DNA Microarray Data Clustering Based on Temporal Variation: FCV with TSD Preclustering

Carla S. Möller-Levet, [*] Kwang-Hyun Cho [†] and Olaf Wolkenhauer [‡] [§]

Abstract

The aim of this paper is to present a new clustering algorithm for short time-series gene expression data that is able to characterize temporal relations in the clustering environment (i.e., data-space), which is not achieved by other conventional clustering algorithms such as k-means or hierarchical clustering. The algorithm called fuzzy c-varieties clustering with Transitional State Discrimination preclustering (FCV-TSD) is a two step-approach which identifies groups of points ordered in a line configuration in particular locations and orientations of the data-space that correspond to similar expressions in the time domain. We present the validation of the algorithm with both artificial and real experimental data sets, where k-means and random clustering are used for comparison. The performance is evaluated with a measure for internal cluster correlation and the geometrical properties of the clusters; showing that the TSD-FCV algorithm has better performance than the k-means algorithm on both data sets.

Keywords: Gene expression data, Short time-series, Transitional state discrimination algorithm, fuzzy c-varieties clustering, Saccharomyces cerevisiae microarray data

Running Head: Clustering short microarray time-series data.

[*] Department of Electrical Engineering and Electronics, Control Systems Centre, UMIST, Manchester, U.K.

[†] School of Electrical Engineering, University of Ulsan, Ulsan, 680-749, Korea.

[‡] Department of Biomolecular Sciences and Department of Electrical Engineering and Electronics, UMIST, Manchester, U.K.

[§] *Author for correspondence.* Address: Control Systems Centre, P.O. Box 88, Manchester M60 1QD, U.K. E-mail: o.wolkenhauer@umist.ac.uk, Tel./Fax: +44-(0)161-200-4672.

1 Introduction

A natural and intuitive approach for visualizing information in gene expression data is to group together genes with similar patterns of expression. This grouping can be achieved by cluster analysis (Everitt 1974, Jain and Dubes 1988), a multivariate procedure for detecting natural groupings within data. There are a wide variety of clustering algorithms available from diverse disciplines such as pattern recognition, text mining, speech recognition and social sciences amongst others. The algorithms are distinguished by the way in which they measure distances between objects and the way they group the objects based upon the measured distances. Unsurprisingly, gene expression data has been analyzed using such a wide range of clustering algorithms. Hierarchical clustering (Eisen et al. 1998), self-organizing maps (Tamayo et al. 1999) and k-means algorithm (Tavazoie et al. 1999) are some of the methods that have reported successful results for particular applications. Nevertheless, there is no single method considered as the best choice for clustering gene expression data since the biological context and experimental design of each experiment (i.e., time course vs. comparative study, single or replicated experiment) determines the choice of algorithm, parameters and how to best interpret the data.

In this paper we describe a clustering algorithm for short time-series gene expression data. Clustering time-series is practiced in fields such as finance and economics (Mitchell and Mulherin 1996), speech recognition (Tran and Wagner 2002, Oates 1999) and medicine (Geva and Kerem 1988). Frequency analysis (Bloomfield 1976) and time warping algorithms (Sankoff and Kruskal 1983) are analysis techniques commonly used in these fields. In gene expression research the required sample size to make sense of these techniques is not always possible to obtain. In addition, classical time-series analysis techniques such as regression analysis, autoregressive processes and serial correlation assume that populations from which samples are drawn are normally distributed, otherwise, when the assumption of normality is not satisfied, these procedures can be justified for large samples on the basis of asymptotic theory (Anderson 1958). Most of the gene expression time-series come from an unknown distribution (Kruglyak and Tang 2001) and are usually very short, therefore,

traditional techniques have to be modified or new strategies have to be implemented.

Gene expression data is usually represented in a matrix known as the Gene Expression Matrix (GEM), where columns represent time points or biological conditions and rows represent the genes. In the data-space, each gene is represented as a point in an n-dimensional space, where the n dimensions correspond to the n sampling time points, as illustrated in Figure 1.

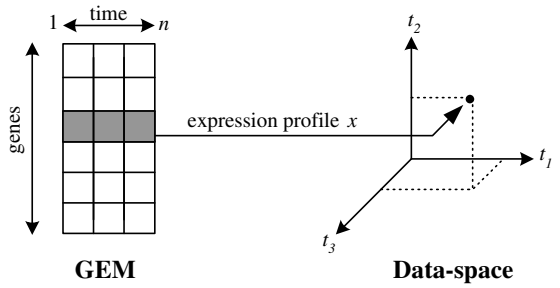


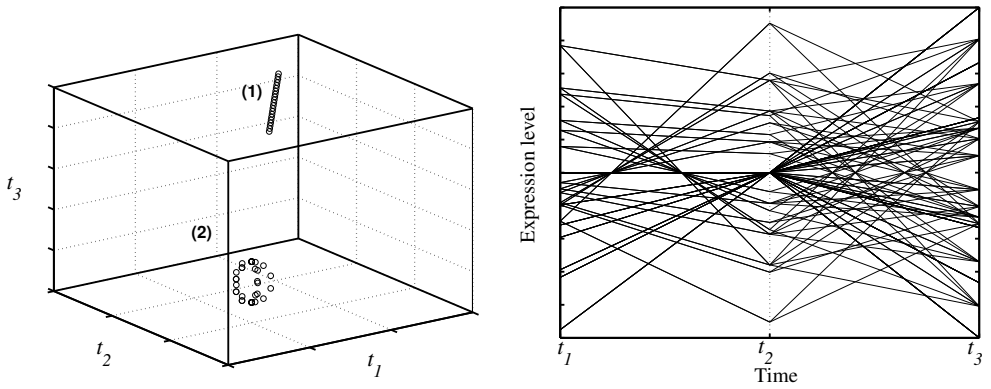
Figure 1: In the data-space, each gene is represented as a point in an n-dimensional space, where the n dimensions correspond to the n sampling time points.

While a time-series expression profile can mathematically be treated as a row vector and thus be clustered by any algorithm that compares and groups genes as points in the data space, here we emphasize the temporal order of measurements, which in general does not allow a change in the order of the columns in the GEM. The algorithm we propose is able to characterize temporal relations in the clustering environment (i.e., data-space) which is not achieved by other conventional clustering algorithms such as k-means or hierarchical clustering. We find that the location, orientation, and shape of the group of points in the data-space are related to different kinds of relations between profiles in the time domain. We can use this information to define clustering targets that reflect similarity in the time domain. The algorithm we present in this paper, referred to as fuzzy c-varieties (FCV) clustering with Transitional State Discrimination (TSD) preclustering (which is to be called FCV-TSD algorithm hereafter), is a two-step approach: First the algorithm, described in Section 3, groups the points in relevant locations and orientations and then the FCV algorithm (Bezdek 1981) looks for linearly shaped clusters within each particular group.

This paper is organised as follows: Section 2 addresses the concept of similarity for time-series and introduces the main idea of the FCV-TSD algorithm. In Section 3, the objectives and basic concepts of the FCV and TSD algorithms are presented and followed by the description of their use in the FCV-TSD algorithm. Section 4 presents the validation of the algorithm with synthetic and real experimental data sets, where k-means and random clustering are used for comparison. The performance is evaluated with a measure for the internal cluster correlation using the Spearman rank-order correlation coefficient, and with the geometry of the clusters. Finally, conclusions are made in Section 5 summarizing the presented research.

2 Similarity of time-series

The first part of this section introduces the concept of similarity for time-series expression profiles when k-means clustering is applied. An example with two real gene expression profiles is analyzed and a more comprehensive concept of similarity is proposed as a basis for the FCV-TSD algorithm.



(a) Two clusters with different shapes in a 3D data-space. For a time-series the three axis correspond to the time points t_1, t_2 and t_3.

(b) Time-series for the spherically shaped cluster (2) of figure (a).

Figure 2: Data-space and time domain representation.

The collection of points that form groups in the data-space can have different shapes, such as the spherical and the linearly shaped clusters shown in Figure 2(a). Clustering algorithms show a preference for a particular cluster shape determined by the selection of the

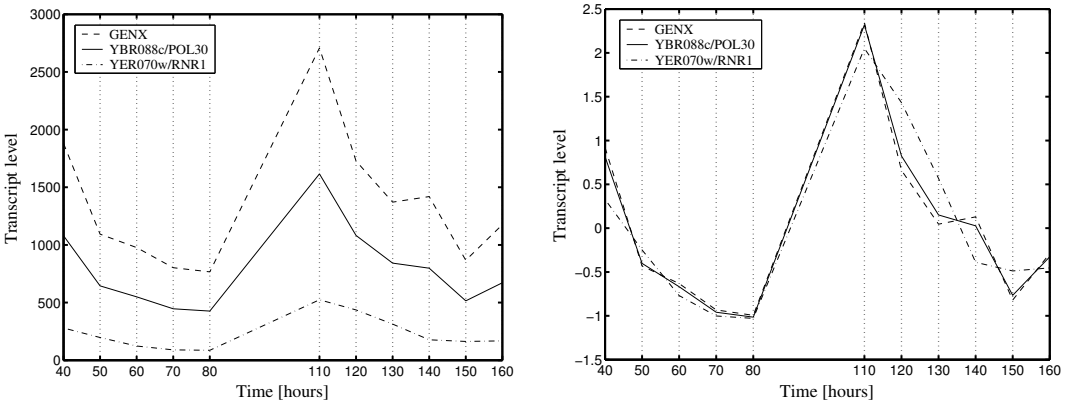
distance norm, objective function and computation of the elements therein. The k-means algorithm looks for circles in $\mathbb{R}^2$, spheres in $\mathbb{R}^3$ or hyperspheres in $\mathbb{R}^n$. By preferring these shapes, the algorithm clusters expression profiles with similar absolute expression levels without considering the shape of the expression profile between dimensions (i.e., time-points). This is illustrated in Figure 2(b) which shows the time-series for the spherically shaped cluster of Figure 2(a). However, it is the overall shape rather than absolute values that are usually relevant in gene expression data analysis. Consequently, a preliminary transformation of the GEM is required for the k-means algorithm to consider the shape of the expression profile. This transformation is the standardization of the time-series to z-scores, i.e., the gene expression profiles are scaled to zero mean and unit standard deviation (Tavazoie et al. 1999, Tamayo et al. 1999). The z-score of the ith time point of a gene x is defined in (1), where $\overline{x}$ is the mean and s_x the standard deviation of all the time points $x_1, ..., x_n$ in vector x:

$$z_i = \frac{(x_i - \overline{x})}{s_x} \tag{1}$$

To visualize the effects in the time domain of this standardization, consider the following example. The microarray analysis of *Saccharomyces cerevisae* by Cho et al. (1998) shows that YBR0088x POL30 and YER070w RNR1 are two of the nineteen functionally characterized genes putatively involved in DNA replication during the late G1 phase of the mitotic cell cycle. These genes present similar expression profiles but different absolute expression levels along the time course experiment. The difference from each time point of POL30 to RNR1 is calculated. The differences are used to create a synthetic gene (GENEX) with POL30 as a reference, such that GENEX and RNR1 have the same Euclidean distance to POL30 in every time point but in opposite directions. After the z-score standardization, the Euclidean distances are recalculated and show that GENEX is closer to POL30 than RNR1. Figure 3 shows that after the standardization, the difference of the absolute expression level of genes with similar shape of expression profile is neglected and original distance relationships over time are transformed. The distance relationships after standardization are related to the strength of linear relationship between genes. The strength of linear relationships between variables can be measured by the sample linear

correlation coefficient, r, (Maurice and Kendall 1961) as defined by (2) where n is number of pairs of observations, $\overline{x}$ is the average and s_x is the standard deviation of the vector x, and $\overline{y}$ is the average and s_y is the standard deviation of the vector y.

$$r\left(x,y\right)=\frac{\sum_{i=1}^{n}\left(x_i-\overline{x}\right)\left(y_i-\overline{y}\right)/\left(n-1\right)}{s_x s_y} \tag{2}$$



(a) Raw data (b) Normalized data

Figure 3: Expression profile of YBR0088x POL30, YER070w RNR1 and GENEX, before and after z-score standardization.

Figure 4 shows the transformed Euclidean distance between genes as the function

$$d_{n_t}\left(r\right)=\sqrt{\frac{r-1}{k_{n_t}}}\ , \tag{3}$$

of their sample linear correlation coefficient r. Here k_{n_t} is a constant that depends on the number of time points n_t. (See Appendix). The more genes are linearly related, the smaller is the Euclidean distance between them after the standardization. Therefore, a tight spherically shaped cluster will contain genes highly linearly related to each other. This means that when the k-means clustering algorithm is used, similarity between two time-series can be understood by the strength of their linear relationship.

In the FCV-TSD algorithm, similarity of expression profiles is not expressed by their strength of linear relationship, but by the form of linear dependency between time points, which is described next. Two time points of a given series are linearly dependent if one is the linear transformation of the other, t_{k+1} is a linear transformation of t_k if $t_{k+1}=bt_k+a$,

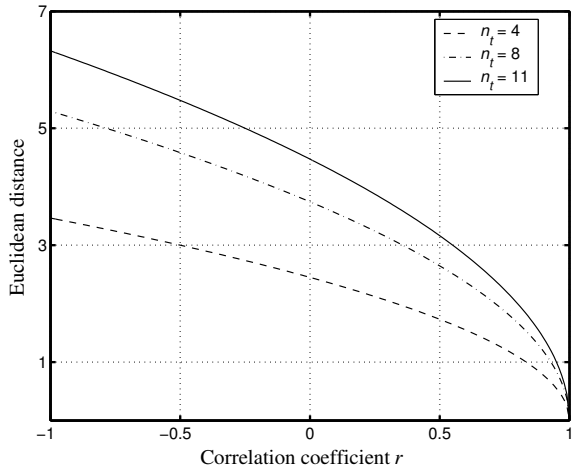


Figure 4: Transformed Euclidean distance between genes as the function of their sample linear correlation coefficient r utilizing different number of time points.

where b and a are the parameters of the transformation describing the linear dependency. Points in an n-dimensional space, ordered in a line configuration, correspond to vectors that share the same form of linear dependency between their time points. Figure 5 shows two linearly shaped groups of points in a two dimensional data-space, where each group has the same transformation parameters among its time points.

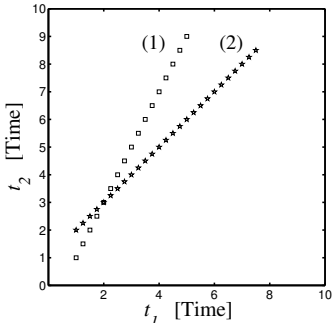


Figure 5: Two linearly shaped groups of points in a two dimensional data-space, where each group has the same transformation parameters among its time points; for (1) $t_2 = 2\,t_1 - 1$, for (2) $t_2 = t_1 + 1$.

The identification of different sets of parameters is necessary to be able to distinguish different sets of shapes of expression profiles in the time domain when all the profiles have the same degree of linear dependency. Linearly shaped groups of points in the data-space are vectors either positively or negatively linearly related depending on the location and

orientation of the group of points. In order to obtain meaningful linearly shaped clusters in the data-space, a preliminary selection of relevant locations and orientations is essential. Hence, we propose the FCV-TSD algorithm to identify such meaningful linear shaped clusters where similarity is related to the form of linear dependency between time points.

3 TSD-FCV algorithm and its implementation

This section presents the TSD and FCV algorithms, and the combination of TSD and FCV forming the FCV-TSD algorithm.

3.1 Transitional State Discrimination (TSD) algorithm

The TSD algorithm groups elements according to the transition of their consecutive time points. The transition is qualified within a range of different states by means of a "pattern vector function" and registered in a "pattern vector" $p_g = [p_{g_k}]$, $1 < k < (n_t - 1)$ where g is the gth gene and n_t is the number of time points. The pattern vector function for sign transition is defined by two states as follows:

$$p_{g_k}\left(x_g\left(t\right)\right) = \begin{cases} 1 & \text{if } x_g\left(t_k\right) - x_g\left(t_{k+1}\right) \leq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

where x_g is the gene expression vector for the gth gene and $x_g(t)$ is the expression of the gene g at time t. Equation (4) evaluates the transition of the gth gene from the time point t_k to the next time point t_{k+1}. The function can be modified in order to cluster particular characteristics of the data set by defining not only states that involve sign change but also changes in relative or absolute magnitudes. Additionally, it could or should be extended to consider the significance of the change in expression level. This can be achieved by methods such as SAM (Tusher et al. 2001) but requires replicates to be available.

If a vector with a finite number of dimensions has n_s possible states for the transition from one time point to the next one, the number of possible state combinations n_c of the transitions across the vector is determined by the dimensionality of the vector n_t and the number of states n_s as:

$$n_c = n_s^{(n_t - 1)}. \tag{5}$$

By having a limited number of combinations it is possible to compare the pattern vector of each gene to every combination and obtain n_c clusters. The aforementioned TSD algorithm is summarized by the pseudo-code in Figure 6.

STEP 1: Initialization
n_g: number of genes
n_t: number of time points
$x_g = \left[x_{g_1}\ x_{g_2 \ldots}\ x_{g_{n_t}}\right]$: gene expression vector for the gth gene where $1 < g < n_g$
n_s : number of defined states for the pattern vector function
$n_c = n_s^{(n_t-1)}$: number of clusters
STEP 2: The pattern vectors
Define the pattern vector function $p_g\left(x_g, t\right)$ with n_s number of states
FOR all the genes $g = 1$ to n_g
FOR all the time points $t = 1$ to n_t
Evaluate the pattern vector function $p_g\left(x_g, t\right)$
END
END
STEP 3: The prototypes
$var = n_s$ % Dynamic variable initialized with n_s
$col\_index = 1$ % Initialize column index
WHILE $var \leq n_c$ % Production of $(n_t - 1)$ column arrays col to obtain n_c row prototypes
FOR $i = 1$ to n_c/var
FOR $j = 0$ to $(n_s - 1)$
$col\_section_j(i) = j$
END
END
$col(col\_index)$ = concatenation of $col\_section_j(\cdot)$ var/n_s times for $1 < j < (n_s - 1)$
$var = var * n_s$
$col\_index = col\_index + 1$
END WHILE
STEP 4: The clusters
FOR all the prototypes $p = 1$ to n_c
FOR all the genes $g = 0$ to n_g
IF the gth pattern vector == prototype p
THEN gene g belongs to the cluster represented by prototype p
END
END
END ALGORITHM

Figure 6: Pseudo code for the TSD algorithm.

Remark 1: Although the number of clusters increases exponentially with the number of time points, for a "high" dimensionality in time a large percentage of the possible combinations do not have any match or are singletons. However, the initial motivation for this algorithm was the fact that for microarray experiments we usually have only few time points.

3.2 Fuzzy c-varieties clustering (FCV) algorithm

Fuzzy clustering partitions data in a way that the transitions between the subsets are gradual rather than immediate. By employing an objective function to measure the desirability of partitions, the method allows objects to belong to several clusters simultaneously with different degrees of membership to each cluster. In the fuzzy c-means clustering (FCM) algorithm (Bezdek 1980), the distance from a data vector to some prototypical object of a cluster is calculated; the choice of the distance measure determines the shape of the clusters. Usually the standard Euclidean norm, which induces spherical clusters, is utilized. The FCV is an extension of the basic FCM that defines the prototypes as r-dimensional linear subspaces of the data-space; this means it allows the prototypes to be r-dimensional linear varieties, i.e., lines ($r = 1$), planes ($r = 2$) or hyperplanes ($2 < r < p$) rather than just points in $\mathbb{R}^p$. The linear variety of dimension r, $0 \leq r \leq p$ through the point $v \in \mathbb{R}^p$, spanned by the linearly independent vectors $\{s_1, s_2, \ldots, s_r\}$ can be denoted as:

$$V_r(v; \{s_i\}) = \{v\} + span(\{s_i\}). \tag{6}$$

In FCV clustering, the linearly independent vectors spanning the variety are the principal r-eigenvectors of the cluster covariance matrix. Based on this, the algorithm can be developed by adding two steps to the iteration process followed by the FCM algorithm. These steps are calculation of the cluster covariance matrices and extraction of the principal r-eigenvectors. Figure 7 shows the iteration steps of the FCM and FCV algorithms. In the FCV algorithm, the distance corresponds to the squared orthogonal distance from a data vector x to V_r when $\{s_i\}$ form an orthonormal basis for their span:

$$d^2(x, V_r) = \|x - v\|^2 - \sum_{j=1}^{r} (\langle x - v, s_j \rangle)^2. \tag{7}$$

Equation (7) describes the Euclidean distance between the r-dimensional variety V_r and a vector x. For $r = 0$ the sum disappears such that the FCV distance function is identical to the FCM distance function. In this application the desired cluster shape is a line; therefore $r = 1$ and the distance is the shortest, perpendicular, distance from a point x to the line $L(v, s)$. Three user-defined parameters are found in the FCV algorithm;

the number of clusters n_c, the threshold of membership to form the clusters α, and the weighting exponent w. The third parameter is related to the fuzziness of the clustering results, a value of one will produce hard clusters and the larger the value of w the fuzzier the clusters become.

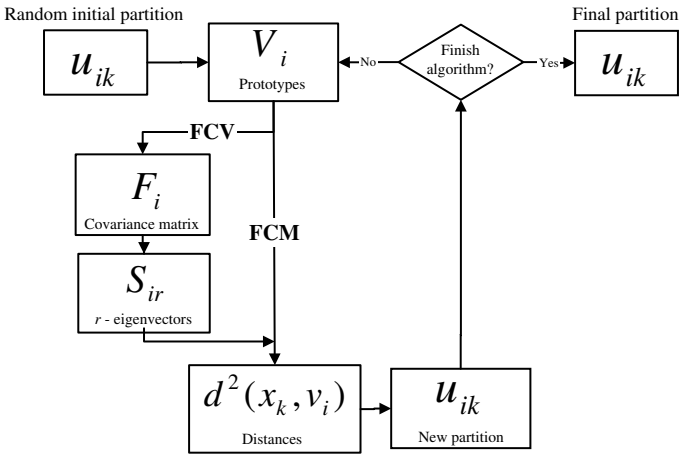


Figure 7: Diagram of the iteration procedure for the FCV and FCM clustering algorithms. Considering the partition of a set $X = [x_1, x_2, \ldots, x_g]$, into c $(2 \leq c < g)$ clusters, the fuzzy clustering partition is represented by a matrix $U = [u_{ik}]$, whose elements are the values of the membership degree of the object x_k to the cluster i, $u_i(x_k) = u_{ik}$. The FCV can be obtained by adding two steps to the basic iteration steps of the FCM algorithm.

3.3 FCV with TSD pre-clustering (FCV-TSD) algorithm

The first step of the FCV-TSD algorithm is TSD clustering where the number of clusters is intrinsic to the data set. By employing the FCV, several clusters within a particular TSD cluster are obtained, which correspond to specific modifications of the original pattern identified by the TSD algorithm. The structure of the FCV-TSD is illustrated in Figure 8. The algorithm retrieves a map where main similitudes and differences between TSD clusters are given by definition, allowing simple connections and relations between clusters. In addition, based on the cluster in which a gene appears and the definition of the pattern vector function, general characteristics of that gene expression can be revealed at once. All algorithms were implemented using MATLAB® (registered trademark by The MathWorks, Inc). The TSD and FCV clustering algorithms implemented in MATLAB are available from http://systemsbiology.umist.ac.uk/.

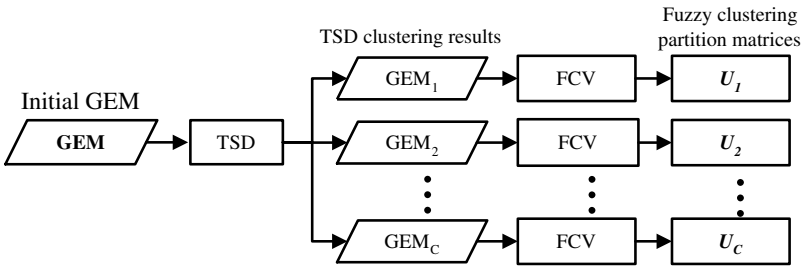


Figure 8: Diagram representing the structure of the FCV-TSD clustering algorithm. The gene expression matrix (GEM) is clustered by the TSD algorithm retrieving c clusters. These clusters are then utilized as input matrices for the c independent FCV clusterings. The fuzzy clustering partitions are represented by the set of matrices U_i with $1 \geq i \leq c$.

4 Comparative studies

This section validates the proposed algorithm using both artificial and real experimental data sets. The performance of the algorithm is compared to k-means and random clustering (Yeung et al. 2001). The latter method is a random grouping of the data into a predefined number of clusters, the results from this clustering algorithm will function as a control in the comparison. The quality of the clustering results produced by the three methods is compared and evaluated using two criteria. The first is the coefficient R defined in (8) where $r_s(g_i, g_j)$ is the Spearman rank-order correlation coefficient (Winkler and Hays 1975) between gene i and gene j, and n_g is the number of genes:

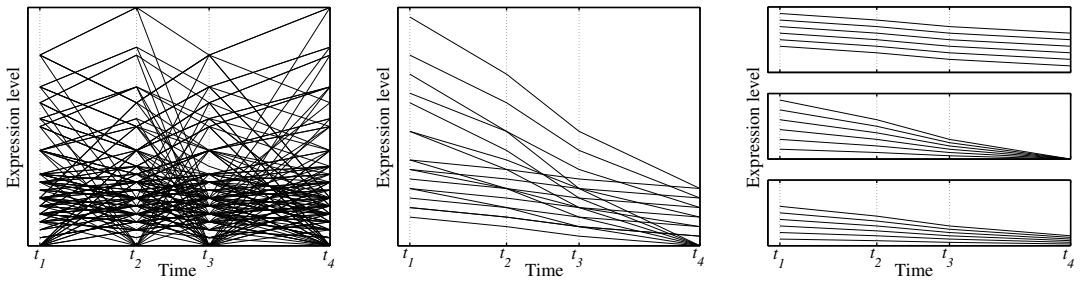
$$R = \frac{1}{n_g^2} \sum_{i=1,j=1}^{n_g} r_s(g_i, g_j) \tag{8}$$

The Spearman rank-order correlation coefficient r_s is here used to measure the time ordered relationship among genes. It is a nonparametric correlation obtained by calculating the Pearson correlation (Maurice and Kendall 1961) of the ranks of the data. The ranking eliminates the influence of extreme variations in expression levels over the control of the correlation. Therefore, the correlation is only controlled by the order of the data, not by the level. To rank the data, the lowest measurement of the gene expression profile becomes one, the second lowest two, and so forth. The second criteria, $\sqrt{\lambda_2}$, is related to the geometry of the cluster where λ_2 refers to the second largest eigenvalue of the covariance matrix of the clusters. The eigenvectors and eigenvalues of the cluster covariance matrix provide

information about the shape and orientation of the cluster (Bezdek 1981, Babuska 1998). The ratio of the lengths of hyperellipsoid axes in a cluster is given by the ratio of the square roots of the eigenvalues of the covariance matrix, and the directions are given by the eigenvectors. In this study the target cluster shape is a line, therefore the root of the second largest eigenvalue $\sqrt{\lambda_2}$ of the cluster covariance matrix should be as small as possible since $\sqrt{\lambda_2} \simeq 0$ for a linearly shaped cluster.

4.1 Validation based on artificial data

To illustrate and compare the performance of the proposed algorithm, a simple example of a four time-point artificial data set is used in this section. The data set is constructed out of eight different vectors that represent all possible combinations of sign transitions for a four time-point vector. Each vector is linearly transformed using three sets of transformation parameters, resulting in three different patterns for each original vector and a total of 24 clusters as shown in Figure 9. The data set is clustered with k-means, random and FCV-TSD clustering algorithms. The quality of the clusters is evaluated using the coefficient R and the value of $\sqrt{\lambda_2}$. The results are summarized in Table 1 and Table 2, respectively.



(a) Artificial data set

(b) A particular sign transition combination of (a)

(c) Three different sets of shapes of expression profiles of the sign transition combination in (b)

Figure 9: Artificial data set with 24 sets of shapes of expression profiles within eight sign transition combinations.

The FCV-TSD algorithm distinguishes the 24 original clusters as shown in Figure 10. The TSD algorithm groups the data into the eight possible different sign transitions using the pattern vector function defined in (4), then the FCV distinguishes the three different lines formed by the three different linear transformations. The k-means algorithm clusters

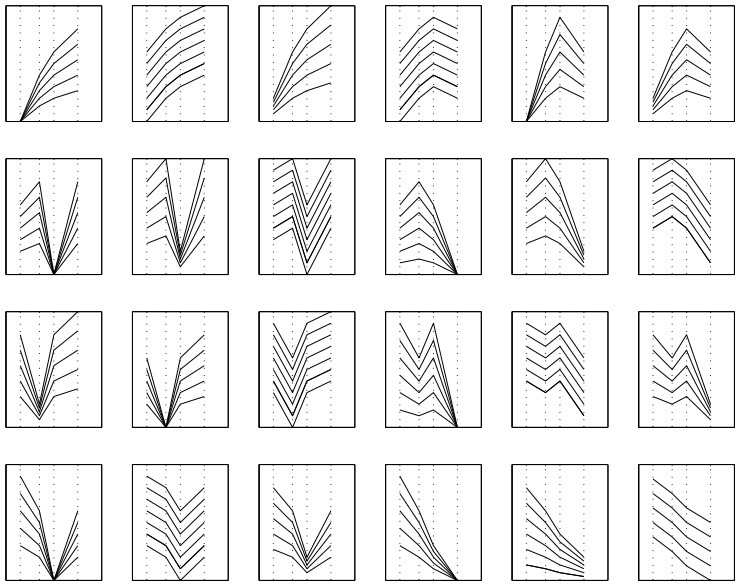


Figure 10: Results of clustering the artificial data set using the FCV-TSD algorithm. In each figure the horizontal axis denotes time and the vertical axis denotes the expression level.

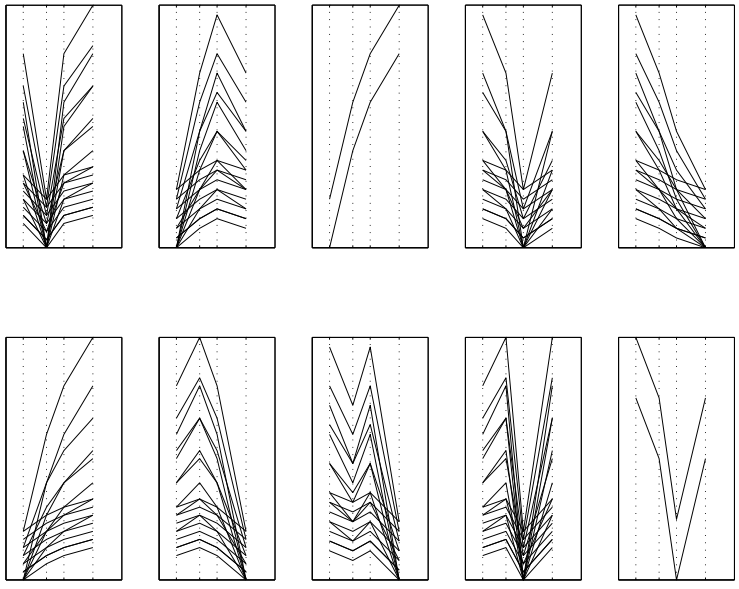


Figure 11: Results of clustering the artificial data set using the k-means algorithm. In each figure the horizontal axis denotes time and the vertical axis denotes the expression level.

the data set into ten clusters as shown in Figure 11. The eight possible different sign transitions are identified without distinguishing the form of the linear transformation and two original shapes are split into two clusters. The first observation is related to the z-score standardization of the gene expression matrix. It transforms all the vectors into the corresponding original eight different vectors and as a consequence, the k-means algorithm is performed on a set of only eight different well separated groups with identical elements forming each group. The second observation is related to the design of the k-means algorithm. The elements are moved to the cluster whose center is closest to them in an iterative manner. The termination occurs either when the centroids of the clusters move less than a predefined threshold or when the predefined number of iterations is achieved. Since several elements are identical, they can move randomly among identical clusters without changing the centroids, and as a consequence the algorithm terminates after the first iteration.

Both k-means and FCV-TSD clustering methods produce clusters with perfect Spearman rank-order correlation between the constituting elements of each cluster as shown in Table 1, both algorithms separate the original eight vectors with their corresponding linear transformations in different clusters. In contrast, the random clustering shows no meaningful internal correlation.

Table 1: Summary of the R values for k-means, random, and FCV-TSD clustering.

	k-means	random	FCV-TSD
median	1	0.18	1
mean	1	0.24	1
standard deviation (s.d.)	0	0.21	0
coefficient of variation (s.d./mean)	0	0.90	0

As expected from its fundamental idea, the FCV-TSD is the unique method which identifies the different lines formed in the data-space. As shown in Table 2, the $\sqrt{\lambda_2}$ for all the FCV-TSD clusters is zero, which indicates the cluster is linearly shaped.

Table 2: Square root of the second largest eigenvalue $\sqrt{\lambda_2}$ of the cluster covariance matrix for k-means, random, and FCV-TSD clustering.

	k-means	random	FCV-TSD
median	2.14	4.34	0
mean	2.73	4.30	0
standard deviation (s.d.)	0.92	2.61	0
coefficient of variation (s.d./mean)	0.53	0.61	0

4.2 Validation based on experimental data: *Saccharomyces cerevisiae* data set

In this section the FCV-TSD algorithm is validated based on the Mitotic cell cycle of *Saccharomyces cerevisiae* data gathered by Cho et al. (1998). The data set is available from http://genomics.stanford.edu. It shows the change of abundance of 6220 mRNA species in synchronized *Saccharomyces cerevisiae* over two cell cycles. As stated by Cho et al. (1998), to obtain synchronous yeast culture, cdc28-13 cells were arrested in late G1 at START by raising the temperature to 37, and the cell cycle was reinitiated by shifting cells to 25. Cells were collected at 17 time points taken at 10 min intervals. We utilize the first four time points which contain temperature-induced effects to produce a short time-series data set.

As with the artificial data set, k-means, random and FCV-TSD algorithms are used to cluster the GEM. The methods for each approach are described in Section 4.2.1. The quality of the clusters is evaluated using the coefficient R and the value of $\sqrt{\lambda_2}$ as that for the artificial data set. The results are summarized in Section 4.2.2. Detailed descriptions of these clusters can be found in http://systemsbiology.umist.ac.uk/.

Remark 2: Since the number of biological clusters is not known *a priori*, there is no previous argument indicating how many clusters should be considered. In this study the number is set to 40 by considering an average size of 55 genes per cluster. Although validity indices for optimal number of clusters should be investigated further for a better clustering performance, note that the objective of this test is not to obtain the optimal clustering results but to understand and compare the performance of each algorithm. The same is true with the FCV parameters since they are not tuned for optimal performance.

4.2.1 Methods

For the k-means algorithm, the original GEM is conducted through three main stages as proposed by Tavazoie et al. (1999). First, the original data is filtered using σ/μ as a metric of variation leaving 2236 genes. Next, the gene expression profiles are z-score standardized and finally, the GEM is clustered with the k-means algorithm. For the FCV-TSD, the original GEM is filtered within the TSD algorithm by means of the pattern vector definition presented in (9), where the $Null$ value is considered as an invalid state which flags the genes for further filtering in the fourth step of the algorithm. That is, if the gth pattern vector contains at least one $Null$ value, the gth gene is not considered for the clustering analysis. The value of β is adjusted to get the same number of genes as with the σ/μ filtering. Next, the resultant n_c clusters from the TSD algorithm are used as the input matrices for the n_c independent FCV clusterings. As previously stated, the clustering parameters are not tuned for optimal performance and for ease of evaluation the parameters α and w are kept constant with $\alpha = 0.75$ and $w = 1.5$ for all the FCV clusterings. As in the k-means approach, the total number of clusters is set to 40.

$$p_{g_k}(x_g(t)) = \begin{cases} 1 & \text{if} \quad x_g(t_k) - x_g(t_{(k+1)}) < 0 \quad \text{and} \\ & \quad |(x_g(t_{(k+1)}) - x_g(t_k))/x_g(t_{(k+1)})| > \beta, \\ 0 & \text{else if} \quad x_g(t_k) - x_g(t_{(k+1)}) > 0 \quad \text{and} \\ & \quad |(x_g(t_{(k+1)}) - x_g(t_k))/x_g(t_k)| > \beta, \\ Null & \text{otherwise.} \end{cases} \tag{9}$$

4.2.2 Results

Table 3 presents the summary for the coefficient R, defined in (8). The FCV-TSD presents lower mean, median and coefficient of variation of the coefficient R than the k-means and random clustering. It shows that the FCV-TSD algorithm does produce clusters with higher correlation between their constituting elements than the k-means algorithm.

The difference in $\sqrt{\lambda_2}$ from the k-means and FCV-TSD results using a real data set is not so evident compared with that of the artificial data set. Although the mean and median values of $\sqrt{\lambda_2}$ for the FCV-TSD clusters are lower than the respective values from the k-means clusters, the mean difference is very small and the FCV-TSD results

Table 3: Summary of the R values for k-means, random, and FCV-TSD clustering.

	k-means	random	FCV-TSD
median	0.904	0.039	0.926
mean	0.883	0.040	0.928
standard deviation (s.d.)	0.092	0.012	0.068
coefficient of variation (s.d./mean)	0.104	0.289	0.073

present a high coefficient of variation (s.d./mean), as presented in Table 4. However, it must be noted that half of the clusters from the FCV-TSD have a value of $\sqrt{\lambda_2}$ lower than 0.375 while less than half of the clusters from the k-means have a value lower than 0.375. The difference between the mean and median of $\sqrt{\lambda_2}$ from the FCV-TSD clusters indicates the presence of outliers. These correspond to clusters where the fixed clustering parameters are not favorable. The $\sqrt{\lambda_2}$ values of the resultant clusters from the FCV-TSD and k-means algorithm would show significant difference if the FCV-TSD was tuned for optimal performance. Nevertheless, the FCV-TSD with arbitrary clustering parameters has already shown a better performance.

Table 4: Square root of the second largest eigenvalue $\sqrt{\lambda_2}$ of the cluster covariance matrix, for k-means, random, and FCV-TSD clustering.

	k-means	random	FCV-TSD
median	0.448	2.015	0.375
mean	0.584	1.989	0.551
standard deviation (s.d.)	0.400	0.134	0.538
coefficient of variation (s.d./mean)	0.684	0.067	0.977

5 Conclusions

The FCV-TSD clustering algorithm was presented as a new clustering method for short time-series gene expression data that is able to characterize temporal relations in the clustering environment. This has not been achieved by other traditional algorithms such as k-means. We introduced the main concept of the proposed algorithm by addressing the issue of similarity of time-series gene expression. Although validating clusterings is a difficult task (Azuaje 2002), suitable parameters of evaluation can be used when the

clustering objectives are well established. We presented a simple clustering example with artificial data set and showed the advantages of the proposed algorithms over the k-means clustering algorithm. In addition, the algorithm was validated on a subset of the Mitotic cell cycle of *Saccharomyces cerevisiae* data gathered by Cho et al. (1998). The k-means algorithm and random clustering were used for comparison. The performance was evaluated with the internal cluster correlation using the Spearman rank-order correlation coefficient, and with the geometrical properties of the clusters. The TSD-FCV algorithm showed better performance than the k-means algorithm in both artificial and real data sets.

6 Acknowledgements

This work was supported in part by grants from ABB Ltd. U.K., an Overseas Research Studentship (ORS) award, Consejo Nacional de Ciencia y Tecnologia (CONACYT), and by the Post-doctoral Fellowship Program of Korea Science & Engineering Foundation (KOSEF).

Appendix

Equation (3) is obtained by fitting a quadratic function to the Euclidean distance d between standardized genes and their sample correlation coefficient r, such that $r = -k_{n_t}\, d^2 + 1$, where k_{n_t} is dependant on the number of time points n_t. In order to obtain k_{n_t} as a function of n_t, a linear regression of $\ln(n_t)$ and $\ln(k_{n_t})$ can be calculated, $\ln(k_{n_t}) = b\, \ln(n_t) + a$, such that $k_{n_t} = n^b\, e^a$.

References

Anderson, T.: 1958, *The Statistical Analysis of Time Series*, Wiley.

Azuaje, F.: 2002, A cluster validity framework for genome expression data, *Bioinformatics* **18**(2), 319–20.

Babuska, R.: 1998, *Fuzzy Modeling for Control*, Kluwer Academic Publishers.

Bezdek, J.: 1980, A convergence theorem for the fuzzy isodata clustering algorithms, *IEEE Trans. Pattern Anal. Machine Intell.* **2**(1), 1–8.

Bezdek, J.: 1981, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press.

Bloomfield, P.: 1976, *Fourier Analuss of Time Series: An Introduction*, New York: Wiley.

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

Eisen, M., Spellman, P., Brown, P. and Botstein, D.: 1998, Cluster analysis and display of genome-wide expression patterns, *Proc. Natl. Acad. Sci.* **95**(1), 14863–68.

Everitt, B.: 1974, *Cluster Analysis*, Heinemann Educational Books.

Geva, A. B. and Kerem, D. H.: 1988, Brain state identification and forecasting of acute pathology using unsupervised fuzzy clustering of EEG temporal patterns, *in* H. Teodorescu, A. Kendel and L. Jain (eds), *Fuzzy and Neuro-Fuzzy Systems in Medicine*, CRC Press, pp. 57–93.

Jain, A. K. and Dubes, R. C.: 1988, *Algorithms for Clustering Data*, Prentice Hall.

Kruglyak, S. and Tang, H.: 2001, A new estimator of significance of correlation in time series data, *Journal of Computational Biology* **8**(5), 463–70.

Maurice, G. and Kendall, M.: 1961, *The Advanced Theory of Statistics*, Vol. 2, Charles Griffin and Company Limited.

Mitchell, M. and Mulherin, J.: 1996, The impact of industry shocks on takeover and restructuring activity, *Journal of Financial Economics* **41**(2), 193–229.

Oates, T.: 1999, Identifying distinctive subsequences in multivariate time series by clustering, *in* S. Chaudhuri and D. Madigan (eds), *Fifth International Conference on Knowledge Discovery and Data Mining*, ACM Press, pp. 222–26.

Sankoff, D. and Kruskal, J.: 1983, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, Addison Wesley.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. and Golub, T.: 1999, Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation, *Proc. Natl. Acad. Sci.* **96**, 2907–12.

Tavazoie, S., Hughes, J., Campbell, M., Cho, R. and Church, G.: 1999, Systematic determination of genetic network architecture, *Nat. Genet.* **22**, 281–85.

Tran, D. and Wagner, M.: 2002, A fuzzy approach to speaker verification, *International Journal of Pattern Recognition and Artificial Intelligence* **16**(7), 913–25.

Tusher, V., Tibshirani, R. and Chu, G.: 2001, Significance analysis of microarrays applied to the ionizing radiation response, *PNAS* **98**(9), 5116–21.

Winkler, R. and Hays, W.: 1975, *Statistics: Probability, Inference and Decision*, New York: Holt, Rinehart and Winston.

Yeung, K., Haynor, D. R. and Ruzzo, W. L.: 2001, Validating clustering for gene expression data, *Bioinformatics* **17**(4), 309–318.