# Fuzzy Clustering of Short Time-Series and Unevenly Distributed Sampling Points

Carla S. Möller-Levet[1], Frank Klawonn[2], Kwang-Hyun Cho[3], and Olaf Wolkenhauer[4]

[1] Control Systems Centre, Department of Electrical Engineering and Electronics, UMIST, Manchester, U.K.
C.Moller-Levet@postgrad.umist.ac.uk
[2] Department of Computer Science, University of Applied Sciences, D-38302 Wolfenbüttel, Germany.
F.Klawonn@FH-Wolfenbuettel.DE
[3] School of Electrical Engineering, University of Ulsan, Ulsan, 680-749, Korea.
ckh@mail.ulsan.ac.kr
[4] Department of Computer Science, Albert Einstein Str. 21, 18051 Rostock, Germany.
wolkenhauer@informatik.uni-rostock.de

**Abstract.** This paper proposes a new algorithm in the fuzzy-$c$-means family, which is designed to cluster time-series and is particularly suited for short time-series and those with unevenly spaced sampling points. Short time-series, which do not allow a conventional statistical model, and unevenly sampled time-series appear in many practical situations. The algorithm developed here is motivated by common experiments in molecular biology. Conventional clustering algorithms based on the Euclidean distance or the Pearson correlation coefficient are not able to include the temporal information in the distance metric. The temporal order of the data and the varying length of sampling intervals are important and should be considered in clustering time-series. The proposed short time-series (STS) distance is able to measure similarity of shapes which are formed by the relative change of amplitude and the corresponding temporal information. We develop a fuzzy time-series (FSTS) clustering algorithm by incorporating the STS distance into the standard fuzzy clustering scheme. An example is provided to demonstrate the performance of the proposed algorithm.

## 1 Introduction

Microarrays revolutionize the traditional way of one gene per experiment in molecular biology [1], [2]. With microarray experiments it is possible to measure simultaneously the activity levels for thousands of genes. The appropriate clustering of gene expression data can lead to the classification of diseases, identification of functionally related genes, and network descriptions of gene regulation, among others [3], [4].

Time course measurements are becoming a common type of experiment in the use of microrarrays, [5], [6], [7], [8], [9]. If a process is subject to variations over time, the conventional measures used for describing similarity (e.g. Euclidean distance) will not provide useful information about the similarity of time-series in terms of the cognitive perception of a human [10]. An appropriate clustering algorithm for short time-series should be able to identify similar shapes, which are formed by the relative change of expression as well as the temporal information, regardless of absolute values. The conventional clustering algorithms based on the Euclidean distance or the Pearson correlation coefficient, such as hard $k$-means (KM) or hierarchical clustering (HC) are not able to include temporal information in the distance measurement. Fig. 1 shows three time-series with different shapes to illustrate this point. An appropriate distance for the three expression profiles would identify $g_2$ as more similar to $g_3$ than to $g_1$, since the deviation of shape across time of $g_3$ from the shape of $g_2$ is less than that of $g_1$. That is, the deviation of expression level of $g_1$ from $g_2$ in the transition of the first to the second time point is one unit per one unit of time, while the deviation of expression level of $g_3$ from $g_2$ in the transition of the third to the fourth time point is one unit per seven units of time. The Euclidean distance and the Pearson correlation coefficient do not take into account the temporal order and the length of sampling intervals; for these metrics both $g_1$ and $g_3$ are equally similar to $g_2$. In this paper we introduce a new clustering algorithm which is able to use the temporal information of uneven sampling intervals in time-series data to evaluate the similarity of the shape in the time domain.

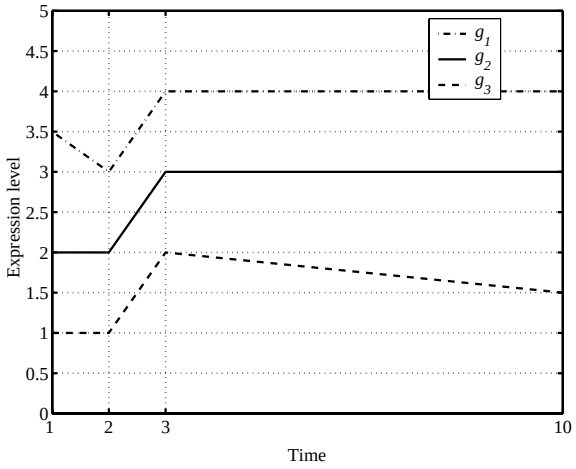



**Fig. 1.** Three unevenly sampled time-series with different shapes

This paper is organized as follows: Section 2 defines the objective and basic concepts of the short time-series (STS) distance based on the requirements of

short time-series clustering. In Section 3, the fuzzy short time-series (FSTS) algorithm is introduced as a modification of the standard fuzzy $c$-means algorithm (FCM). Section 4 presents an artificial data set to illustrate and compare the performance of the proposed algorithm with FCM, KM and single linkage HC. Finally, conclusions are made in Section 5 summarizing the presented research.

## 2 Short Time-Series Distance

This section presents a measure of similarity for microarray time-series data. The performance of the distance is illustrated by means of simple tests for which temporal information is a key aspect.

The objective is to define a distance which is able to capture differences in the shapes, defined by the relative change of expression and the corresponding temporal information, regardless of the difference in absolute values. We approach the problem by considering the time-series as piecewise linear functions and measuring the difference of slopes between them. Considering a gene expression profile $x = [x_0, x_1, \ldots, x_{n_t}]$, where $n_t$ is the number of time points, the linear function $x(t)$ between two successive time points $t_k$ and $t_{(k+1)}$ can be defined as $x(t) = m_k t + b_k$, where $t_k \leq t \leq t_{(k+1)}$, and

$$m_k = \frac{x_{(k+1)} - x_k}{t_{(k+1)} - t_k} \tag{1}$$

$$b_k = \frac{t_{(k+1)} x_k - t_k x_{(k+1)}}{t_{(k+1)} - t_k} \, . \tag{2}$$

The STS distance we propose corresponds to the square root of the sum of the squared differences of the slopes obtained by considering time-series as linear functions between measurements. The STS distance between two time-series $x$ and $v$ is defined as:

$$d^2_{\mathrm{STS}}(x, v) = \sum_{k=0}^{n_t - 1} \left( \frac{v_{(k+1)} - v_k}{t_{(k+1)} - t_k} - \frac{x_{(k+1)} - x_k}{t_{(k+1)} - t_k} \right)^2 . \tag{3}$$

To evaluate the performance of this distance in comparison with the Euclidean distance and the Pearson correlation coefficient, two tests are performed. The objective of the first test is to evaluate the ability to incorporate temporal information into the comparison of shapes. The objective of the second test is to evaluate the ability to compare shapes regardless of the absolute values.

For the first test, let us consider the time-series shown in Fig. 1. Table 1 illustrates the corresponding STS distance, Euclidean distance, and the Pearson correlation coefficient between $g_2$ and $g_1$, and $g_2$ and $g_3$, respectively. The results show that the STS distance is the unique distance metric which reflects the temporal information in the comparison of shapes.

For the second test, let us consider a linear transformation of the absolute values of the time-series shown in Fig. 1. These modified series are shown in

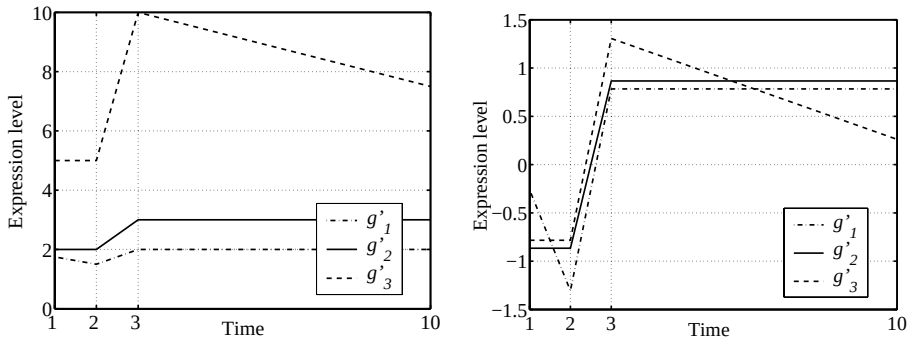
**Table 1.** STS distance, Euclidean distance, and Pearson correlation coefficient between $g_2$ and $g_1$, and $g_2$ and $g_3$ in Fig. 1

| | Euclidean distance | STS distance | Pearson correlation coefficient |
|---|---|---|---|
| $(g_2, g_1)$ | 2.29 | 0.500 | 0.904 |
| $(g_2, g_3)$ | 2.29 | 0.071 | 0.904 |

Fig. 2(a). Since the STS and the Euclidean distance are both sensitive to scaling, a $z$-score standardization of the series is required for them to neglect absolute values [11]. The $z$-score of the $i$th time point of a gene $x$ is defined in (4), where $\overline{x}$ is the mean and $s_x$ the standard deviation of all the time points $x_1, \ldots, x_n$ in vector $x$

$$z_i = \frac{(x_i - \overline{x})}{s_x} \ . \tag{4}$$

The time-series after standardization are shown in Fig. 2(b).




(a) Before standardization (b) After standardization

**Fig. 2.** Three unevenly sampled time-series data with different shapes, which correspond to linear transformations of the time-series in Fig. 1

Table 2 summarizes the STS distance, the Euclidean distance, and the Pearson correlation coefficient between $g'_2$ and $g'_1$, and $g'_2$ and $g'_3$. The results show that the STS distance is the unique distance measure which can properly capture temporal information, regardless of the absolute values.

## 3 Fuzzy Short Time-Series Clustering Algorithm

This section introduces the FSTS clustering algorithm as a new member of the fuzzy $c$-means (FCM) family [12], [13], [14]. We present the minimization of the standard objective function and the resulting cluster prototypes.

**Table 2.** STS distance, the Euclidean distance, and the Pearson correlation coefficient between $g_2'$ and $g_1'$, and $g_2'$ and $g_3'$ in Fig. 2(b)

| | Euclidean distance | STS distance | Pearson correlation coefficient |
|---|---|---|---|
| $(g_2, g_1)$ | 0.756 | 1.103 | 0.904 |
| $(g_2, g_3)$ | 0.756 | 0.386 | 0.904 |

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 [11], [15]. 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. In the previous section we have already established the way in which we desire the "distance" between objects to be measured; hence, in this section, we focus on the way of grouping the objects based upon the measured distance. For this purpose we select a fuzzy clustering scheme, since fuzzy sets have a more realistic approach to address the concept of similarity than classical sets [16], [14]. A classical set has a crisp or hard boundary where the constituting elements have only two possible values of membership, they either belong or not. In contrast, a fuzzy set is a set with fuzzy boundaries where each element is given a degree of membership to each set.

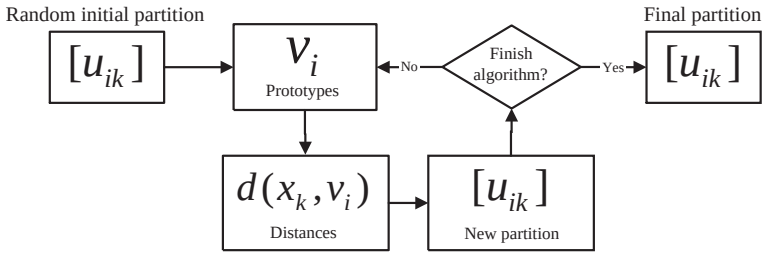



**Fig. 3.** Diagram of the iteration procedure for the FCM clustering algorithms. Considering the partition of a set $X = [x_1, x_2, \ldots, x_{n_g}]$, into $2 \leq n_c < n_g$ clusters, the fuzzy clustering partition is represented by a matrix $U = [u_{ik}]$, whose elements are the values of the membership degree of the object $x_k$ to the cluster $i$, $u_i(x_k) = u_{ik}$

Fuzzy clustering is a partitioning-optimization technique which allows objects to belong to several clusters simultaneously with different degrees of membership to each cluster [12], [13]. The objective function that measures the desirability of partitions is described in (5), where $n_c$ is the number of clusters, $n_g$ is the number of vectors to cluster, $u_{ij}$ is the value of the membership degree of the vector $x_j$ to the cluster $i$, and $d^2(x_j, v_i)$ is the squared distance between the vector $x_j$ and the prototype $v_i$ and $w$ is a parameter (usually set between 1.25

and 2), which determines the degree of overlap of fuzzy clusters.

$$J(x, v, u) = \sum_{i=1}^{n_c} \sum_{j=1}^{n_g} u_{ij}^w \, d^2(x_j, v_i) \; . \tag{5}$$

Fig. 3 illustrates the iteration steps of the FCM algorithm, the representative of the fuzzy clustering algorithms. In order to use the STS distance following the conventional fuzzy clustering scheme, we need to obtain the value of the prototype $v_k$ that minimizes (5), when (3) is used as the distance. Substituting (3) into (5) we obtain

$$J(x, v, u) = \sum_{i=1}^{n_c} \sum_{j=1}^{n_g} u_{ij}^w \sum_{k=0}^{n_t - 1} \left( \frac{v_{i(k+1)} - v_{ik}}{t_{(k+1)} - t_k} - \frac{x_{j(k+1)} - x_{jk}}{t_{(k+1)} - t_k} \right)^2 . \tag{6}$$

The partial derivative of (6) with respect to $v_{ik}$ is:

$$\begin{aligned}
&\frac{\partial J(x, v, u)}{\partial v_{ik}} = \\
&\sum_{j=1}^{n_g} u_{ij}^w \frac{\partial}{\partial v_{ik}} \left( \left( \frac{v_{(k+1)} - v_k}{t_{(k+1)} - t_k} - \frac{x_{(k+1)} - x_k}{t_{(k+1)} - t_k} \right)^2 + \left( \frac{v_k - v_{(k-1)}}{t_k - t_{(k-1)}} - \frac{x_k - x_{(k-1)}}{t_k - t_{(k-1)}} \right)^2 \right) = \\
&\sum_{j=1}^{n_g} u_{ij}^w \left[ \frac{2 \left( v_{ik} - v_{i(k+1)} - x_{jk} + x_{j(k+1)} \right)}{\left( t_k - t_{(k+1)} \right)^2} \right] - \left[ \frac{2 \left( v_{i(k-1)} - v_{ik} - x_{j(k-1)} + x_{jk} \right)}{\left( t_k - t_{(k-1)} \right)^2} \right] = \\
&\qquad \sum_{j=1}^{g} 2 u_{ij}^w \frac{\left( a_k v_{i(k-1)} + b_k v_{ik} + c_k v_{i(k+1)} + d_k x_{j(k-1)} + e_k x_{jk} + f_k x_{j(k+1)} \right)}{\left( t_k - t_{(k+1)} \right)^2 \left( t_k - t_{(k-1)} \right)^2}
\end{aligned} \tag{7}$$

where

$$\begin{aligned}
a_k &= -(t_{(k+1)} - t_k)^2 \qquad & b_k &= -(a_k + c_k) \qquad & c_k &= -(t_k - t_{(k-1)})^2 \\
d_k &= (t_{(k+1)} - t_k)^2 \qquad & e_k &= -(d_k + f_k) \qquad & f_k &= (t_k - t_{(k-1)})^2 \; .
\end{aligned}$$

Setting (7) equal to zero and solving for $v_{ik}$ we have

$$a_k v_{i(k-1)} + b_k v_{ik} + c_k v_{i(k+1)} = - \frac{\sum_{j=1}^{n_g} u_{ij}^w \left( d_k x_{j(k-1)} + e_k x_{jk} + f_k x_{j(k+1)} \right)}{\sum_{j=1}^{n_g} u_{ij}^w}$$

$$a_k v_{i(k-1)} + b_k v_{ik} + c_k v_{i(k+1)} = m_{ik} \tag{8}$$

where

$$m_{ik} = - \frac{\sum_{j=1}^{n_g} u_{ij}^w \left( d_k x_{j(k-1)} + e_k x_{jk} + f_k x_{j(k+1)} \right)}{\sum_{j=1}^{n_g} u_{ij}^w} \; .$$

Equation (8) yields an undetermined system of equations. We know the relations of the prototype values among the time points, but not the absolute value

at each time point. That is, we know the shape but not the absolute level. If we add two fixed time points at the beginning of the series with a value of 0, and solve the system for any $n_t$, the prototypes can be calculated as

$$v(i,n) = \sum_{r=2}^{n-3} m_{ir} \prod_{q=1}^{r-1} c_q \left[ \prod_{q=r+1}^{n-1} a_q + \prod_{q=r+1}^{n-1} c_q + \sum_{p=r+3}^{n} \prod_{j=p-1}^{n-1} c_j \prod_{j=r+1}^{p-2} a_j \right] / \prod_{q=2}^{n-1} c_q + m_{i(n-1)} \prod_{q=1}^{n-2} c_q / \prod_{q=2}^{n-1} c_q + m_{i(n-2)} \prod_{q=1}^{n-3} c_q (a_{(n-1)} + c_{(n-1)}) / \prod_{q=2}^{n-1} c_q \quad (9)$$

where $1 \leq i \leq n_c$, $3 \leq n \leq n_t$ (since $v(i,1) = 0$ and $v(i,2) = 0$), $m_{i1} = 0$ and $c_1 = 1$ .

The same scheme of the iterative process as for the FCM, described in Fig. 3 is followed, but the distance and the prototypes are calculated using (3) and (9), respectively. The same three user-defined parameters found in the FCM algorithm; the number of clusters $n_c$, the threshold of membership to form the clusters $\alpha$, and the weighting exponent $w$ are also found in the proposed FSTS algorithm. Fig. 4 illustrates the pseudocode of the proposed FSTS clustering algorithm.

## 4 Illustrative Example

This section presents a simple artificial data set to illustrate and compare the performance of the proposed FSTS clustering algorithm in terms of the cognitive perception of a human. Four groups of five vectors are created where each group has the same parameters of linear transformation between time points, as shown in Table 3. That is, for the group $i$, $1 \leq i \leq 4$, $x_{j(k+1)} = m_{ik} x_{jk} + b_{ik}$ with $0 \leq k < (n_t - 1)$ and $1 \leq j \leq 5$. The values of $m$ and $b$ were obtained randomly for each group.

**Table 3.** Artificial profile $x = [x_0, x_1, \ldots, x_{n_t}]$. A group of vectors with a similar shape can be obtained by changing the initial value

| Time points | Value |
|---|---|
| $x_0$ | initial value |
| $x_1$ | $m_1 x_0 + b_1$ |
| $x_2$ | $m_2 x_1 + b_2$ |
| $\vdots$ | $\vdots$ |
| $x_{n_t}$ | $m_{(n_t-1)} x_{(n_t-1)} + b_{(n_t-1)}$ |

The resulting artificial data set, shown in Fig. 5(a), was clustered using FCM, FSTS, KM and HC algorithms, respectively. All the algorithms were able to

**STEP 1: Initialization**

| | | |
|---|---|---|
| $n_g$ | : | number of genes |
| $n_t$ | : | number of time points |
| $X$ | : | gene expression matrix (GEM) $[n_g \times n_t]$ |
| $n_c$ | : | number of clusters |
| $w$ | : | fuzzy weighting factor |
| $a$ | : | threshold for membership |
| $\epsilon$ | : | termination tolerance |

**STEP 2: Initialization of the partition matrix**

Initialize the partition matrix randomly, $U^{(0)}$ $[n_c \times n_g]$.

**STEP 3: Repeat for** $l = 1, 2, ...$

3.1 Compute the cluster prototypes:

$v(i,1)^{(l)} = 0$,

$v(i,2)^{(l)} = 0$,

For $v(i,n)^{(l)}$ use Equation (9) $1 \le i \le n_c, \quad 3 \le n \le n_t$.

3.2 Compute the distances:

$$d^2_{\mathrm{STS}}(x_j, v_i) = \sum_{k=0}^{n_t-1} \left( \frac{v^{(l)}_{i(k+1)} - v^{(l)}_{ik}}{t_{(k+1)} - t_k} - \frac{x_{j(k+1)} - x_{jk}}{t_{(k+1)} - t_k} \right)^2$$

$$1 \le i \le n_c, \quad 1 \le j \le n_g.$$

3.3 Update the partition matrix:

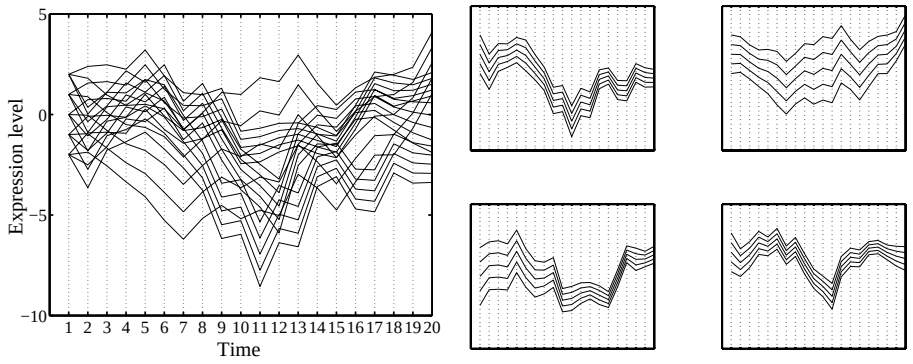
if $d_{\mathrm{STS}ij} > 0$ for $1 \le i \le n_c$, $1 \le j \le n_g$,

$$u^{(l)}_{ij} = \frac{1}{\sum_{q=1}^{n_c} (d_{\mathrm{STS}ij}/d_{\mathrm{STS}qj})^{1/(w-1)}},$$

otherwise $u^{(l)}_{ij} = 0$ if $d_{\mathrm{STS}ij} > 0$, and $u^{(l)}_{ij} \in [0,1]$ with $\sum_{i=1}^{n_c} u^{(l)}_{ij} = 1$.

**Until** $|U^{(l)} - U^{(l-1)}| < \epsilon$.

**Fig. 4.** Pseudo code of the FSTS clustering algorithm.




(a) Artificial Data (b) Clustering results

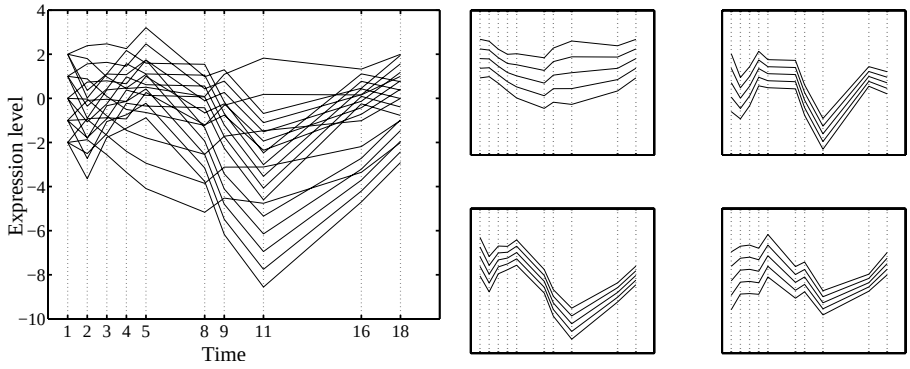
**Fig. 5.** Artificial data set and clustering results for FCM, FSTS, HK and HC algorithms

identify the four clusters shown in Fig. 5(b) successfully. The clustering parameters for both fuzzy algorithms which yield successful results were $w = 1.6$ and $\alpha = 0.4$.

The second test used a subset of the artificial data set shown in Fig. 5(a). The original data set was "resampled" selecting 10 time points randomly out of the 20 original time points. The resulting data set is shown in Fig. 6(a). In this case, only the FSTS algorithm is able to identify the four clusters successfully, while FCM and HC identify two clusters and the other two mixed, as shown in Fig. 7, and KM does not produce consistent results. The clustering parameters for the FSTS algorithm were $w = 1.2$ and $\alpha = 0.6$. Different parameters were tested for the FCM, $1.2 < w < 2.5$ and $0.3 < \alpha < 0.6$ (54 combinations) giving unsuccessful results. Finally the algorithms were evaluated using the three time-series data presented in Section 2. The objective is to cluster $g_1$, $g_2$ and $g_3$ in two clusters. The FSTS algorithm is the unique method capable of grouping $g_2$ with $g_3$ separated from $g_1$ consistently. FCM, HK, and HC do not have consistent results since they find $g_2$ as similar to $g_1$ as to $g_3$ as described in Section 2.

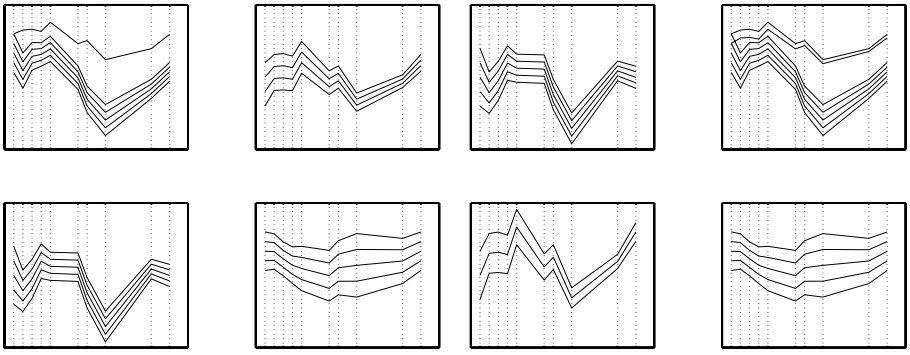
## 5 Conclusions

The FSTS clustering algorithm was presented as a new approach to cluster short time-series. The algorithm is particularly well suited for varying intervals between time points, a situation that occurs in many practical situations, in particular in biology. The FSTS algorithm is able to identify similar shapes formed by the relative change and the temporal information, regardless of the absolute levels. Conventional clustering algorithms, including FCM, KM, or HC, are not able to properly include the temporal information in the distance metric. We tackled the problem by considering the time-series as piecewise linear functions and measuring the difference of slopes between the functions. We illustrated the




(a) Unevenly resampled artificial data (b) Consituting clusters

**Fig. 6.** Unevenly resampled artificial data set and the constituting clusters



(a) FCM clustering results (b) HC clustering results

**Fig. 7.** Clustering results for FCM and HC algorithms

algorithm with an artificial data set. The FSTS algorithm showed better performance than the conventional algorithms in clustering unevenly sampled short time-series data.

## 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).

## References

[1] Brown, P.O., Botstein, D.: Exploring the new world of the genome with DNA microarrays. Nature Genetics supplement **21** (1999) 33–37
[2] Duggan, D.J., Bittner, M., Chen, Y., Meltzer, P., Trent, J.M.: Expression profiling using cDNA microarrays. Nature **21** (1999) 10–14
[3] D'Haeseleer, P., Wen, X., Fuhrman, S., Somogyi, R.: Linear modeling of mRNA expression levels during CNS development and injury. In Pacific Symposium on biocomputing, Hawaii. (1999) 41–52
[4] Tavazoie, S., Huges, J.D., Campbell M.J., Cho R.J., Church, G.M.: Systematic determination of genetic network architecture. Nature Genetics **22** (1999) 281–285
[5] DeRisi, J.L., Iyer, V.R., Brown, P.O.: Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale. Science **278** (1997) 680–686
[6] Chu, S. DeRisi, J., Eisen, M. Mulholland, J. Botstein, D. Brown, P.O. Herskowitz, I.: The Transcriptional Program of Sporulation in Budding Yeast. Science **282** (1998) 699–705
[7] 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., Davis, R.W.: A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle. Molecular Cell **2** (July 1998) 65–73
[8] Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. **95** (Nov 1998) 14863-14868
[9] Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive Identification of Cell Cycle-regulated Genes of Yeast *Saccharamyces cerevisiae* by Microarray Hybridization. Molecular Biology of the Cell **9** (1998) 3273–3297
[10] Höppner, F.: Learning Temporal Rules from State Sequences. In IJCAI Workshop on Learning from Temporal and Spatial Data, Seattle, USA (2001) 25-31
[11] Everitt, B.: Cluster Analysis. Heinemann Educational Books, London, England (1974)
[12] Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
[13] Höppner, F., Klawonn, F., Krause, R., Runkler, T.: Fuzzy Cluster Analysis. John Wiley & Sons. Chichester, England (1999)
[14] Wolkenhauer, O.: Data Engineering: Fuzzy Mathematics in System Theory and Data Analysis. John Wiley and Sons, New York (2001)

[15] Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs, NJ (1998)
[16] Zadeh, L.A.: Fuzzy sets. Information and Control **8** (1965) 338–352