Microarray data clustering based on temporal variation: FCV with TSD preclustering

Carla S Möller-Levet, Kwang-Hyun Cho, 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 characterise temporal relations in the clustering environment (ie 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 datasets, where k-means and random clustering are used for comparison. The performance was evaluated with a measure for internal cluster correlation and the geometrical properties of the clusters, showing that the FCV-TSD algorithm had better performance than the k-means algorithm on both datasets.

Keywords: gene expression data, short time-series, transitional state discrimination algorithm, fuzzy c-varieties clustering, Saccharomyces cerevisiae microarray data

Introduction

A natural and intuitive approach for visualising 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 analysed using such a wide range of clustering algorithms. Hierarchical clustering (Eisen et al 1998), self-organising maps (Tamayo et al 1999) and the 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 (ie time course versus 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 (Oats 1999; Tran and Wagner 2002) and medicine (Geva and Kerem 1998). Frequency analysis (Bloomfield 1976) and time-warping algorithms (Sankoff and Kruskal 1983) are analysis techniques commonly used in these fields. In gene expression research it is not always possible to obtain the required sample size to make sense of these techniques. 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. While time-series expression profiles can mathematically be treated as row vectors and thus be clustered by any algorithm that compares and groups genes as points in the data-space, here we emphasise 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 characterise temporal relations in the clustering environment (ie 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, called fuzzy $c$-varieties (FCV) clustering with transitional state discrimination (TSD) preclustering (referred to as the FCV-TSD algorithm hereafter), is a two-step approach: first the algorithm (described later) groups the points in relevant locations and orientations, and then the FCV algorithm (Bezdek 1981) looks for linear-shaped clusters within each particular group.

This paper is organised as follows. First, we address the concept of similarity for time-series and introduce the main idea of the FCV-TSD algorithm. The objectives and basic concepts of the FCV and TSD algorithms are then presented and followed by the description of their use in the FCV-TSD algorithm. The comparative studies section presents the validation of the algorithm with synthetic and real experimental datasets, 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 summarising the presented research.

**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 analysed and a more comprehensive concept of similarity is proposed as a basis for the FCV-TSD algorithm.

The collection of points that form groups in the data-space can have different shapes, such as the spherical- and linear-shaped clusters shown in Figure 2a. 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 $R^2$, spheres in $R^3$ or hyperspheres in $R^n$. By preferring these shapes, the algorithm clusters expression profiles with similar absolute expression levels without considering the shape of the expression profile between dimensions (ie time points). This is illustrated in Figure 2b, which shows the time-series for the spherical-shaped cluster of Figure 2a. 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 standardisation of the time-series to $z$-scores, ie the gene expression profiles are scaled to zero mean and unit standard deviation (Tamayo et al 1999; Tavazoie et al 1999). The $z$-score of the $ith$ time point of a gene $x$ is defined in equation (1), where $\bar{x}$ is the mean and $s_x$ the standard deviation of all the time points $x_1,...,x_n$ in vector $x$:

$$z_i = \frac{x_i - \bar{x}}{s_x}$$

(1)

To visualise the effects in the time domain of this standardisation, consider the following example. The microarray analysis of *Saccharomyces cerevisiae* by Cho et al (1998) shows that YBR0088w POL30 and YER070w RNRI are two of the nineteen functionally characterised 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 \( \text{POL30} \) to \( \text{RNR1} \) is calculated. The differences are used to create a synthetic gene (\( \text{GENEX} \)) with \( \text{POL30} \) as a reference, such that \( \text{GENEX} \) and \( \text{RNR1} \) have the same Euclidean distance to \( \text{POL30} \) in every time point but in opposite directions. After the z-score standardisation, the Euclidean distances are recalculated and show that \( \text{GENEX} \) is closer to \( \text{POL30} \) than \( \text{RNR1} \). Figure 3 shows that after the standardisation, the difference of the absolute expression level of genes with a similar shape of expression profile is neglected and original distance relationships over time are transformed. The distance relationships after standardisation are related to the strength of the linear relationship between genes. The strength of linear relationships between variables can be measured by the sample linear correlation coefficient, \( r \), as defined by equation (2), where \( n \) is number of pairs of observations, \( \bar{x} \) is the average and \( s_x \) is the standard deviation of the vector \( x \), and \( \bar{y} \) is the average and \( s_y \) is the standard deviation of the vector \( y \) (Kendall and Stuart 1961).

\[
r(x, y) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})/(n-1)}{s_x s_y}
\]

(2)

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

\[
d_{d_{1/r}}(r) = \sqrt{\frac{r-1}{k_{d_{1/r}}}}
\]

(3)
of their sample linear correlation coefficient \( r \). Here, \( k_{nt} \) is a constant that depends on the number of time points \( nt \). The more genes are linearly related, the smaller the Euclidean distance is between them after the standardisation. Therefore, a tight, spherical-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 \( t_{k+1} = bt_k + a \), 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 linear-shaped groups of points in a two-dimensional data-space, where each group has the same transformation parameters among its time points.

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. Linear-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. To obtain meaningful linear-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 these meaningful linear-shaped clusters where similarity is related to the form of linear dependency between time points.

**FCV-TSD algorithm and its implementation**

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

**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 is registered in a 'pattern vector', \( P_g = \{P_{gk}\} \), \( 1 < k < (nt-1) \) where \( g \) is the \( g \)th gene and \( nt \) is the number of time points. The pattern vector function for sign transition is defined by two states as follows:

\[
P_{gk}(x_g(t)) = \begin{cases} 
1 \text{ if } x_g(t_k) - x_g(t_{k+1}) \leq 0, \\
0 \text{ otherwise} 
\end{cases} 
\]  

(4)

where \( x_g \) is the gene expression vector for the \( g \)th gene and \( x_g(t) \) is the expression of gene \( g \) at time \( t \). Equation (4) evaluates the transition of the \( g \)th gene from the time point \( t_k \)
to the next time point $t_{k+1}$. The function can be modified to cluster particular characteristics of the dataset 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:

$$n_c = n_s^{(n_t-1)}$$  \hfill (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 summarised by the pseudo-code in Figure 6.

---

**Figure 6** Pseudo-code for the TSD algorithm.

```plaintext
STEP 1: Initialisation

$n_g$: number of genes

$n_t$: number of time points

$x_g = [x_{g1}, x_{g2}, \ldots, x_{gnt}]$: gene expression vector for the $g$th 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 $p_g (x_g(t))$ 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 $p_g (x_g(t))$

END

END

STEP 3: The prototypes

$var = n_s$: % Dynamic variable initialised with $n_s$

$col_index = 1$: % Initialise 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 $(n_s - 1)$

$col_sectionj(i) = j$

END

END

$col(col_index) = \text{concatenation of } col_section(\cdot) \text{ var}/n_s$ 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 $g$th pattern vector == prototype $p$

THEN gene $g$ belongs to the cluster represented by prototype $p$

END

END

END ALGORITHM
```
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 (Bedzek 1980; Wolkenhauer 2001), 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 utilised. The FCM 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, ie lines ($r = 1$), planes ($r = 2$) or hyperplanes ($2 < r < p$), rather than just points in $R^p$. The linear variety of dimension $r$, $0 \leq r \leq p$, through the point $v \in R^p$, spanned by the linearly independent vectors $\{s_1, s_2, \ldots, s_r\}$ can be denoted as:

$$V_r(v;\{s_i\}) = \{v\} + \text{span}(\{s_i\})$$

(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 the calculation of the cluster covariance matrices and extraction of the principal $r$-eigenvectors.

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 $L(v,s)$. The second user-defined parameters are found in the FCV algorithm: the number of clusters $n_c$, the threshold of membership to form the clusters $\alpha$, 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.

FCV with TSD preclustering (FCV-TSD) algorithm

The first step of the FCV-TSD algorithm is TSD clustering where the number of clusters is intrinsic to the dataset. 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 immediately. All algorithms were implemented using MATLAB® (registered trademark of The MathWorks, Inc). The TSD and FCV clustering algorithms implemented in MATLAB® are available from http://www.sbi.uni-rostock.de/tsd-paper/tsd-paper.htm.

Comparative studies

This section validates the proposed algorithm using both artificial and real experimental datasets. 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 equation (8), where $r_{ij}(g_i, g_j)$ 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:

$$R = \frac{1}{n_g^2} \sum_{i,j=1}^{n_g} r_{ij}(g_i, g_j)$$  \hspace{1cm} (8)

The Spearman rank-order correlation coefficient $r_{ij}$ is here used to measure the time-ordered relationship among genes. It is a nonparametric correlation obtained by calculating the Pearson correlation (Kendall and Stuart 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 $\lambda_2$ 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 $\sqrt{\lambda_2} = 0$ for a linear-shaped cluster.

Validation based on artificial data

To illustrate and compare the performance of the proposed algorithm, a simple example of a four time-point artificial dataset is used in this section. The dataset 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 dataset 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 summarised in Table 1 and Table 2, respectively.

### Table 1 Summary of the $R$ values for $k$-means, random and FCV-TSD clustering of artificial data

<table>
<thead>
<tr>
<th></th>
<th>$k$-means</th>
<th>Random</th>
<th>FCV-TSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>median</td>
<td>1</td>
<td>0.18</td>
<td>1</td>
</tr>
<tr>
<td>mean</td>
<td>1</td>
<td>0.24</td>
<td>1</td>
</tr>
<tr>
<td>standard deviation ($s$)</td>
<td>0</td>
<td>0.21</td>
<td>0</td>
</tr>
<tr>
<td>coefficient of variation ($s$/mean)</td>
<td>0</td>
<td>0.90</td>
<td>0</td>
</tr>
</tbody>
</table>

### 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 of artificial data

<table>
<thead>
<tr>
<th></th>
<th>$k$-means</th>
<th>Random</th>
<th>FCV-TSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>median</td>
<td>2.14</td>
<td>4.34</td>
<td>0</td>
</tr>
<tr>
<td>mean</td>
<td>2.73</td>
<td>4.30</td>
<td>0</td>
</tr>
<tr>
<td>standard deviation ($s$)</td>
<td>0.92</td>
<td>2.61</td>
<td>0</td>
</tr>
<tr>
<td>coefficient of variation ($s$/mean)</td>
<td>0.53</td>
<td>0.61</td>
<td>0</td>
</tr>
</tbody>
</table>
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 equation (4), then the FCV distinguishes the three different lines formed by the three different linear transformations. The k-means algorithm clusters the dataset 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 standardisation 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 in an iterative manner to the cluster whose centre is closest to them. The termination occurs either when the centroids of the clusters move less than a predefined threshold or when the predefined number of iterations are 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 constituent 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.

As expected from its fundamental idea, the FCV-TSD is a 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 linear-shaped.

**Validation based on experimental data:**

**Saccharomyces cerevisiae dataset**

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 dataset is available from http://171.65.26.52/yeast_cell_cycle/cellcycle.html. It shows the change in abundance of 6220 mRNA species in synchronised *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 °C, and the cell cycle was re-initiated by shifting the cells to 25 °C. Cells were collected at 17 time points taken at 10 minute intervals. We utilise the first four time points, which contain temperature-induced effects, to produce a short time-series dataset.

As with the artificial dataset, k-means, random and FCV-TSD algorithms are used to cluster the GEM. The methods for each approach are described below. The quality of the clusters is evaluated using the coefficient $R$ and the value of $\sqrt{\lambda_2}$, as for the artificial dataset. The results are summarised in the results section below. Detailed descriptions of these clusters can be found at http://www.sbi.uni-rostock.de/tsd-paper/tsd-paper.htm.

**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 $\sigma/\mu$ as a metric of variation leaving 2236 genes. Next, the gene expression profiles are $z$-score standardised 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 equation (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 $g$th pattern vector contains at least one Null value, the $g$th gene is not considered for the clustering analysis. The value of $\beta$ is adjusted to get the same number of genes as with the $\sigma/\mu$ filtering. Next, the resultant $nc$ clusters from the TSD algorithm are used as the input matrices for the $nc$ independent FCV clusterings. As previously stated, the clustering parameters are not tuned for optimal performance, and for ease of evaluation the parameters $\alpha$ 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.

\[
p_n(x_g(t)) = \begin{cases} 
1 & \text{if } x_g(t) - x_g(t_{(t_{\text{null}})}) < 0 \text{ and } \\
0 & \text{else if } x_g(t) - x_g(t_{(t_{\text{null}})}) > 0 \text{ and } \\
\left|\frac{x_g(t_{(t_{\text{null}}})} - x_g(t)}{x_g(t) - x_g(t_{(t_{\text{null}})})}\right| > \beta, \\
\text{Null otherwise}
\end{cases}
\]  

(9)

**Results**

Table 3 presents the summary for the coefficient $R$, defined in equation (8). The FCV-TSD presents a 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 a higher correlation between their constituent elements than the $k$-means algorithm.

<table>
<thead>
<tr>
<th>Table 3 Summary of the $R$ values for $k$-means, random and FCV-TSD clustering of experimental data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$-means</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>median</td>
</tr>
<tr>
<td>mean</td>
</tr>
<tr>
<td>standard deviation ($s$)</td>
</tr>
<tr>
<td>coefficient of variation ($s$/mean)</td>
</tr>
</tbody>
</table>
The difference in $\sqrt{\lambda_2}$ between the $k$-means and FCV-TSD results using a real dataset is not so evident compared with that of the artificial dataset. 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 present a high coefficient of variation (standard deviation divided by 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 favourable. The $\sqrt{\lambda_2}$ values of the resultant clusters from the FCV-TSD and $k$-means algorithms 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.

Conclusions
The FCV-TSD clustering algorithm was presented as a new clustering method for short time-series gene expression data which is able to characterise 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 an artificial dataset 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 FCV-TSD algorithm showed better performance than the $k$-means algorithm in both artificial and real datasets.

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

Notes
1 Equation (3) is obtained by fitting a quadratic function to the Euclidean distance $d$ between standardised genes and their sample correlation coefficient $r$, such that $r = -k \cdot d^2 + 1$, where $k$ is dependent on the number of time points $n_t$. To obtain $k$ as a function of $n_t$, a linear regression of $\ln(n_t)$ and $\ln(k)$ can be calculated, $\ln(k) = b \ln(n_t) + a$, such that $k = e^a$.
2 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 a few time points.
3 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 the 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.

References
Clustering microarray time-series data