshirani, R. & Friedman, J. (2001) *The elements of statistical learning - Data mining, inference, and prediction*. Springer Series in Statistics, Springer-Verlag.

Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R., Armour, C. D., Bennett, H. A., Coffey, E., Dai, H., He, Y. D., Kidd, M. J., King, A. M., Meyer, M. R., Slade, D., Lum, P. Y., Stepaniants, S. B., Shoemaker, D. D., Gachotte, D., Chakraburtty, K., Simon, J., Bard, M. & Friend, S. H. (2000) Functional discovery via a compendium of expression profiles. *Cell,* **102** (1), 109–126.

Jones, D. L., Petty, J., Hoyle, D. C., Hayes, A., Ragni, E., Popolo, L., Oliver, S. G. & Stateva, L. I. (2003) Transcriptome profiling of a *Saccharomyces cerevisiae* mutant with a constitutively activated Ras/cAMP pathway. *Physiol. Genomics,* **16** (1), 107–118.

Kaminski, N. & Friedman, N. (2002) Practical approaches to analyzing results of microarray experiments. *Am J Respir. Cell Mol. Biol.,* **27** (2), 125–132.

Kamvar, S. D., Klein, D. & Manning, C. D. (2002) Interpreting and extending classical agglomerative clustering algorithms using a model-based approach. ICML pp. 283–290 Morgan Kaufmann, Sydney, Australia.

MacKay, D. J. C. (2003) *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge.

McLachlan, G. J., Bean, R. W. & Peel, D. (2002) A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics,* **18** (3), 413–422.

Monti, S., Tamayo, P., Mesirov, J. & Golub, T. (2003) Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Mach. Learn.,* **52** (1-2), 91–118.

Pan, W., Lin, J. & Le, C. T. (2002) Model-based cluster analysis of microarray gene-expression data. *Genome Biol.,* **3** (2), 1–9.

Ramoni, M. F., Sebastiani, P. & Kohane, I. S. (2002) Cluster analysis of gene expression dynamics. *Proc. Natl Acad. Sci. U. S. A,* **99** (14), 9121–9126.

Reiner, A., Yekutieli, D. & Benjamini, Y. (2003) Identifying differentially expressed genes using false discovery rate controlling procedures. *Bioinformatics,* **19** (3), 368–375.

Rocke, D. M. & Durbin, B. (2001) A model for measurement error for gene expression arrays. *J Comput. Biol.,* **8** (6), 557–569.

Rocke, D. M. & Durbin, B. (2003) Approximate variance-stabilizing transformations for gene-expression microarray data. *Bioinformatics,* **19** (8), 966–972.

Sharan, R., Maron-Katz, A. & Shamir, R. (2003) CLICK and EXPANDER: a system for clustering and visualizing gene expression data. *Bioinformatics,* **19** (14), 1787–1799.

Smet, F. D., Mathys, J., Marchal, K., Thijs, G., Moor, B. D. & Moreau, Y. (2002) Adaptive quality-based clustering of gene expression profiles. *Bioinformatics,* **18** (5), 735–746.

Strehl, A. & Ghosh, J. (2002) Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research,* **3**, 583–617.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S. & Golub, T. R. (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA,* **96** (6), 2907–2912.

Tavazoie, S., Hughes, J. D., Campbell, M. J., Cho, R. J. & Church, G. M. (1999) Systematic determination of genetic network architecture. *Nat. Genet,* **22** (3), 281–285.

Tusher, V. G., Tibshirani, R. & Chu, G. (2001) Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl Acad. Sci. USA,* **98** (9), 5116–5121.

Venet, D. (2003) MatArray: a Matlab toolbox for microarray data. *Bioinformatics,* **19** (5), 659.

Ward, J. H. (1963) Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.,* **58** (301), 236–244.