Clustering of gene expression profiles: creating initialization-independent clusterings by eliminating unstable genes

Wim De Mulder [1a], Martin Kuiper[2] and René Boel[1]

[1]Systems Research Group, University of Ghent, Ghent, Belgium

[2]Systems Biology Group, NTNU, Trondheim, Norway

Summary

Clustering is an important approach in the analysis of biological data, and often a first step to identify interesting patterns of coexpression in gene expression data. Because of the high complexity and diversity of gene expression data, many genes cannot be easily assigned to a cluster, but even if the dissimilarity of these genes with all other gene groups is large, they will finally be forced to become member of a cluster. In this paper we show how to detect such elements, called unstable elements. We have developed an approach for iterative clustering algorithms in which unstable elements are deleted, making the iterative algorithm less dependent on initial centers. Although the approach is unsupervised, it is less likely that the clusters into which the reduced data set is subdivided contain false positives. This clustering yields a more differentiated approach for biological data, since the cluster analysis is divided into two parts: the pruned data set is divided into highly consistent clusters in an unsupervised way and the removed, unstable elements for which no meaningful cluster exists in unsupervised terms can be given a cluster with the use of biological knowledge and information about the likelihood of cluster membership. We illustrate our framework on both an artificial and real biological data set.

1 Introduction

An important tool in analyzing biological data sets is cluster analysis, the partitioning of a data set into groups based on a specified distance measure so that data points within a group are more similar to each other than to points in different groups. A typical application of cluster analysis is the clustering of gene profiles: ”the large number of genes and the complexity of biological networks greatly increases the challenges of comprehending and interpreting the resulting mass of data, which often consists of millions of measurements. A first step toward addressing this challenge is the use of clustering techniques, which is essential in the data mining process to reveal natural structures and identify interesting patterns in the underlying data” [4].
Iterative clustering algorithms, such as k-means [5], are an important subclass of clustering algorithms. These algorithms start with initial centers and reapply a characteristic deterministic procedure to obtain a set of final centers. The clustering, i.e. the final set of clusters, is then easily obtained from these centers that are considered as representatives of the clusters.
However, iterative algorithms have an undesirable feature: these are local search procedures

[a]Corresponding author: wim.demulder@ugent.be

and it is well known that the performance heavily depends on the initial starting conditions [7], especially the choice of initial centers. This drawback is serious, since it implies that different executions of a clustering algorithm can give completely different clusterings which could suggest that clustering is, to some extent, an arbitrary process.
To remedy this problem, initialization methods have been developed that try to select better than random initial centers. However, these methods are heuristic and their improvements are not overwhelming. In [8] three initialization methods for the EM clustering algorithm [6] were studied, among which random initialization, and it was observed that the performance of the three methods was similar for all examined data sets. In [9] a comparison was made between five initialization methods with the use of three performance measures. The performance of the examined methods depended both on the data set and the chosen performance measure. This limits the general applicability of these initialization methods.
Another popular solution to avoid the arbitrariness encapsulated in iterative algorithms is to generate many clusterings and to select an optimal one according to some cluster validation measure [11]. However, there exists many cluster validation measures and they can give completely different optimal clusterings which introduces again some arbitrariness, in the choice of a validation measure. We call the problem of having to make an arbitrary choice to obtain a clustering from a (large) set of given clusterings, e.g. the arbitrary choice of initial centers or the arbitrary choice of a validation measure, the arbitrariness-problem.
In this paper we demonstrate the use of an iterative clustering algorithm, while avoiding its dependency on initial centers, thus constructing a robust clustering. At the same time this clustering will be well-defined, in the sense that there is no arbitrariness-problem. The basic idea is to reduce the initial data set to a set that is less dependent on initial centers. Our method is explained in section 3 and is illustrated both on an artificial and biological data set in section 4.

We notice that because our goal is to generate a robust clustering for gene expression data, we prefer to use iterative algorithms over hierarchical algorithms. Hierarchical clustering has been noted by statisticians to suffer from lack of robustness [10] and thus is not suitable for our purpose. Indeed, iterative algorithms are dependent on initial centers and thus also vulnerable to noise, but these algorithms are nondeterministic and we will make use of this feature to develop a statistical framework in section 3.1. This framework allows to define the expected clustering, which can be seen as independent of initial centers and thus as robust. Such a statistical framework is not possible for hierarchical clustering, since these algorithms are deterministic. Furthermore, gene expression data sets are typically very large and in this case the construction of a dendrogram is computationally prohibitive [6].

2 Related work

The removal of data elements to obtain a more 'clusterable' subset of the original data set $\{g_1, \ldots, g_n\}$ is also discussed in [12]. In fact, the condition for a data element g_j to be removed is more or less the same for both methods. Given the average clustering $\bar{C}$, to be defined in section 3.1.1, an element g_j is considered to be unstable if $\bar{C}(j,k)$ is around 0.5 for all or many $k \neq j$ (the concept of instability is more rigorously considered in section 3.1). While our definition for instability differs somewhat from the one given in [12], the goal of both methods is essentially to detect such unstable elements and to remove them. Although the basic ideas behind both methods coincide, our discussion differs in three important ways from the one in

[12]. First, while the discussion in [12] is heuristic, we *prove* that under certain conditions the removal of appropriate elements gives a more 'clusterable' data set, see section 3. Secondly, the most important condition to obtain a more clusterable data set is that $\bar{C}(j,k)$ is not altered after the removal of a data element $g_k \notin \{g_j, g_k\}$. This is a non trivial condition, since by eliminating data elements the structure of the reduced data set can be different from the original one, which is overlooked in [12]. This condition is more fully discussed in section 3. Thirdly, the final clustering in [12] is obtained by applying a hierarchical clustering algorithm. We show in section 3.2 how the final clustering can be obtained in a more intuitive way, without the need for an extra clustering step.
We mention that other criteria than instability can be used to select elements for removal. For example, the silhouette width [11] of an element indicates the confidence with which this element belongs to the produced clusters. Consequently, elements with a low silhouette width can be selected for removal. However, as already indicated, we show that the use of instability makes it possible to state theorems concerning the 'clusterability' of a reduced data set.

3 Methods

3.1 Cluster stability variance

3.1.1 M-set and expected clustering

Given is a data set $D = \{g_1, \ldots, g_n\}$ that is clustered by a given iterative clustering algorithm. A clustering can be represented as a matrix C with elements $C(j,k), j = 1...n, k = 1...n$:

$$C(j,k) = 1 \quad \text{if } g_j \text{ and } g_k \text{ are placed in different clusters} \tag{1}$$
$$= 0 \quad \text{if } g_j \text{ and } g_k \text{ are placed in the same cluster} \tag{2}$$

Given N clusterings C_i, generated with randomly chosen initial centers and possibly by different iterative algorithms, we call the set $M = \{C_1, ..., C_N\}$ an M-set. The representation of a clustering as a matrix gives the advantage that an average clustering $\bar{C}$ can be defined, given an M-set $\{C_1, \ldots, C_N\}$, as follows: $\bar{C}(j,k) = \frac{1}{N}\sum_{i=1}^{N} C_i(j,k)$. We can then also define the expected clustering $E[C]$ as the matrix with elements $E[C](j,k) = E[C(j,k)]$, provided that the expected values $E[C(j,k)]$ exist.
The importance of the expected clustering is that it can be seen as independent of initial centers (and of clustering algorithm), since it is the uniquely defined probability-weighted sum over all possible clusterings generated by the given iterative algorithm or algorithms. In practice this expected clustering is approximated by the above defined average clustering $\bar{C}$ which is considered to be still much less dependent on initial centers than a particular clustering, especially if the initial centers for the generation of the C_i are randomly selected and if N is large enough. This suggests that $\bar{C}$ can be chosen as the clustering that is highly independent of initial centers. This suggestion is not correct, because $\bar{C}$ does not necessarily represent a clustering, since it is not necessarily true that $\bar{C}(j,k) = 1$ or 0, $\forall j,k$. The goal is now to find a clustering that combines the main characteristic of $\bar{C}$, being highly independent of initial centers, and that of the corresponding M-set, containing clusterings that satisfy (1)-(2).
Note that $\bar{C}(j,k)$ denotes the fraction of clusterings whereby g_j and g_k are placed in different clusters. Large values of $\bar{C}(j,k)$ indicate that g_j and g_k end up in different clusters for most

choices of initial centers, and small values indicate the opposite. Values around 0.5 indicate uncertainty about whether g_j and g_k belong to different clusters or to the same cluster.
Since both small and large values of $\bar{C}(j,k)$ indicate a stable relationship between g_j and g_k in the sense that under these conditions g_j and g_k clearly belong to the same cluster or clearly belong to different clusters, a 'stability function' σ can be introduced:

$$\begin{aligned} \sigma(a) &= 1-a \quad 0.5 \leq a \leq 1 \\ &= a \qquad 0 \leq a < 0.5 \end{aligned}$$

for $a \in [0,1]$. Thus, the lower $\sigma(\bar{C}(j,k))$, the more stable the relationship is between g_j and g_k in the sense described above, and vice versa.
As a next step a measure is constructed that summarizes the stability of *all* the relationships between the given data elements. This can be done by summing $\sigma(\bar{C}(j,k))$ over all the elements. We thus define the instability of an M-set $M = \{C_1, \ldots, C_N\}$ as

$$\mu(M) = \frac{2}{n(n-1)} \sum_{j=1}^{n-1} \sum_{j<k\leq n} \sigma(\bar{C}(j,k)) \tag{3}$$

The larger $\mu(M)$, the larger the instability of M, which intuitively means that there is more uncertainty about which elements belong together.

3.1.2 Cluster stability variance

In section 3.1.1 it is shown that a clustering can be interpreted as a random variable, where randomness arises from the random choice of initial centers. This probabilistic view allows us to define the variance of a random clustering as $E[d(C, E[C])^2]$, where $d(C, E[C])$ denotes the 'squared distance' from C to $E[C]$ which we define as:

$$d(C, E[C]) = \frac{2}{n(n-1)} \sum_{j=1}^{n-1} \sum_{j<k\leq n} |C(j,k) - E[C](j,k)| \tag{4}$$

For finite N we use the following as an approximation of the variance, given an M-set M:

$$CSV(M) = \frac{1}{N-1} \sum_{i=1}^{N} d(C_i, \bar{C})^2 \tag{5}$$

where CSV is an abbreviation for what we call the 'cluster stability variance' associated with the M-set $\{C_1, \ldots, C_N\}$. Since differences between clusterings in an M-set arise from different initial centers, the CSV can only be nonzero because of the dependence on initial centers. Thus the CSV is a measure for the dependence of an M-set on initial centers. Reducing the dependency of clusterings in a given M-set thus amounts to reducing the CSV. This can also be seen from a different point of view: from formula (5) it follows that the CSV can be interpreted as the distance from an M-set to the corresponding average clustering, or thus as an approximation of the distance from a given M-set to the expected clustering, which is completely independent of initial centers. In this respect, the reduction of the CSV equals the reduction of the distance of the M-set to the expected clustering, which amounts to making the clusterings contained in the M-set more independent of initial centers.
However, there is no clue as to how to reduce this CSV. We take a detour by going back to the above defined concept of instability.

3.1.3 Reducing the instability of an M-set

We now define the instability of a given data element g_k. As for the definition of the instability of an M-set (3) we use the values $\sigma(\bar{C}(j,k))$, but we restrict the summation to those elements where k is involved:

$$\mu(g_k) \;=\; \frac{1}{n-1}\Big(\sum_{j=1}^{k-1}\sigma(\bar{C}(j,k)) + \sum_{j=k+1}^{n}\sigma(\bar{C}(k,j))\Big)$$

Intuitively we expect that the more unstable the data elements are, the more unstable the associated M-set. This is now proven.

Theorem 1 1. $\frac{1}{n}\sum_{k=1}^{n}\mu(g_k) = \mu(M)$

Proof.

$$\begin{aligned}\sum_{k=1}^{n}\mu(g_k) &= \sum_{k=1}^{n}\frac{1}{n-1}\Big(\sum_{j=1}^{k-1}\sigma(\bar{C}(j,k)) + \sum_{j=k+1}^{n}\sigma(\bar{C}(k,j))\Big)\\ &= n\frac{2}{n(n-1)}\sum_{j=1}^{n}\sum_{k=1}^{n}\sigma(\bar{C}(j,k)) \;=\; n\,\mu(M)\end{aligned}$$

□

Lemma 1 1.

$$\sigma(\bar{C}(j,k)) \;=\; \min(|C_i(j,k)-\bar{C}(j,k)|, 1-|C_i(j,k)-\bar{C}(j,k)|)$$

Proof. Suppose first that $\bar{C}(j,k) > 0.5$. This implies: $\sigma(\bar{C}(j,k)) = 1-\bar{C}(j,k)$.
Case 1: $\min(|C_i(j,k)-\bar{C}(j,k)|, 1-|C_i(j,k)-\bar{C}(j,k)|) = |C_i(j,k)-\bar{C}(j,k)|$. Then we have

$$\begin{aligned}|C_i(j,k)-\bar{C}(j,k)| &\leq 1-|C_i(j,k)-\bar{C}(j,k)|\\ \Rightarrow |C_i(j,k)-\bar{C}(j,k)| &\leq 0.5\end{aligned}$$

This implies that $C_i(j,k) = 1$ and so we have: $\sigma(\bar{C}(j,k)) = 1-\bar{C}(j,k) = C_i(j,k)-\bar{C}(j,k) = |C_i(j,k)-\bar{C}(j,k)|$.
Case 2: $\min(|C_i(j,k)-\bar{C}(j,k)|, 1-|C_i(j,k)-\bar{C}(j,k)|) = 1-|C_i(j,k)-\bar{C}(j,k)|$. This is proved in a similar manner as case 1. In this case we have $C_i(j,k) = 0$ and $\sigma(\bar{C}(j,k)) = 1-|C_i(j,k)-\bar{C}(j,k)|$. The part where $\bar{C}(j,k) \leq 0.5$ is checked similarly. □

We now present a relationship between $CSV(M)$ and $\mu(M)$.

Theorem 2 1. $CSV(M) \;\leq\; \frac{N}{N-1}\,\mu(M)$

Proof. We put the elements $C_i(j,k)$ for given j and k, and $i = 1,...,N$ in two sets A_1 and A_2 as follows: $A_0 = \{C_i(j,k)\,|\,C_i(j,k) = 0\}$ and $A_1 = \{C_i(j,k)\,|\,C_i(j,k) = 1\}$. Note

that $|A_0| = N(1 - \bar{C}(j,k))$ and $|A_1| = N\bar{C}(j,k)$, where $|A_0|$ and $|A_1|$ denote the number of elements of A_0 resp. A_1. From the previous lemma we know that

$$N\sigma(\bar{C}(j,k)) = \sum_{i=1}^{N} \min(|C_i(j,k) - \bar{C}(j,k)|, 1 - |C_i(j,k) - \bar{C}(j,k)|) \quad (6)$$

Suppose now that $\bar{C}(j,k) > 0.5$. From the proof of the above lemma it follows that

$$C_i(j,k) = 1 \Rightarrow \sigma(\bar{C}(j,k)) = |C_i(j,k) - \bar{C}(j,k)| \quad (7)$$
$$C_i(j,k) = 0 \Rightarrow \sigma(\bar{C}(j,k)) = 1 - |C_i(j,k) - \bar{C}(j,k)| \quad (8)$$

Denote

$$|C_i(j,k) - \bar{C}(j,k)|_0 = |C_i(j,k) - \bar{C}(j,k)| = \bar{C}(j,k) \quad \text{if } C_i(j,k) \in A_0 \quad (9)$$
$$|C_i(j,k) - \bar{C}(j,k)|_1 = |C_i(j,k) - \bar{C}(j,k)| = 1 - \bar{C}(j,k) \quad \text{if } C_i(j,k) \in A_1 \quad (10)$$

From (6), (7) and (8) it then follows that

$$N\sigma(\bar{C}(j,k)) = |A_0|(1 - |C_i(j,k) - \bar{C}(j,k)|_0) + |A_1|\,|C_i(j,k) - \bar{C}(j,k)|_1$$

and because of (9)-(10) this is equivalent to

$$N\sigma(\bar{C}(j,k)) = |A_0|(1 - \bar{C}(j,k)) + |A_1|\,(1 - \bar{C}(j,k)) \quad (11)$$
$$= N(1 - \bar{C}(j,k)) \quad (12)$$

We have also the following

$$\sum_{i=1}^{N} |C_i(j,k) - \bar{C}(j,k)|^2 = \sum_{i=1}^{|A_0|} |C_i(j,k) - \bar{C}(j,k)|_0^2 + \sum_{|A_0|+1}^{N} |C_i(j,k) - \bar{C}(j,k)|_1^2 \quad (13)$$
$$= |A_0|\,|C_i(j,k) - \bar{C}(j,k)|_0^2 + |A_1|\,|C_i(j,k) - \bar{C}(j,k)|_1^2 \quad (14)$$
$$= N\bar{C}(j,k)(1 - \bar{C}(j,k)) \quad (15)$$

after suitable substitutions. Subtracting (12) from (15) gives

$$\sum_{i=1}^{N} |C_i(j,k) - \bar{C}(j,k)|^2 - N\sigma(\bar{C}(j,k)) = -N(1 - \bar{C})^2 \quad (16)$$

$$\Rightarrow \quad \frac{1}{N}\sum_{i=1}^{N} |C_i(j,k) - \bar{C}(j,k)|^2 = \sigma(\bar{C}(j,k)) - (1 - \bar{C})^2 \leq \sigma(\bar{C}(j,k)) \quad (17)$$

Given that (see (4))

$$\frac{1}{N}\sum_{i=1}^{N} d(C_i, \bar{C}) = \frac{1}{N}\frac{2}{n(n-1)}\sum_{i=1}^{N}\sum_{k=1}^{n-1}\sum_{k<j\leq n} |C_i(j,k) - \bar{C}(j,k)|$$

(17) becomes

$$\frac{1}{N}\sum_{i=1}^{N} d(C_i, \bar{C})^2 \leq \frac{2}{n(n-1)}\sum_{k=1}^{n-1}\sum_{k<j\leq n} \sigma(\bar{C}(j,k)) = \mu(M) \quad (18)$$

It can be checked that the case $\bar{C}(j,k) \leq 0.5$ gives the same result. The statement about the CSV is now easily demonstrated from (18). □

The importance of the previous theorem is that it gives the missing clue of how to reduce the CSV: the CSV can possibly be lowered by reducing the instability. The next question is thus how $\mu(M)$ can be lowered. It is now shown that this can be done by eliminating unstable elements. But first we introduce a new notation. Notice that $E[C]$ and thus $\bar{C}$ depends on the chosen clustering algorithm(s) which we keep fixed during the elimination of elements, but also depends on the data set D which changes while eliminating data elements. Since the following two theorems deal with deleting elements, we make this dependency explicit by using the notation $\bar{C}_D$ instead of $\bar{C}$ for the following theorems. Furthermore we will need to denote submatrices of $\bar{C}$, which we define as $\bar{C}(A) = \{\bar{C}(j,k) \mid d_j, d_k \in A\}$ and with $A \subseteq D$. Likewise the instability of a data element depends on the current data set. Thus we use the notation $\mu_D(g_l)$ to denote the instability of g_l regarding data set D and $\mu_D(M)$ to denote the instability of M regarding D.

Theorem 3 1.

$$\mu_D(g_l) = \max_{1 \le k \le n} \mu_D(g_k), \bar{C}_D(D \setminus \{g_l\}) = \bar{C}_{D\setminus\{g_l\}}(D \setminus \{g_l\}) \Rightarrow \mu_{D\setminus\{g_l\}}(M) \le \mu_D(M)$$

Proof.

$$\begin{aligned}
\mu_{D\setminus\{g_l\}}(M) - \mu_D(M) &= \frac{1}{n-1} \sum_{k=1,k\neq l}^{n} \mu_{D\setminus\{g_l\}}(g_k) - \frac{1}{n} \sum_{k=1}^{n} \mu_D(g_k) \quad \text{(by theorem 1)} \\
&= \frac{1}{n-1} \sum_{k=1,k\neq l}^{n} \mu_D(g_k) - \frac{1}{n} \sum_{k=1}^{n} \mu_D(g_k) \\
&= \frac{\sum_{k=1,k\neq l}^{n} \mu_D(g_k) - (n-1)\mu_D(g_l)}{n(n-1)} \\
&\le 0
\end{aligned}$$

where the second line follows from the first by the assumption $\bar{C}_D(D \setminus \{g_l\}) = \bar{C}_{D\setminus\{g_l\}}(D \setminus \{g_l\})$, and the fourth line from the third because of the assumption $\mu_D(g_l) = \max_{1 \le k \le n} \mu_D(g_k)$. □

Thus eliminating the most unstable element will reduce the instability of M or preserve the instability in the worst case, under the assumption that the data structure, as seen by the considered clustering algorithm(s), does not change, i.e. $\bar{C}_D(D \setminus \{g_l\}) = \bar{C}_{D\setminus\{g_l\}}(D \setminus \{g_l\})$. Furthermore deleting the most unstable element will lead to the greatest decrease in instability, as expressed by the following theorem.

Theorem 4 1.

$$\begin{aligned}
&\mu_D(g_l) \ge \mu_D(g_m), \bar{C}_D(D \setminus \{g_l\}) = \bar{C}_{D\setminus\{g_l\}}(D \setminus \{g_l\}), \\
&\bar{C}_D(D \setminus \{g_m\}) = \bar{C}_{D\setminus\{g_m\}}(D \setminus \{g_m\}) \Rightarrow \mu_{D\setminus\{g_l\}}(M) \le \mu_{D\setminus\{g_m\}}(M)
\end{aligned}$$

Proof. The proof is completely analogous to the proof of theorem 3. □

3.2 A highly initialization-independent clustering

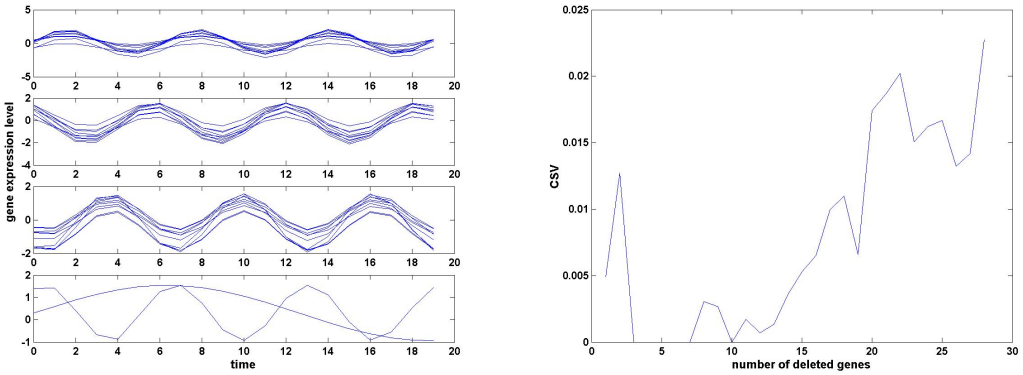
In section 3.1.1 it was stated that we want to find a clustering that is highly independent of initial centers, a characteristic possessed by $\bar{C}$, but at the same time constituting a clustering according to definitions (1) - (2). In section 3.1.2 it was stated that the clusterings of a given M-set can be given this characteristic by reducing the CSV, and from section 3.1.3 it follows that removing unstable elements is a very good approach to accomplish this. From theorems 3 and 4 one would conclude that the more elements are removed, starting with the most unstable element, then the second most unstable one, etc, the lower the instability of the M-set and thus the lower the CSV potentially becomes by theorem 2. Thus all elements except one should be removed to obtain the maximally independent clustering, which is clearly undesired.
However, notice that this conclusion is not true, since the above theorems only hold if the data structure is not changed by removing data elements. By this we mean that the same elements are clustered together before and after the removal of some data element. This consideration gives two main cases: 1. the assumption of unchanging structure is more or less true and thus there is no well-defined highly initialization-independent clustering, 2. the assumption is not valid and after the removal of a number of unstable elements the CSV reaches a minimum. In this case the optimal data set is the given data set after the elimination of these unstable elements and the highly initialization-independent clustering is then defined as the clustering in the given M-set that is closest to $\bar{C}$, where distance is measured according to (4).
Three remarks are in place. First, we let an M-set consists of clusterings generated with a fixed number of initial centers. This ensures that the clusterings in a given M-set share an important characteristic, namely that each clustering has the same number of clusters. This gives also the possibility to generate several M-sets, with different numbers of initial centers. Some will possibly fall in case 1, some possibly in case 2. The advantage then is that for those that fall in case 2, we immediately know unambiguously the number of clusters.
Second, it could be argued that the highly initialization-dependent clustering is not necessarily 'better', according to some validation measure, than the other clusterings in the M-set. However, the original objective was that by eliminating unstable elements, and thus reducing the CSV, the clusterings of the reduced data set are only marginally dependent on initial centers (on the condition that the CSV can be made small enough) and thus the differences between the clusterings in the M-set are small. If we accept that these small differences are not critical, a validation measure is not necessary, except for the case that there are several M-sets that fall in case 2 and we have to choose from their highly initialization-independent clusterings.
Third, a removed gene g_j can be given a membership degree $m(g_j, c)$ in cluster c in a natural way, as follows: $m(g_j, c) = \sum_{g_k \in c} \bar{C}(j,k)/|c|$.

The fact that the removed unstable elements are not part of a cluster, opens the door for investigators to subsequently enhance the clustering process. The removed elements can be placed in a suitable cluster on the basis of biological knowledge, while the membership degrees can be used as additional unsupervised knowledge.

4 Results

4.1 Artificial data set

We created a simple artificial data set that mimics periodic gene times series data, by starting from the most simple period function $f(x) = \sin(x)$ and adding some noise to it. Three clusters of genes are then created as follows. First, the three phase shifts $0, 2\pi/3$ and $4\pi/3$ are used to characterize each cluster. Second, for each cluster 10 gene time series are defined by selecting 20 time points from the corresponding sine function. For this selection we introduce two levels of randomness: one on the level of the gene and one on the level of the time points. For each gene g a random number $R_1(g) \sim \mathcal{N}(0, \sigma_1^2)$ is generated. For each gene g *and* each pth time point, $p = 1, \ldots, 20$, a second Gaussian number $R_2(g, p) \sim \mathcal{N}(p - 1, \sigma_2^2)$ is generated. The value of the pth time point, $t_p(g)$, is then defined as $t_p(g) = R_1(g) + R_2(g, p) + P(g)$, where $P(g)$ denotes the phase shift of g. Whereas $R_1(g)$ is fixed for each time point, the value of $R_2(g, p)$ is dependent on the time point. We choose $\sigma_1 = 0.05$ and $\sigma_2 = 0.01$.
Finally, two extra elements are introduced that do not clearly belong to a cluster. The first element is created in the same way as the above 30 genes, but with a phase shift of $\pi/3$. The second element is given a time shift of 0 and a large period of 8π.
Figure 1(a) shows the generated genes according to the cluster to which they belong. The two extra elements have been placed in an imaginary fourth cluster. K-means with the correlation distance as distance measure was applied with $N = 10$ and 3 centers. The resulting CSV is shown in Figure 1(b). A first minimum is at 0 after the two extra elements were deleted. Thus, the method described above allows to detect these elements that do not belong to any of the predefined clusters. If this would be a real biological data set, it would be up to the biologist to decide to which cluster, if any, these 2 elements belong on the basis of biological knowledge.



(a) 3 well separated clusters and 2 unstable elements (b) CSV

Figure 1: Artificial data set

4.2 Real biological data set

With the use of the above described method we analyzed the fission yeast cell cycle microarray data set discussed in [1], which concerns yeast cells synchronized by elutriation, and subsequently grown and sampled intermittently to obtain time series growth data. In three independent experimental time series, each gene was measured at 20 time points.
From this large data set the top 500 genes reported in [2] were selected for further consideration. This data set was then further reduced as follows. First, for each gene g_i the fraction of non-missing time points was calculated, denoted as $n_1(g_i)$. For example, if g_i had 45 non-missing values, then $n_1(g_i) = 45/60 = 0.75$. Obviously, we wanted to proceed further with only those genes for which n_1 was high enough. Now, denote the highest value of n_1 over all genes as a_1, the second highest value as a_2, etc. Let $S_i = \{g_j \in G \,|\, n_1(g_j) \geq a_i\}$, where G denotes the set of 500 genes. Our purpose was to select a set of genes S_i for which the corresponding a_i was as large as possible, since this implied that the number of missing values was small. At the same time we wanted $|S_i|$, the number of elements in S_i, as large as possible so that the number of genes that was not considered further would be limited. However, the fact that the sequence of a_i is decreasing and the sequence of $|S_i|$ is increasing, gave two conflicting requirements. A compromise was found as follows. Let $c_i = a_i + |S_i|/|G|$. The optimal S_i is then defined as the one for which the corresponding c_i is maximal, which was found to consist of 338 genes.
Missing values were filled in using k-nearest neighbor [3], and finally for each of the 338 genes the average was taken over the three experimental repeats to give the final data set.

We then generated 4 M-sets with $N = 50$ with 3, 4, 5 and 6 clusters. We chose a small number of clusters, since in [1] this yeast data set was also analyzed and 4 clusters were found. However, notice that because of different preprocessing methods our analyzed data set was not completely identical. For each M-set unstable elements were sequentially eliminated starting with the most unstable one, and after each elimination the CSV was calculated again. The results are shown in Figure 2. To make the decision whether an M-set belongs to case 1 or 2 (see section 3.2) we plotted also a second-degree polynomial interpolation. It is clear that the M-sets for 4, 5 and 6 clusters belong to case 1 and thus that there is no well-defined maximally initialization-independent clustering for which the data set can be divided in 4, 5 or 6 clusters, since the CSV decreases steadily with the elimination of genes. On the other hand, there is a clear minimum in the case of 3 clusters, where the minimum corresponds with the elimination of 84 (unstable) elements. We then found the desired highly initialization-independent clustering as the one that was closest to $\bar{C}$, as explained in section 3.2. The distance of the closest clustering to the average clustering was found to be around 0.06, thus assuring that our final clustering is highly independent of initial centers. Furthermore, thirty-seven clusterings of the 50 clusterings in the generated M-set were exactly equal to this clustering, thus further confirming that we have substantially reduced the dependence on initial centers.
Notice that it is not stated that the given data set cannot be meaningfully divided into 4, 5 or 6 clusters, but only that the described method does not tell how the data set should be optimally subdivided in a part that is clustered in an unsupervised way and a part for which previously gained biological knowledge is used to enhance the unsupervised clustering. A solution is to do the clustering completely unsupervised, as traditional practice, or to predefine a threshold for the CSV and to eliminate genes as long as this threshold is not reached.

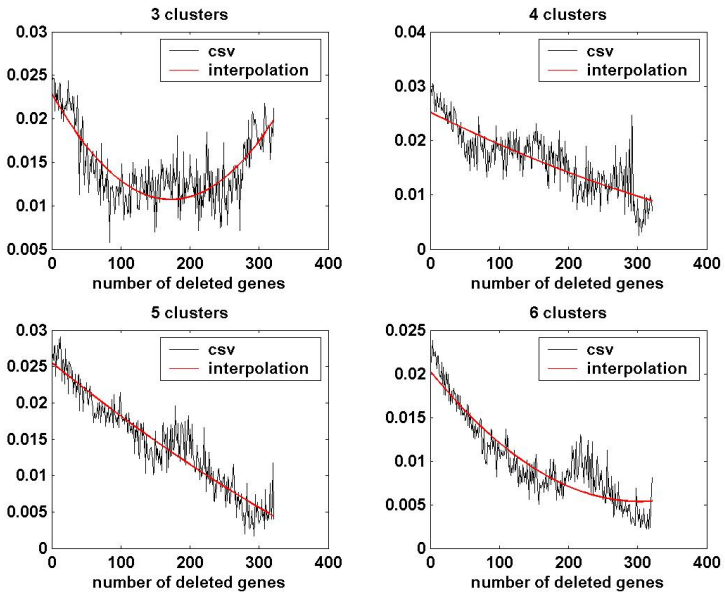


Figure 2: CSV of fission yeast data set

5 Discussion and future research

We presented a method that detects and removes unstable elements from a given gene expression data set. The data characteristics of these genes are too different from those of the other genes to reliably assign them to a cluster in an unsupervised way. The remaining, stable genes can be well-divided into clusters and if the the data set is such that distance-based similarity corresponds, to some extent, to similarity in terms of biological functional characteristics, the subsequent biological analysis performed by the biologist is greatly facilitated. After the biological characteristics of these clusters have become clear, the unstable elements can be assigned to one of the clusters in a supervised way, integrating supervised and unsupervised analysis in a natural way.

We plan to apply the described method to several real biological data sets. Data sets that will be used for this purpose are the fission yeast data from section 4.2, data sets from the Many Microbe Microarrays Database [13] and the S. cerevisiae data set described in [14].

Although classical hierarchical algorithms are not appropriate to generate a robust clustering of gene expression data (see section 1), a more recent technique, called SOTA [15], combines both hierarchical and SOM clustering, allowing a visual representation of the clusters and being rather insensitive to noise. An interesting direction for future research is to compare the proposed approach with SOTA in terms of robustness, time complexity, and ease of evaluation of the clustering for the biologist.

References

[1] G. Rustici et al., *Periodic gene expression program of the fission yeast cell cycle*, Nature Genetics, vol. 36(8), pp. 809-817, 2004.

[2] S. Marguerat et al., *The more the merrier: comparative analysis of microarray studies on cell cycle-regulated genes in fission yeast*, Yeast, vol. 23(4), pp. 261-277, 2006.

[3] A. Zhang, *Advanced analysis of gene expression microarray data*, World Scientific Publishing Company, 2006.

[4] D. Jiang, C. Tang and A. Zhang, *Cluster analysis for gene expression data: a survey*, IEEE Transactions on Knowledge and Data Engineering, vol. 16(11), pp. 1370-1386, 2004.

[5] A.K. Jain and R.C. Dubes, *Algorithms for clustering data*, Prentice Hall, 1988.

[6] A.K. Jain, M.N. Murty and P.J. Flynn, *Data clustering: a review*, ACM Computing Surveys, vol. 31(3), pp. 264-323, 1999.

[7] A. Likas, N. Vlassis and J.J. Verbeek, *The global k-means clustering algorithm*, Pattern Recognition, vol. 36(2), pp. 451-461, 2003.

[8] M. Meila and D. Heckermann, *An experimental comparison of model-based clustering methods*, Machine Learning, vol. 42(1/2), pp. 9-29, 2001.

[9] J. He et al., *Initialization of cluster refinement algorithms: a review and comparative study*, Proceedings of International Joint Conference on Neural Networks, vol. 1, pp. 297-302, 2004.

[10] P. Tamayo et al., *Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation*, Proceedings of the National Academy of Sciences, vol. 96, pp. 29072912, 1999.

[11] M. Halkidi, Y. Batistakis and M. Vazirgiannis, *On clustering validation techniques*, Journal of Intelligent Systems, vol. 17(2/3), pp. 107-145, 2001.

[12] T.T. Nguyen, R.S. Nowakowski and I. Androulakis, *Unsupervised selection of highly coexpressed and noncoexpressed genes using a consensus approach*, Journal of Integrative Biology, vol. 13, pp. 219-237, 2009.

[13] J.J. Faith et al., *Many microbe microarrays database: uniformly normalized Affymetrix compendia with structured experimental metadata*, Nucleic Acids Research, vol. 36, pp. 866-870, 2008.

[14] T.R. Hughes et al., *Functional discovery via a compendium of expression profiles*, Cell, vol. 102, pp. 109-126, 2000.

[15] L. Yin, C. Huang and J. Ni, *Clustering of gene expression data: performance and similarity analysis*, BMC Bioinformatics, vol. 7(Suppl 4):S19, 2006.