Adaptive quality-based clustering of gene expression profiles

Frank De Smet*, Janick Mathys, Kathleen Marchal, Gert Thijs, Bart De Moor, Yves Moreau

ESAT-SISTA/COSIC/DocArch, K.U.Leuven, Kasteelpark Arenberg 10, 3001 Leuven-Heverlee, Belgium

Running head: Adaptive quality-based clustering

Keywords: clustering, expression profiles, quality-based, adaptive, microarray

* To whom correspondence should be addressed

Abstract

Motivation: Microarray experiments generate a considerable amount of data, which analysed properly help us gain a huge amount of biologically relevant information about the global cellular behaviour. Clustering (grouping genes with similar expression profiles) is one of the first steps in data analysis of high-throughput expression measurements. A number of clustering algorithms have proved useful to make sense of such data. These classical algorithms, though useful, suffer from several drawbacks (e.g., they require the predefinition of arbitrary parameters like the number of clusters; they force every gene into a cluster despite a low correlation with other cluster members). In the following we describe a novel adaptive quality-based clustering algorithm that tackles some of these drawbacks.

Results: We propose a heuristic iterative two-step algorithm: First, we find in the high-dimensional representation of the data a sphere where the 'density' of expression profiles is locally maximal (based on a preliminary estimate of the radius of the cluster – quality-based approach). In a second step, we derive an optimal radius of the cluster (adaptive approach) so that only the significantly coexpressed genes are included in the cluster. This estimation is achieved by fitting a model to the data using an EM-algorithm. By inferring the radius from the data itself, the biologist is freed from finding an optimal value for this radius by trial-and-error. The computational complexity of this method is approximately linear in the number of gene expression profiles in the data set. Finally, our method is successfully validated using existing data sets.

Availability: http://www.esat.kuleuven.ac.be/~thijs/Work/Clustering.html

Contact: frank.desmet@esat.kuleuven.ac.be

Supplementary information: http://www.esat.kuleuven.ac.be/~fdesmet/paper/adaptpaper.html

Introduction

A variety of techniques (Self-Organizing Maps (Tamayo et al., 1999), hierarchical clustering (Carr et al., 1997; Eisen et al., 1998), Self-Organising Tree Algorithm (Herrero et al., 2001), K-means (Tavazoie et al., 1999; Tou and Gonzalez, 1979), simulated annealing (Lukashin and Fuchs, 2001), Principal Component Analysis (Quackenbush, 2001), MultiDimensional Scaling (Bittner et al., 2000), Cluster

Affinity Search Technique (Ben-Dor et al., 1999), …) has been implemented and successfully been used to analyse or cluster high-dimensional microarray data (DeRisi et al., 1997; Lander, 1999; Schena et al., 1995). One of the objectives of these methods is to detect groups of genes that exhibit a highly similar expression profile (here defined as coexpressed). Since gene expression profiles are encoded in real vectors (whose elements are the different measurements of the expression levels of a specific gene), these algorithms intend to group gene expression vectors that are sufficiently close to each other (according to a certain distance or similarity measure). Most clustering algorithms originate from non-biologically related research fields. Therefore, although useful, the original implementations suffer from some drawbacks as has been highlighted by Sherlock (2000). These deficiencies can be summarised as follows. Firstly, algorithms such as K-means and Self-Organizing Maps require the predefinition of the number of clusters (parameter of the algorithm). When clustering gene expression profiles, the number of clusters present in the data is usually unknown. Changing this parameter usually affects the final clustering result considerably. Clustering, using for example K-means, therefore involves extensive parameter fine-tuning to detect the optimal clustering and the choice of the final parameter setting remains somehow arbitrary (e.g., based on visual inspection of the clusters). When using hierarchical clustering, the number of clusters is determined by cutting the tree structure at a certain level. The resulting cluster structure is therefore highly influenced by the choice of this level, which in turn is rather arbitrary. Secondly, the idea of forcing each gene of the data set into a cluster is a significant drawback of these implementations. If genes are, despite a rather low correlation with other cluster members, forced to end up in one of the clusters, the average profile of this cluster is corrupted and the composition of the cluster becomes less suitable for further analyses (such as motif finding or functional annotation (Roth et al., 1998; van Helden et al., 2000)).

Much effort is currently being done to adapt clustering algorithms towards the specific needs of biological problems. In this context the ideas of quality-based clustering (Heyer et al., 1999) and gene shaving (Hastie et al., 2000) were developed (gene shaving also uses a quality measure to select the cluster size). Heyer et al. (1999) proposed an algorithm (QT_Clust) that tries to identify clusters that have a certain quality (representing the minimal degree of coexpression needed – see below for the exact definition used in this paper) and where every cluster contains a maximal number of points. Genes not exhibiting this minimal degree of coexpression with any of the clusters are excluded from further analysis. A problem with

the quality-based approach of Heyer et al. (1999), however, is that this quality is a user-defined parameter that is hard to estimate (it is hard to find a good trade-off or optimal value: setting the quality too strictly will exclude a considerable number of coexpressed genes, setting it too loose will include too many genes that are not really coexpressed). Moreover, it should be noted that the optimal value for this quality is, in general, different for each cluster and data set dependent.

In this paper, we describe an adaptive quality-based clustering method starting from the principles described by Heyer et al. (1999) (quality-based approach; locating clusters, with a certain quality, in a volume where the density of points is maximal). The algorithm described below is in essence a heuristic, two-step approach that defines the clusters sequentially (the number of clusters is not known in advance, so it is not a parameter of the algorithm). The first step locates a cluster (quality-based approach) and the second step derives the quality of this cluster from the data (adaptive approach). The performance of the algorithm is tested on a real microarray data set and the result is compared with an already published analysis (K-means) using the same data. Finally, we make a theoretical comparison between our algorithm and the algorithm of Heyer et al. (1999).

Methods

Normalisation

It is common practice to normalise gene expression vectors before cluster analysis. In this paper, we normalise the expression profiles so that their mean is zero and their variance is one before proceeding with the actual cluster algorithm. If $g_i(g_i^1, g_i^2, \ldots, g_i^E)$ is a normalised expression vector, this means that

$$\mu_i = \frac{1}{E}\sum_{j=1}^{E} g_i^{\,j} = 0, \quad (1)$$

$$\sigma_i = \sqrt{\frac{1}{E-1}\sum_{j=1}^{E}(g_i^{\,j} - \mu_i)^2} = 1. \quad (2)$$

Normalised expression profiles or vectors therefore are located in an E-dimensional space on the intersection of a hyperplane (Equation (1)) and a hypersphere with a radius equal to $\sqrt{(E-1)}$ (Equation (2)).

Quality R of a cluster

The definition used in this paper for the quality R of a cluster, is as follows: In a collection of gene expression profiles G={g_i, i=1,…,N}, a cluster K with center C_K and quality R_K (also called radius of cluster K), will only contain the profiles satisfying the following property:

$$\|g_i - C_K\|_2 < R_K. \quad (3)$$

Equation (3) means that cluster K only contains gene expression profiles with a minimum degree of coexpression (represented by the quality guarantee R_K). The norm or distance measure we use in this paper is the 2-norm or Euclidean distance.

Algorithm and implementation

Global algorithm

The global cluster algorithm (Adap_Cluster) is, as mentioned previously, a heuristic iterative two-step approach where the basic steps are as described below. In this implementation we use two user-defined parameters (MIN_NR_GENES and S – the values between brackets are default values), several internal tuning parameters that have a fixed value (the user is not allowed to change these values) and the data set itself (G).

```
Adap_Cluster (G ={g_i, i=1,…,N}, MIN_NR_GENES <2>, S <0.95>)

ACCUR_RAD = 0.1          /* Set internal tuning parameter */

Initialise R_K _PRELIM     /* Radius estimate initialisation */

WHILE Stop criterion NOT TRUE

        C_K = locate_cluster_center (G, R_K _PRELIM)          /* Localisation of a cluster center – Step 1*/

        R_K = recalculate_radius (G, C_K, R_K _PRELIM, S)     /* Re-estimation of radius – Step 2 */

        IF ( | R_K - R_K _PRELIM | / R_K _PRELIM) < ACCUR_RAD   /* Check accuracy of radius estimation */

                CLUSTER = {g_i ∈ G | ||g_i - C_K || < R_K }

                G = G \ CLUSTER          /* Remove cluster from data set G */

                IF #CLUSTER >= MIN_NR_GENES        /* Valid cluster ? */

                        Output CLUSTER

                END IF

        END IF

        R_K _PRELIM = R_K          /* Update radius estimate */

END WHILE
```

During each iteration, this algorithm first finds a cluster center (C_K) using a preliminary estimate (R_K _PRELIM) of the radius or quality of the cluster (Step 1). When the cluster center has been located, the algorithm determines a new estimate for the radius (R_K) of the cluster (Step 2). Now there are two possibilities:

1. If this new estimate is approximately equal to the preliminary estimate (e.g., within 10% - ACCUR_RAD), the set of genes defined by the cluster center and the new estimate of the radius is removed from the data set G. Furthermore, if the number of genes in this set is equal or larger than a predefined value (MIN_NR_GENES – user-defined; default 2), this set is a valid cluster. The preliminary estimate of the radius to be used in Step 1 of the next iteration (for the next cluster) is updated with the new estimate of the radius calculated in Step 2 of the current iteration (in most cases, the best preliminary estimate for the radius of the next cluster seems to be the radius of the previous cluster).
2. If the new estimate of the radius is substantially different from the preliminary estimate, the preliminary estimate R_K _PRELIM is also updated with the new estimate R_K and a new iteration is started. This is repeated until the relative difference between R_K and R_K _PRELIM falls under ACCUR_RAD.

The iterations are terminated when the stop criterion is satisfied (see below).

The algorithm was implemented in MATLAB and is publicly available for data analysis. Note that this implementation can deal with missing values often occurring in expression data (also see Troyanskaya et al. (2001)). We used the methodology suggested by Kaufman and Rousseeuw (1990) to handle missing values. A detailed mathematical description of the missing values management can be found on our supplementary web site.

Below we will discuss the initialisation of the preliminary estimate of the radius before the first iteration, the procedures used in Step 1 and 2, the stop criterion (WHILE loop) and the computational and memory complexity of the overall algorithm.

Radius estimate initialisation

In the global cluster algorithm, the preliminary estimate of the radius (R_K _PRELIM) has to be initialised before the first iteration (radius estimate for the first cluster – line 3 of Adap_Cluster). We use

half of the radius of the hypersphere defined by normalisation of the expression profiles (see above). This is given by:

$$R_K_PRELIM = \frac{\sqrt{E-1}}{2} \quad (4)$$

where E is the dimension of the gene expression vectors (number of expression vector components).

Localisation of a cluster center - quality-based clustering (Step 1)

Given a collection G of gene expression profiles, the objective of Step 1 is to find a cluster center in an area of the data set where the 'density' (or number) of expression profiles, within a sphere with radius or quality equal to R_K_PRELIM (preliminary estimate of the radius), is locally maximal. The method described here is based on the principles used by Heyer et al. (1999) but is significantly faster (also see discussion – Table 2). The disadvantage with this approach is that the quality or radius of the clusters is a parameter that is not very intuitive (it is often hard to find a 'good' value for this parameter; often a trial-and-error approach is used with manual validation of the clusters). Furthermore, all the clusters are forced to have the same radius.

The basic steps of the algorithm used for the first step are described below:

$\mathbf{C_K}$ = locate_cluster_center (**G**, $\mathbf{R_K}$ **_PRELIM**)

MAXITER = 50 /* Set internal tuning parameter – maximum number of iterations */

DIV = 1/30 /* Set internal tuning parameter – fraction needed to determine DELTARAD */

$\mathbf{C_K}$ = mean (**G**) /* Cluster center initialisation */

RAD = max ($\{\|g_i - \mathbf{C_K}\| \mid g_i \in \mathbf{G}\}$) /* Start with maximal radius */

DELTARAD = (**RAD** - $\mathbf{R_K}$ **_PRELIM**) * ***DIV*** /* Determine step for decreasing radius */

RAD = **RAD** - **DELTARAD** /* Decrease radius */

GENES_IN_SPHERE = $\{g_i \in \mathbf{G} \mid \|g_i - \mathbf{C_K}\| < \mathbf{RAD}\}$ /* Determine profiles within sphere */

ME = mean (**GENES_IN_SPHERE**) /* Recalculate mean */

ITER = 1

WHILE (**ME** $\neq$ $\mathbf{C_K}$ AND **ITER** < ***MAXITER***) OR **RAD** > $\mathbf{R_K}$ **_PRELIM**

/* Iterate until convergence or maximal number of iterations has been reached */

    ITER = **ITER** + 1

    $\mathbf{C_K}$ = **ME** /* Move cluster center to cluster mean */

    IF **RAD** > $\mathbf{R_K}$ **_PRELIM**

        RAD = **RAD** – **DELTARAD**

        /* Decrease radius if desired quality has not yet been reached */

    END IF

    GENES_IN_SPHERE = $\{g_i \in \mathbf{G} \mid \|g_i - \mathbf{C_K}\| < \mathbf{RAD}\}$ /* Determine profiles within sphere */

    ME = mean (**GENES_IN_SPHERE**) /* Re-calculate mean */

END WHILE

IF **ME** $\neq$ $\mathbf{C_K}$

    $\mathbf{C_K}$ = empty /* Undefined cluster center if no convergence */

END IF

After initialisation of the cluster center (with the mean profile of all the expression profiles in the data set G), all the expression profiles within a sphere with radius RAD are selected. Iteratively, the mean profile of these expression profiles is calculated and subsequently the cluster center is moved to this mean profile. This approach moves the cluster in the direction where the 'density' of profiles is higher (conceptually visualised in Figure 1).

The radius RAD of the sphere is initialised so that all profiles in the data set are located within this sphere. Every iteration, this radius is decreased with a constant value (DELTARAD, a fraction (DIV) of the difference between the initial value of RAD and R_K _PRELIM) until the radius has reached the desired value (R_K _PRELIM) and then remains constant. In the first iterations (when RAD is still 'large') this technique will move the cluster center to regions of the data where the 'global' density is higher (these regions often contain the largest cluster(s)). After some iterations (when RAD is equal or close to R_K _PRELIM) the cluster center will move towards an actual cluster where the density is 'locally' higher.

Convergence is reached if the cluster center remains stationary after RAD has reached R_K _PRELIM. If this does not happen within a certain (MAXITER) number of iterations, C_K is emptied and the algorithm stops.

Note that, the number of distance calculations performed during each iteration of locate_cluster_center is equal to the number (=N) of all expression profiles in G (only the distances from the expression profiles to the current cluster center have to be calculated). Note also that the computational complexity of the calculation of one distance is O(E) (E is the dimensionality of the expression vectors). Because the number of iterations is limited (MAXITER), the computational complexity for the localisation of one cluster center is thus $O(N \times E)$.

Re-estimation or adaptation of the cluster quality (Step 2)

In the previous paragraph we located a cluster center C_K in a collection G of gene expression profiles, using a preliminary estimate R_K _PRELIM of the radius of the cluster. The objective of the method described in this paragraph is, given the cluster center that remains fixed, to re-calculate the radius R_K of the current cluster as to assess that genes belonging to this cluster are significantly coexpressed.

To substantiate the method described here, we introduce a randomised version of the original data set where the components of each expression vector are randomly and independently permuted (Herrero et al., 2001). This randomised version of the data will only be used for conceptual reasons, it will not be used during the actual calculations. This process of randomisation destroys the correlation between the expression vectors that was introduced through non-accidental mechanisms (e.g., experimental setup). Any correlation still existing after this procedure can be attributed to chance.

First, we calculate the Euclidean distance r from every expression vector in the data set to the cluster center C_K. Imagine doing the same for every vector present in the randomised data. The distribution of these distances in the original data consists of two parts (this approach has some similarities with the work of Sharan et al. (2000)) – see Figure 2:

1. Background: these are the expression profiles with a distance to the cluster center that is also significantly present in the distance distribution of the randomised data set. Genes belonging to the background of the current cluster center either do not belong to any cluster (noise; are not significantly coexpressed with other genes) or belong to another cluster. Genes belonging to other clusters (if not too dominant) will not significantly show up in the distance distribution for the current cluster center (they 'drown' in the noise or background).
2. Cluster: these are the expression profiles with a distance to the cluster center that is not significantly present in the distance distribution of the randomised data set (left-sided tail in the distribution of the original data set). Genes belonging to the cluster are significantly coexpressed.

To calculate the true radius of the cluster we need to construct a model (probability density estimation) describing the total distribution of the distance r in the original data. We propose the following model structure:

$$p(r) = P_C . p(r \mid C) + P_B . p(r \mid B) \quad (5)$$

where

$$p(r \mid C) = \frac{S_D}{\left(2\pi\sigma^2\right)^{D/2}} r^{D-1} \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (6)$$

$$p(r \mid B) = \frac{S_D}{S_{D+1}\left(D+1\right)^{D/2}} r^{D-1} \quad (7)$$

$$P_C + P_B = 1 \quad (8)$$

and

$$D = E - 2 \quad (9)$$

$$S_D = \frac{2\pi^{D/2}}{\Gamma(D/2)} \quad (10)$$

$$\Gamma(x) = \int_0^\infty u^{x-1} e^{-u}\, du. \quad (11)$$

E is the dimensionality of the gene expression vectors, S_D is the surface area of a unit sphere in D dimensions and $\Gamma(x)$ is the gamma function.

Note that the model structure assumes that the distance measure used for r is the Euclidean distance. This means that our method cannot be directly extrapolated to other distance measures.

The model for the total distribution described in Equation (5) is the sum of two terms (also see Figure 2). One term represents the distribution of the cluster (see Equation (6)), the other term represents the distribution of the background (see Equation (7)), each multiplied by the associated a priori probability (P_C and P_B). Note that Equations (6) and (7) are only valid for normalised gene expression vectors. Note also that this model is an approximation and only reliable in the neighbourhood of the cluster. A detailed mathematical discussion of Equations (6) and (7) and of the assumptions used to construct them, can be found on our supplementary web site. Notice that two parameters (σ and P_C (or P_B)) still have to be determined by fitting the model to the distance distribution of the original data (the randomised data is *not used* for the actual calculations). This is done by an EM-algorithm (Bishop, 1995). We use the preliminary estimate of the radius R_K_PRELIM (see localisation of a cluster center) to initialise the two parameters to be determined by the EM-algorithm. Because the model only has to fit the distribution of r (distance to the cluster center – one dimension), the computational complexity of the EM-algorithm is low as compared to the computational complexity of the cluster center localisation in Step 1 and therefore can be neglected if E is sufficiently large. The accuracy of the fit (which represents the validity of the assumptions we made to construct our model) for the clusters found in the Cho et al. data (see Figure 3 on our supplementary web site and the Results section) can be inspected in Figure 2 for the first four clusters of Figure 3 and on our supplementary web site for all the clusters of Figure 3.

After the estimation of σ and P_C, we determine the radius R_K of the current cluster so that points that are assigned to the cluster have a probability S or more (significance level - user-defined; default setting: S = 95%) to belong to the cluster:

$$P(C \mid R_K) = \frac{P_C . p(R_K \mid C)}{P_C . p(R_K \mid C) + P_B . p(R_K \mid B)} = S. \quad (12)$$

To summarise, the complete input-output relation of the method explained in this paragraph is given by

R_K = recalculate_radius (**G**, **C_K**, **R_K _PRELIM**, **S**).

R_K will be empty if C_K is empty (cluster center localisation did not converge) or if the EM-algorithm to determine the model parameters did not converge.

Stop criterion

The iteration (WHILE loop) in the global algorithm ends when the stop criterion is satisfied. This is the case when one of the three following conditions holds true:

1. Step 1 or 2 stops converging.
2. If, for a specific cluster, the number of iterations necessary to decrease the relative difference between R_K and R_K_PRELIM (under ACCUR_RAD), is larger than a predefined number.
3. If the clusters removed from the data are not valid (number of genes below MIN_NR_GENES) for a predefined and consecutive number of times.

Computational and memory complexity of the global algorithm

It is difficult to give an exact measure for the computational complexity of this heuristic approach. However, we can give an indication of the role of the most important variables. As previously said, the computational complexity of one cluster center localisation is approximately $O(N \times E)$ (N is the number of gene expression profiles in the data set, E is the dimensionality of the expression vectors) and the computational complexity of the re-estimation of the cluster quality is negligible. So, the computational complexity of one iteration in the global algorithm (WHILE loop) is also approximately $O(N \times E)$. Notice also that Condition 2 of the stop criterion sets a limit for the maximum number of iterations in the global algorithm needed to define one cluster (which is only valid if the number of genes in this cluster equals or

exceeds MIN_NR_GENES). Moreover, the number of invalid clusters (number of genes less than MIN_NR_GENES) found before one of the conditions of the stop criterion is true, is in practice also more or less proportional to the number of valid clusters found (e.g., for each invalid cluster found, two valid clusters will be found). This number of valid clusters is no classical attribute of the data (like N or E) used to express computational complexity but it is rather a measure for the complexity of the structure of the data. Taken together, this means that the number of iterations in the global algorithms is also more or less proportional to this number of valid clusters in the data set and since the computational complexity of one iteration is approximately $O(N \times E)$, the computational complexity of the global algorithm is thus approximately $O(N \times E \times VC)$ (VC = number of valid clusters). Notice also that, after finding a certain number of clusters, the number of genes left in the data is smaller than N (clusters are discarded from the data). The computational complexity, as described above, is thus an upper limit.

Since only the distances from the expression profiles to the current cluster center have to be kept in memory (this is true at any stage of the algorithm), the memory complexity of the global algorithm is O(N).

Results

Mitotic cell cycle of Saccharomyces cerevisiae

The algorithm was tested on the expression profiling experiment of Cho et al. (1998), studying the yeast cell cycle in a synchronised culture on an Affymetrix chip (also see Spellman et al. (1998)). This data set (http://cellcycle-www.stanford.edu) can be considered as a benchmark (Heyer et al., 1999; Jakt et al., 2001; Tamayo et al., 1999; Tavazoie et al., 1999; Yeung et al., 2001) and contains expression profiles for 6220 genes over 17 time points taken at 10-min intervals, covering nearly two full cell cycles. The majority of the genes included in the data set have been functionally classified (Mewes et al., 2000), which makes this data set an ideal candidate to correlate the results of new clustering algorithms with the biological reality. For more details about the data set itself, we refer to the original paper of Cho et al. (1998).

Our pre-processing included the following steps: data corresponding to the 90 and 100-min measurements were removed (Tavazoie et al., 1999). Also, we selected the 3000 most variable genes using σ/μ as a metric of variation (Tavazoie et al., 1999) (filtering). Finally, we normalised the gene expression

profiles as described in the normalisation section. The results of the cluster analysis with our algorithm (MIN_NR_GENES=10, S=0.95) are shown in Figure 3 (see supplementary web site). Table 1a summarises the biological validation of this result by looking for enrichment of functional categories in individual clusters as described in Tavazoie et al. (1999). We mapped the genes in each cluster to the functional categories in the Munich Information center for Protein Sequences (MIPS) (Mewes et al., 2000) Comprehensive Yeast Genome Database. For each cluster we calculated P-values for observing the frequencies of genes in particular functional categories using the cumulative hypergeometric probability distribution. In the same table we also show, as a comparison and in parallel (where possible, we compare P-values of functionally matching clusters), the results obtained by Tavazoie et al. (1999) on the same data using the K-means algorithm. Note that the three most important clusters found by Tavazoie et al. (1999) (cluster 1, 4 and 2 in Tavazoie et al. (1999)) could be matched with three clusters discovered by Adap_Cluster (cluster 1, 2 and 5). The degree of enrichment in the clusters identified by Adap_Cluster, however, was considerably higher and biologically more consistent.

In the biological validation and comparison discussed above, we filtered the data using the same metric of variance (σ/μ) as proposed by Tavazoie et al. (1999) because different filtering strategies could produce different clusters independent of the clustering technique (we did not want different filtering to interfere with our comparison). However, in general, if filtering is performed, we recommend using simple measures of variation, like the standard deviation σ (not σ/μ) or the difference between the minimum and maximum value, together with Adap_Cluster. Using Adap_Cluster with the Cho et al. (1998) data indeed resulted in biologically more relevant results when using the standard deviation (σ) as the metric of variance. This analysis produced several clusters enriched in top-level functional categories (see Table 1b).

We were able to determine the role of every cluster presented in Table 1b within the yeast cell cycle context and correlate this role with the behaviour of the average profiles of the clusters. We have also found several protein complexes where nearly all members belong to the same cluster. A detailed discussion of these findings can be found on our supplementary web site.

Note that the results of Adap_Cluster in this section have been obtained without additional fine-tuning (we used the default value for S) of one or more parameters (unlike, for example, K-means (used by Tavazoie et al. (1999)) where the number of clusters has to be estimated in advance, which is certainly not

trivial) and that these results can be obtained very easily and almost instantaneously (maximum 1.5 minutes for the examples above on a typical PC).

Additional results/simulations

A discussion of the clusters found by our algorithm in other data sets (with different mathematical and biological characteristics) can be found on our supplementary web site:

- Response to mechanical wounding in Arabidopsis (Reymond et al.,2000)
- Central nervous system development (Wen et al., 1998)
- Measurement of expression levels in different tissues (data not publicly available – manuscript in preparation)
- Artificial data (with and without missing values)

Discussion

The algorithm proposed in this paper is designed to find clusters of significantly coexpressed genes (higher degree of coexpression than could be expected by chance) in high-density areas of the data (high-density areas were assumed by Heyer et al. (1999) to be, biologically seen, the most interesting regions in the data). Genes not exhibiting an expression profile significantly similar to the expression profile of other genes in the data are not assigned to any of the clusters. The same applies to genes lying in low-density areas of the data. The size or radius for each cluster separately is determined - through the significance level S - by making a trade-off between the probability of false positive results (a gene assigned to the cluster that is not really coexpressed with the other members of the cluster) and the probability of false negative results (genes not assigned to the cluster but coexpressed with the members of the cluster). The default value for the significance level S guarantees that a gene, which has been assigned to the cluster, has a probability of 95% or more to belong to the cluster (this means that the probability of being a false positive is 5% or less). In other words, the genes in the cluster are significantly coexpressed with a certain confidence. The significance level S, in turn, can be seen as a constant quality criterion for the clusters (while the quality criterion R as defined in Equation (3) differs among the clusters defined by our algorithm). Our algorithm can thus be regarded as being a pure quality-based clustering method where all the clusters have a constant quality represented by the significance level S (the term adaptive quality-

based clustering is thus only valid when using Equation (3) as quality criterion). When compared to the previous definition (quality measure R), this new quality measure S has the advantage that it has a strict statistical meaning (it is much less arbitrary) and that, in most cases, it can be chosen independently of a specific data set or cluster. In addition, it allows for the setting of a meaningful default value (95%).

In Table 2a a detailed comparison between our global algorithm (Adap_Cluster) and the algorithm proposed by Heyer et al. (1999) (QT_Clust) is made. Because we focus on algorithmic aspects, the QT_Clus algorithm in our comparison uses the same distance and quality measure as we did (Euclidean distance and quality defined as in Equation (3) – In Heyer et al. (1999) the jackknife correlation was used together with a quality measure defined as a diameter). This change of distance and quality measure does not significantly change the structure of QT_Clus and in essence, there is no fundamental difference between a quality defined as a radius and a quality defined as a diameter.

To complete the picture, Table 2b gives a summary of the differences between our method, hierarchical clustering, SOM and K-means.

Clusters formed by our algorithm might be good 'seeds' for further analysis of expression data (Thijs et al., 2002) since they only contain a limited number of false positives. When the presence of false positives in a cluster is undesirable, a more stringent value for the significance level S might be applied (e.g., 99%; for noise-sensitive analyses such as motif finding) which will result in smaller clusters exhibiting a more tightly related expression profile.

Conclusively, Adap_Cluster can be considered as an intuitively appealing, user-friendly (no need for a predefinition of the number of clusters, statistical and intuitive definition of the quality measure with a meaningful default value) and fast clustering algorithm.

Acknowledgements

Frank De Smet is a research assistant at the K.U.Leuven. Yves Moreau is a post-doctoral researcher of the FWO. Bart De Moor is a full professor at the K.U.Leuven. This work is supported by the Flemish Government (Research Council KUL (GOA Mefisto-666, IDO), FWO (G.0256.97, G.0240.99, G.0115.01, Research communities ICCoS, ANMMM, PhD and postdoc grants), Bil.Int. Research Program, IWT (Eureka-1562 (Synopsis), Eureka-2063 (Impact), Eureka-2419 (FLiTE), STWW-Genprom, IWT

project Soft4s, PhD grants)), Federal State (IUAP IV-02, IUAP IV-24, Durable development MD/01/024),

EU (TMR-Alapades, TMR-Ernsi, TMR-Niconet), Industrial contract research (ISMC, Data4s, Electrabel,

Verhaert, Laborelec). The scientific responsibility is assumed by its authors.

References

Ben-Dor, A., Shamir, R., and Yakhini, Z. (1999) Clustering gene expression patterns. *J. Comput. Biol.*, **6**, 281-297.

Bishop C.M. (1995) *Neural Networks for pattern recognition.* Oxford University Press Inc., New York.

Bittner, M., Meltzer, P., Chen, Y., Jiang, Y., Seftor, E., Hendrix, M., Radmacher, M., Simon, R., Yakhini, Z., Ben-Dor, A., Sampas, N., Dougherty, E., Wang, E., Marincola, F., Gooden, C., Lueders, J., Glatfelter, A., Pollock, P., Carpten, J., Gillanders, E., Leja, D., Dietrich, K., Beaudry, C., Berens, M., Alberts, D., and Sondak, V. (2000) Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, **406**, 536-540.

Carr, D.B., Somogyi, R., and Michaels, G. (1997) Templates for looking at gene expression clustering. *Statistical Computing & Statistical Graphics Newsletter*, **8**, 20-29.

Cho, R.J., Campbell, M.J., Winzeler, E.A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T.G., Gabrielian, A.E., Landsman, D., Lockhart, D.J. and Davis, R.W. (1998) A genome wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, **2**, 65-73.

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

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

Hastie, T., Tibshirani, R., Eisen, M.B., Alizadeh, A., Levy, R., Staudt, L., Chan, W.C., Botstein, D. and Brown, P. (2000) 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biol*., **1**, research0003.1-0003.21.

Herrero, J., Valencia, A., and Dopazo, J. (2001) A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics*, **17**, 126-136.

Heyer, L.J., Kruglyak, S., and Yooseph, S. (1999) Exploring expression data: Identification and analysis of coexpressed genes. *Genome Res.*, **9**, 1106-1115.

Jakt, L.M., Cao, L., Cheah, K.S., and Smith, D.K. (2001) Assessing clusters and motifs from gene expression data. *Genome Res.*, **11**, 112-123.

Kaufman, L. and Rousseeuw, P.J. (1990) *Finding groups in data: an introduction to cluster analysis.* Wiley, New York.

Lander, E.S. (1999) Array of hope. *Nat. Genet.*, **21**, 3-4.

Lukashin, A.V. and Fuchs, R. (2001) Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters. *Bioinformatics*, **17**, 405-414.

Mewes, H.W., Frishman, D., Gruber, C., Geier, B., Haase, D., Kaps, A., Lemcke, K., Mannhaupt, G., Pfeiffer, F., Schuller, C., Stocker, S., and Weil, B. (2000) MIPS: a database for genomes and protein sequences. *Nucleic Acids Res.*, **28**, 37-40.

Quackenbush J. (2001) Computational analysis of microarray data. *Nat. Rev. Genet.*, **2**, 418-427.

Reymond, P., Weber, H., Damond, M., and Farmer, E.E. (2000) Differential gene expression in response to mechanical wounding and insect feeding in Arabidopsis. *Plant Cell*, **12**, 707-720.

Roth, F.P., Hughes, J.D., Estep, P.W., and Church, G.M. (1998) Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole genome mRNA quantitation. *Nat. Biotechnol.*, **16**, 939-945.

Schena, M., Shalon, D., Davis, R., and Brown, P. (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, **270**, 467-470.

Sharan, R. and Shamir, R. (2000) CLICK: a clustering algorithm with applications to gene expression analysis. Proc. ISMB 2000, pp. 307-316.

Sherlock, G. (2000) Analysis of large-scale gene expression data. *Curr. Opin. Immunol*., **12,** 201-205.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., and Futcher, B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, **9**, 3273-3297.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E.S. and 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**, 2907-2912.

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

Thijs, G., Moreau, Y., De Smet, F., Mathys, J., Lescot, M., Rombauts, S., Rouze, P., De Moor, B., and Marchal, K. (2002) INCLUSive: INtegrated Clustering, Upstream sequence retrieval and motif Sampling. *Bioinformatics*, in press.

Tou, J.T. and Gonzalez, R.C. (1979) Pattern classification by distance functions. In Tou, J.T. and Gonzalez, R.C. (eds), *Pattern recognition principles*. Addison-Wesley, Reading (Mass.), pp. 75-109.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R.B. (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**, 520-525.

van Helden, J., Rios, A.F., and Collado-Vides, J. (2000) Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucl. Acids Res.*, **28**, 1808-1818.

Wen, X., Fuhrman, S., Michaels, G.S., Carr, D.B., Smith, S., Barker, J.L., and Somogyi, R. (1998) Large-scale temporal gene expression mapping of central nervous system development. . *Proc. Natl. Acad. Sci. USA*, **95**, 334-339.

Yeung, K.Y., Haynor, D.R., and Ruzzo, W.L. (2001) Validating clustering for gene expression data. *Bioinformatics*, **17**, 309-318.

Table 1a. Biological validation of the cluster result in Figure 3 (see supplementary web site) and comparison with the result of Tavazoie et al. (1999).
The genes in each cluster have been mapped to the functional categories in the MIPS database and ($-\log_{10}$ transformed) P-values (representing the degree of enrichment – also see text) have been calculated for each functional category in each cluster. Only significantly enriched functional categories are shown ($-\log_{10}$ transformed P-values ≥ 4) and clusters without enrichment are not listed. As a comparison and in parallel (functionally matching clusters are shown in the same row), the results obtained by Tavazoie et al. (1999) (K-means) are also included. NR = Not Reported.

Cluster number Adap_Cluster	Cluster number K-means (Tavazoie et al.)	Number of ORFs Adap_Cluster	Number of ORFs K-means (Tavazoie et al.)	MIPS functional category	ORFs within functional category Adap_Cluster	ORFs within functional category K-means (Tavazoie et al.)	P-value ($-\log_{10}$) Adap_Cluster	P-value ($-\log_{10}$) K-means (Tavazoie et al.)
1	1	302	164	ribosomal proteins	101	64	80	54
				organisation of cytoplasm	146	79	77	39
				protein synthesis	119	NR	74	NR
				cellular organisation	211	NR	34	NR
				translation	17	NR	9	NR
				organisation of chromosome structure	4	7	1	4
2	4	315	170	mitochondrial organization	62	32	18	10
				energy	35	NR	8	NR
				proteolysis	25	NR	7	NR
				respiration	16	10	6	5
				ribosomal proteins	24	NR	4	NR
				protein synthesis	33	NR	4	NR
				protein destination	49	NR	4	NR
5	2	98	186	DNA synthesis and replication	20	23	18	16
				cell growth, cell division and DNA synthesis	48	NR	17	NR
				recombination and DNA repair	12	11	8	5
				nuclear organization	32	40	8	4
				cell-cycle control and mitosis	20	30	7	8
6		58		mitochondrial organisation	15		7	
				peroxisomal organization	4		4	
				energy	9		4	
8		58		tRNA-synthetases	5		5	
				organisation of cytoplasm	14		4	
16		15		cellular transport and transport mechanisms	6		4	
21	17	20	83	transcription	9	21	4	4
31	14	28	74	organisation of centrosome	3	6	4	6
				nuclear biogenesis	1	3	2	5
				organisation of cytoskeleton	2	7	2	4
36		14		tRNA transcription	3		4	
37	18	10	101	organisation of cytoplasm	7	30	6	9
				ribosomal proteins	4	16	4	7
				protein synthesis	4	20	3	7
				cellular organisation	7	55	2	5
40		11		organization of endoplasmatic reticulum	4		4	
42		12		cellular transport and transport mechanisms	6		4	
	5		152	cell rescue, defense, cell death		22		5
				carbohydrate metabolism		24		4
				stress response		12		4
				energy		16		4
				metabolism of energy reserves		6		4
	7		101	cell-cycle control and mitosis		17		5
				budding, cell polarity, filament formation		10		4
				DNA synthesis and replication		7		4
	8		148	TCA pathway		5		4
				Carbohydrate metabolism		22		4
	21		70	protein synthesis		14		5
				organisation of cytoplasm		18		5
				ribosomal proteins		10		4
	30		60	nitrogen and sulphur metabolism		9		8
				amino acid metabolism		12		7

Table 1b. Biological validation of the cluster analysis of the Cho et al. (1998) data with Adap_Cluster (MIN_NR_GENES=10, S=0.95) using the standard deviation (σ) as the metric of variance for filtering. The algorithm retrieved 38 clusters. We looked for enrichment (represented by the P-values) of *top-level* functional categories (from the MIPS database) in individual clusters. Notice the periodic behaviour of the clusters enriched with cell-cycle specific genes (cluster 3, 6 and 9).

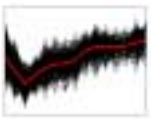
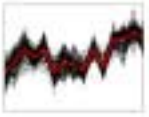
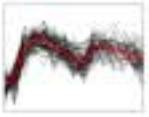
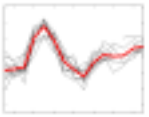
Cluster number	Graphical representation of cluster	Number of ORFs	MIPS functional category (top-level)	ORFs within functional category	P-value ($-\log_{10}$)
1		426	energy transport facilitation	47 40	10 5
3		196	cell growth, cell division and DNA synthesis	48	5
4		149	protein synthesis cellular organisation	71 107	50 19
5		159	cell rescue, defense, cell death and ageing	20	4
6		171	cell growth, cell division and DNA synthesis	76	24
9		78	cell growth, cell division and DNA synthesis	23	4
37		11	metabolism	9	6

Table 2a. Comparison between Adap_Cluster and QT_Clust.

	Adap_Cluster	**QT_Clust**
User-defined parameters	1. Data set G 2. Significance level S 3. Minimum number of genes MIN_NR_GENES	1. Data set 2. Quality (radius R or diameter d) 3. Minimum number of genes (termination criterion)
Computational Complexity	~ $O(N \times E \times VC)$	~ $O(N^2 \times E \times VC)$
Cluster radius R	Automatically calculated for each cluster separately – not constant	Constant and user-defined
Quality measure	Significance level S: statistical parameter that can be chosen independently of data set (default value (0.95) almost always gives meaningful results)	Arbitrary parameter that has to be set by the user in function of a specific data set, after visual inspection of clusters formed at different quality-levels (optimal value is not straightforward) – no meaningful default value
Number of clusters	Not predefined	Not predefined
Inclusion of all genes in clusters	No	No
Result	Deterministic	Deterministic

Table 2b. Comparison between Adap_Cluster, hierarchical clustering, Self-Organizing Maps and K-means.

	Adap_Cluster	**Hierarchical clustering** Eisen et al. (1998)	**SOM** Tamayo et al. (1999)	**Standard K-means** Tou and Gonzalez (1979)
Format of result	Set of clusters	Tree structure difficult to interpret for large data sets	Set of predefined number of clusters	Set of predefined number of clusters
Principal user-defined parameter	Significance level S	-	Number of clusters / Node topology	Number of clusters
Additional requirements from the user	Limited (fine-tuning of S is rarely necessary)	Definition of (an arbitrary) level where the tree structure has to be cut	Extensive parameter fine-tuning (comparison of several runs with different parameter settings) is almost always necessary	Extensive parameter fine-tuning (comparison of several runs with different parameter settings) is almost always necessary
Statistical definition of clusters	Yes	No	No	No
Inclusion of all genes in clusters	No	Yes	Yes	Yes
Missing values management	Yes	Yes	Not discussed	Not standard
Computational complexity of one run of the algorithm	Linear in N	Quadratic in N	Linear in N	Linear in N

Figure legends

Fig. 1. Conceptual visualisation of cluster center (X_{CK}) relocation to the cluster mean (X_{ME}) in two dimensions (one iteration – cluster radius constant – data not normalised). The number of profiles (black dots) within the sphere after relocation is substantially higher than the number of profiles before relocation.

Fig. 2. Distribution of r (distance from expression vectors to a certain cluster center) for the first 4 clusters found in the data set from Cho et al (1998) (see Figure 3 on our supplementary web site). In each box, the histogram on the left represents the distribution of r in the actual data and the histogram on the right represents the distribution of r in the randomised data (note that the cluster center for the randomised distribution is identical to the cluster center for the actual distribution – the randomisation is not applied to the cluster center itself). For each cluster, the model (see Equation (5)-(11)) fitted by the EM-algorithm is superposed on the distribution of the actual data (after multiplication with an appropriate factor to fit the scale (this accounts for the bin size and the number of expression profiles in the data set)). The model for the background (see Equation (7)) is also superposed on the distribution of the randomised data.

Fig. 1.

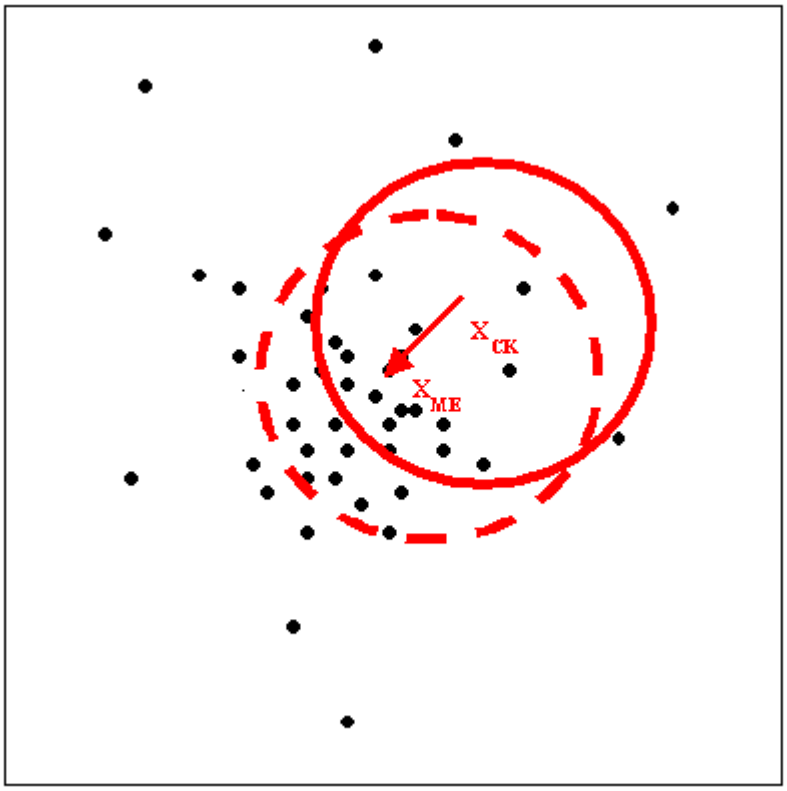
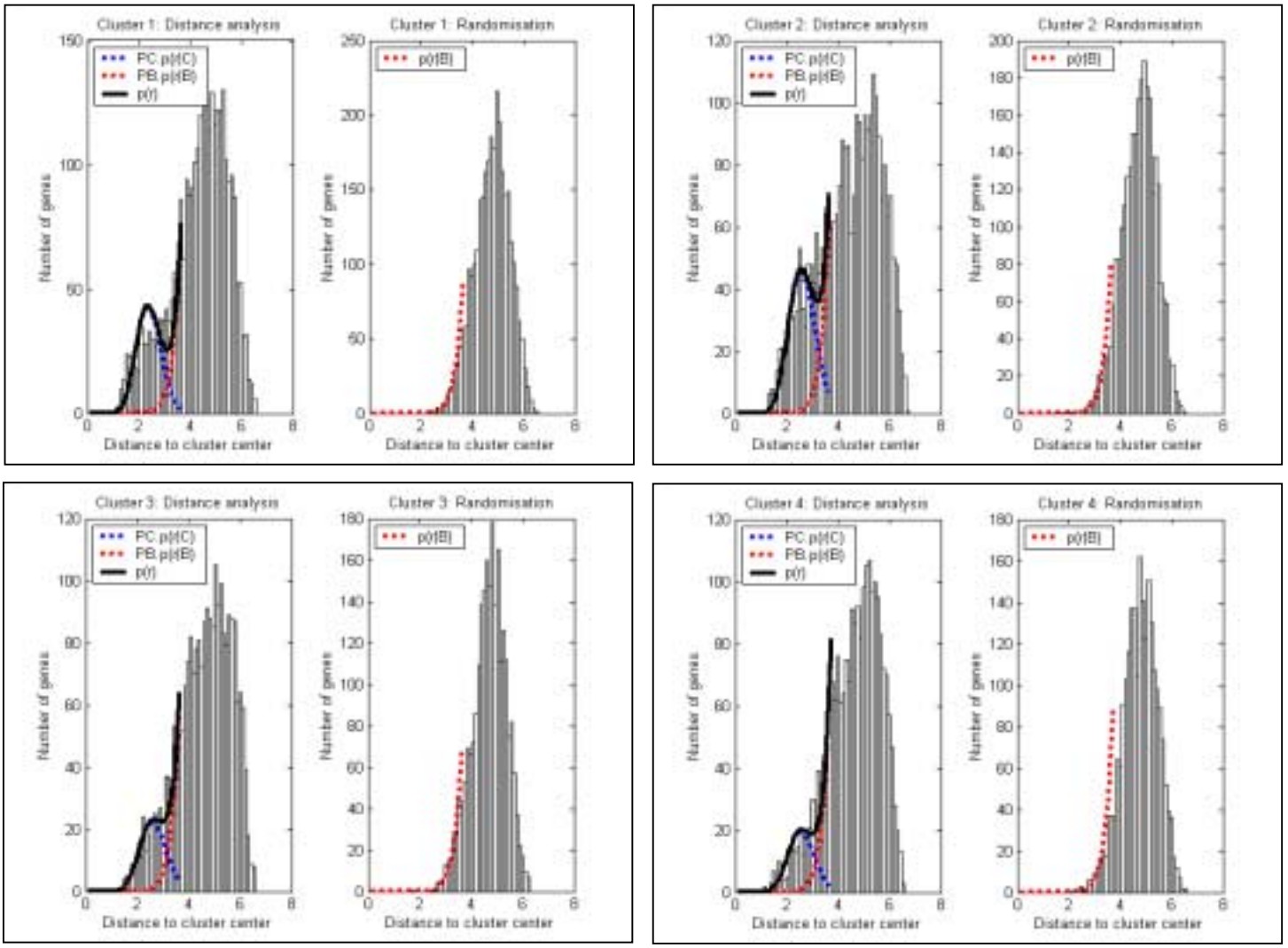


Fig. 2.


Cluster 1: Distance analysis
Cluster 1: Randomisation
Cluster 2: Distance analysis
Cluster 2: Randomisation
Cluster 3: Distance analysis
Cluster 3: Randomisation
Cluster 4: Distance analysis
Cluster 4: Randomisation
PC: p(r|C)
PB: p(r|B)
p(r)
p(r|B)
Number of genes
Distance to cluster center