International Journal of Neural Systems, Vol. 15, No. 4 (2005) 311–322

World Scientific
www.worldscientific.com

MODELING AND ANALYSIS OF GENE EXPRESSION TIME-SERIES BASED ON CO-EXPRESSION

CARLA S. MÖLLER-LEVET* and HUJUN YIN†
School of Electrical and Electronic Engineering, The University of Manchester
Manchester, M60 IQD, United Kingdom
**c.moller-levet@postgrad.manchester.ac.uk*
†*h.yin@manchester.ac.uk*

In this paper a novel approach is introduced for modeling and clustering gene expression time-series. The radial basis function neural networks have been used to produce a generalized and smooth characterization of the expression time-series. A co-expression coefficient is defined to evaluate the similarities of the models based on their temporal shapes and the distribution of the time points. The profiles are grouped using a fuzzy clustering algorithm incorporated with the proposed co-expression coefficient metric. The results on artificial and real data are presented to illustrate the advantages of the metric and method in grouping temporal profiles. The proposed metric has also been compared with the commonly used correlation coefficient under the same procedures and the results show that the proposed method produces better biologicaly relevant clusters.

Keywords: Similarity metric; gene expressions; co-expressed; time-series; fuzzy clustering; neural networks.

1. Introduction

Microarray experiments measure simultaneously the activity levels of thousands of genes.[1] An appropriate clustering of microarray expression data can lead to classification of diseases, identification of co-expressed and possibly functionally related genes, logical descriptions of gene regulation, etc. A wide variety of statistical and clustering methods have been applied to microarray data.[2–5]

Microarray gene expression time-series can be noisy and are usually short and unevenly sampled. To overcome these undesirable characteristics the time-series can be generalized and smoothed. In this paper we propose to model gene expression profiles using the radial basis function (RBF) neural networks.[6] Most existing methods directly operate on the scattered time points, while modeling the profiles can lead to more generalized, smooth characterization of gene expressions. Standard time-series approaches are limited by the underlying assumptions of the models, such as stationarity or length of the time-series. In contrast, the use of neural networks for modeling time-series is not restricted by model assumptions and the limitations on linearity, noise distribution, irregular sampling and shortness of the time-series. Other modeling techniques proposed for temporal gene expressions are based on splines.[a] In Refs. 7 and 8, gene expression time-series are modeled using mixed-effect models within a mixture model based clustering in which cubic splines are used to model both mean and random effects. The continuous function describing gene expression level is given by the sum of the cluster mean spline function, a spline function for individual gene effects, and Gaussian measurement noise. One of the advantages of the proposed RBF modeling over the mixed-effects modeling is that each gene is modeled independently, which makes the models useful for different types of analysis (e.g., shortest path analysis for transitive functional annotation[9]).

In microarray experiments, the absolute intensities of gene expression are not as revealing as the relative change of intensity, forming the shape of the

[a]Spline functions are piecewise polynomials of degree n that are connected together (at points called knots) so as to have $n-1$ continuous derivations.

expression profile, which is regarded as characteristic and informative. In addition, biological processes are often sampled at short or long intervals of time when intense or moderate biological activity is taking place, leading to unevenly distributed sampling points in the time axis. Direct comparison and operation on these raw time points can easily overlook these characteristics — this may result in incorrect conclusions. An appropriate metric for measuring the similarities of the expression profiles is pivotal to the success of further analysis such as clustering.

The shape of the profiles can be described by the rate of change of expression level in time. To evaluate the similarity of the gene expressions based on their temporal shapes and the distribution of time points we define a co-expression coefficient as the cosine of the first order time-derivative of the modeled profiles. This measure is further incorporated in a cosine based fuzzy clustering algorithm.

The shape issue on unevenly sampled profiles was first addressed by using linear slopes between time points.[10] The time-series are considered as piecewise linear functions and the slopes, defined as $\Delta x/\Delta t$, where x is the gene expression and t is the time, are compared. However, these slopes are proportionally inverse to the length of sampling intervals Δt. Expression levels at long sampling intervals could have too weak impact in the comparison, while those at short sampling intervals could have too strong impact. These intervals vary from some minutes to several hours. By modeling the series it is possible to overcome this drawback. The models will reconstruct the underlying continuous expression profiles, which can be resampled at frequent regular intervals for further analysis.[11] The slope idea of the piecewise linear function between time points has now been extended and generalized to derivatives, quantified by n_s slopes. The value of n_s corresponds to the ratio of the length of sampling interval and the evenly resampling period,

$$n_s = \frac{t_{k+1} - t_k}{\tau}, \tag{1}$$

where t are the sampling time points with $1 \leq k \leq (n_t - 1)$, n_t is the number of time points, and τ is the resampling period. The derivatives of the finer steps are more accurate than the slopes of original time points in describing the shape of the profiles and are also less sensitive to the noise of original measurements.

2. Modeling with Radial Basis Function Neural Networks

The RBF neural networks are versatile and popular model of feedforward neural networks, for which fast, linear learning algorithms exist. The RBF neural networks have a single hidden layer, where the nodes are Gaussian kernels, and a linear output layer, as illustrated in Fig. 1.

The RBF neural network has the form:

$$f(x) = \sum_{i=1}^{n_r} w_i \phi(\|c_i - x\|) + b, \tag{2}$$

where x is the input vector, $\phi(\cdot)$ is a Gaussian kernel, $\|\cdot\|$ denotes the Euclidean norm, w_i are the weights of the output layer, c_i is the center of the ith kernel, and n_r is the total number of kernels.

The Gaussian kernel $\phi(\cdot)$ is defined as:

$$\phi(\|c_i - x\|) = e^{-\frac{(c_i - x)^T (c_i - x)}{2\sigma^2}}, \tag{3}$$

where σ is the width of the kernel. Figure 2 illustrates a simple example of (2).

The problem of RBF modeling is to find appropriate centers and widths of the hidden nodes, and weights of the linear layer. The network is linear in the parameters when all RBF centers and widths of the hidden layer are chosen. Then, the output layer linearly combines the output of the hidden layer and the only adjustable parameters are the weights.

The orthogonal least squares (OLS) learning algorithm has been widely used for training the RBF

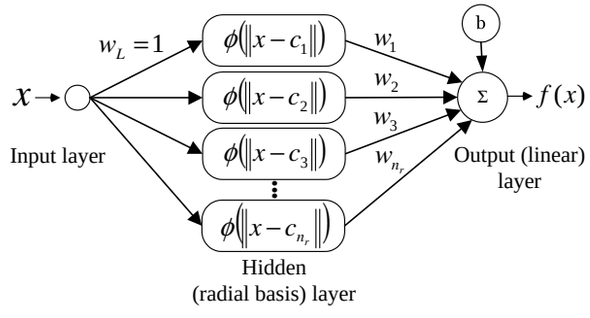

Fig. 1. Radial basis networks have a single hidden layer where the nodes are Gaussian kernels, and an output linear layer. The weights for the first layer are $w_L = 1$ and for the output layer w_i, $1 \leq i \leq n_r$. The number of neurons in the hidden layer, n_r, is less or equal to the number of sample points.

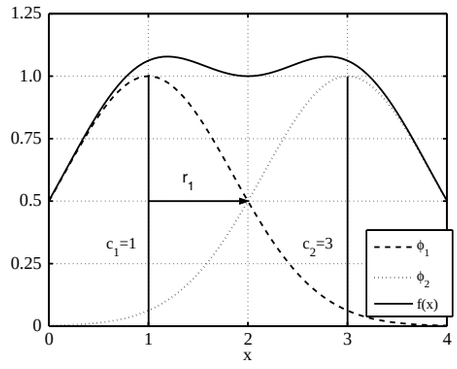

Fig. 2. Two Gaussian kernels, ϕ_1 and ϕ_2, centered at $c_1 = 1$ and $c_2 = 3$ respectively, with radius $r_1 = 1$ (at height 0.5); $f(x)$ is the weighted sum ($w_1 = 1$ and $w_2 = 1$) of ϕ_1 and ϕ_2.

neural networks.[12] The algorithm allows the selection of the centers one by one in a rational procedure. Each selected center maximizes the increment to the explained variance of the desired output and it is not necessary to use all the time points as the centers. However, this method considers all kernels with an equal width, which is inadequate for gene expression modeling when the sampling points are not evenly distributed. In order to improve the approximation, the OLS learning algorithm can be complemented with a heuristic search for the optimal width for each of the candidate centers.[13]

The widths are computed by the nearest neighbor heuristic. The width of the radial basis i centered at c_i is set to the Euclidean distance between c_i and its nearest neighbor center or candidate center c_j multiplied by an overlap constant q, such that:

$$\sigma_i = \frac{q}{\sqrt{-2\log 0.5}} \min(\|c_i - c_j\|). \tag{4}$$

This method takes into account the variations of the distribution of the data. Using this approach, the search for the width is conducted within a fixed interval $[\sigma_{\min}, \sigma_{\max}]$, where

$$\begin{aligned} \sigma_{\min_i} &= \frac{q_{\min}}{\sqrt{-2\log 0.5}} \min(\|c_i - c_j\|), \\ \sigma_{\max_i} &= \frac{q_{\max}}{\sqrt{-2\log 0.5}} \min(\|c_i - c_j\|). \end{aligned} \tag{5}$$

The optimal width minimizes the mean square error of a piecewise linear fit of a segment of the series and the RBF model of the segment. The segment is formed by the h (typically $h = 4$) nearest centers or candidate centers. In general, performing the search using values around $0.5 \leq q_{\min} \leq 2.5$ and $1 \leq q_{\max} \leq 3.5$ leads to satisfactory results. The search for the optimal width is carried out for each of the candidate centers before the selection of a further center. Then, the regression matrix is recalculated with the optimal widths for the candidate centers and a new center is selected. The search procedure is presented in the following pseudocode.

Define:
x_j, time points, $1 \leq j \leq n_t$
$f(x_j)$, expression vector, $1 \leq j \leq n_t$
c_{can_i}, center candidates, $1 \leq i \leq n$
c_{fix_i}, fixed centers.
$q_{\min}$, min overlap constant
$q_{\max}$, max overlap constant
x_{seg}, segment of x
h, defines the length of x_{seg}
For $1 \leq i \leq n$
 Find j such that $x_j = c_{\text{can}_i}$,
 If $h < j \leq (n_t - h)$,
  $x_{\text{seg}} = [x_{j-h}, x_{j-h+1}, \ldots, x_j, \ldots, x_{j+h-1}, x_{j+h}]$
  else if $j \leq h$, $x_{\text{seg}} = [x_1, x_2, \ldots, x_{2h+1}]$
  else if $j > (n_t - h)$, $x_{\text{seg}} = [x_{n_t-(2h-1)}, \ldots, x_{n_t-1}, x_{n_t}]$
 End If
 Calculate $\sigma_{\min_i}$ and $\sigma_{\max_i}$ with Eq. (5).
 For $\sigma_{\min_i} \leq \sigma \leq \sigma_{\max_i}$
  Set $c = [c_{\text{fix}}\ c_{\text{can}_i}]$,
  Calculate:
  $A = e^{\frac{-(|c-x|)^2}{2\sigma^2}}$ and W and B from $f(X) = WA + B$
  $A_{\text{seg}} = e^{\frac{-(|c-x_{\text{seg}}|)^2}{2\sigma^2}}$ and $Y_{\text{seg}} = WA_{\text{seg}} + B$
  $\text{MSE}(\sigma) = \frac{1}{2h+1} \sum (f(x_{\text{seg}}) - Y_{\text{seg}})^2$
 End For
  Assign σ_i such that $\text{MSE}(\sigma)$ is the minimum.
End For

In addition, extra time points should be inserted to the profiles in order to increase the linearity between time points and to aid the approximation of sharp turns. The extra time points correspond to the middle points between original time points, and additional time points in the intervals forming small angles.

3. Co-Expression Coefficient

The Pearson's product-moment correlation coefficient, $\rho \in [-1, 1]$, is a popular similarity measure for comparing time-series. It is a statistical term measuring the linear relationship between two variables.

The correlation coefficient of variables x and y is defined by

$$\rho(x,y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}, \quad (6)$$

where n is number of observations, and $\bar{x}$ and $\bar{y}$ are the mean values of x and y respectively. The correlation coefficient is symmetric and does not change as the result of a linear transformation, $ax + b$, except for a change of sign if a is negative. For linearly independent variables, $\rho(x,y) = 0$.

In Ref. 14, it was found that given its ability to capture the shape while disregarding the magnitude of the series, the correlation coefficient conforms well to the intuitive biological notion of co-expression. Yet, the correlation is not necessarily coherent with the shape and it does not consider uneven sampling intervals. For example, similar to Ref. 15, Fig. 3 shows a set of profiles which present a higher similarity of shape than the set of profiles shown in Fig. 4. However, the correlations indicate the opposite.

The shape of the profiles can be described by the rate of change of expression level in time, $d(x(t))/d(t)$, where $x(t)$ is a continuous function of time t, or Δx, when x is a discrete gene expression by evenly resampling the modeled profile. In order to effectively compare shapes we define a co-expression coefficient, ce, as

$$ce(x,y) = \frac{\sum_{i=1}^{n}\Delta x_i \Delta y_i}{\sqrt{\sum_{i=1}^{n}\Delta x_i^2 \sum_{i=1}^{n}\Delta y_i^2}}, \quad (7)$$

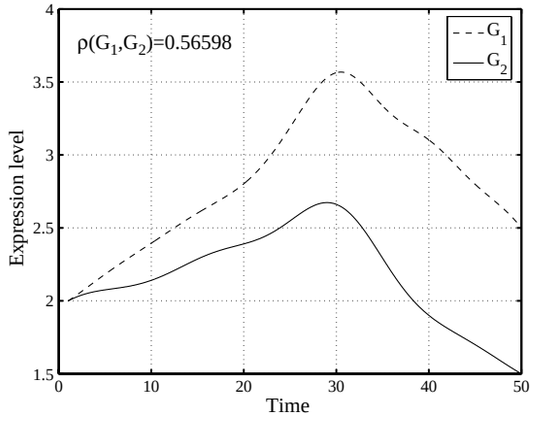

Fig. 3. Profiles with higher similarity of shape than profiles in Fig. 4, however their correlation coefficient is lower.

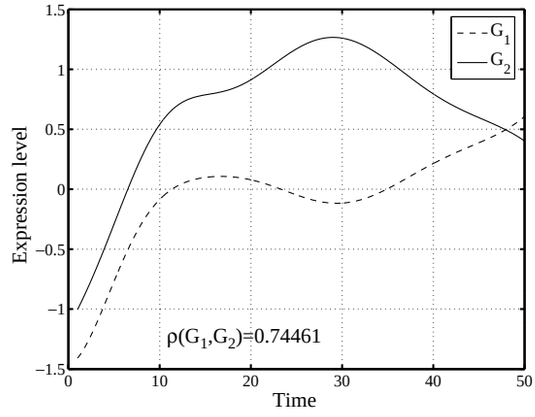

Fig. 4. Profiles with lower similarity of shape than profiles in Fig. 3, however their correlation coefficient is higher.

which corresponds to the cosine or (uncentered) correlation coefficient of the first order time-derivative of the modeled profiles.[b] $-1 \leq ce \leq 1$, $ce = 1$ when two profiles are identical and $c = -1$ when they have opposite shapes.

A similar measure has been used in the field of astronomy, where the index is known as the comovement coefficient,[16] defined as the correlation coefficient of differentiated time-series. However, the correlation coefficient does not consider vertical shifts which, in the case of positive and negative derivatives, is important to identify. Therefore, the uncentered correlation coefficient is used here, the mean is not subtracted and the sign of the derivatives is preserved.

Figures 5 and 6 show the derivatives of the profiles presented in Figs. 3 and 4 respectively, and the co-expression coefficients, which successfully incorporate the shape information in the comparison of the temporal profiles.

To illustrate the advantages of the co-expression coefficient in handling unevenly sampled series, consider three profiles, a, b and c, as shown in Fig. 7. Each has five time points, marked with $\circ$, Δ and $*$ respectively, where the sampling time points correspond to $t = 1, 2, 3, 4, i$, and i varies as follows: $i = 4.5, 5, 6, 7, 8$. Profile b is identical to a in all time points except for the first one, while c is identical to a in all time points except for the last one. When the sampling time points are evenly distributed,

[b]The RBF modeled profiles are further smoothed to reduce noise impact of the derivatives.

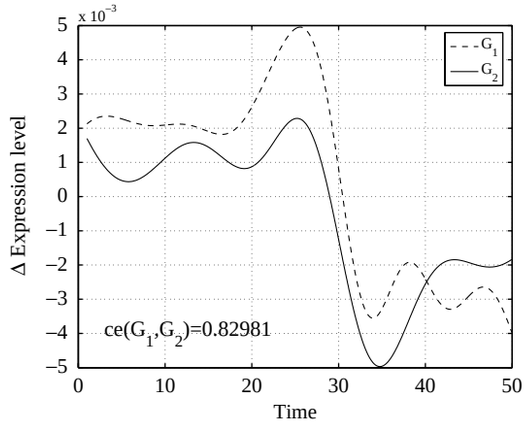

Fig. 5. Differentiation of the profiles presented in Fig. 3.

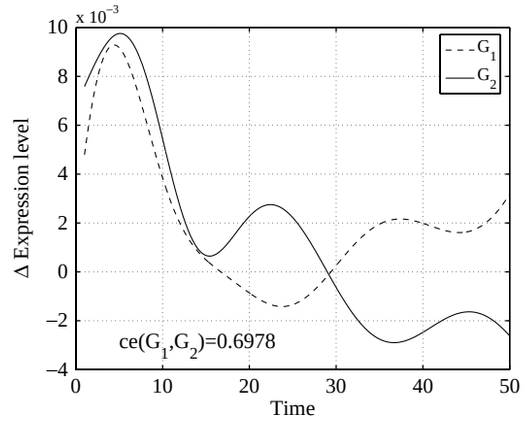

Fig. 6. Differentiation of the profiles presented in Fig. 4.

b and c are equally similar to a. However, if the sampling interval varies, the rate of change varies accordingly. Figure 7 (top) presents the RBF modeling of the three profiles with evenly distributed time points. The correlation coefficients of a and b, and a and c, revel that b and c are equally similar to a, ($\rho(a,b) = \rho(a,c) = 0.945$). However, as i increases (Fig. 7 (bottom)), the similarity of a and c is expected to grow, while the similarity of a and b is expected to maintain constant. This behavior cannot be correctly reflected in the direct correlation coefficient of the profiles but can be observed when calculating the proposed co-expression coefficient. Table 1 shows the co-expression coefficient and correlation coefficient of the models for different values of i. The correlation coefficient on the original time points is invariant to the change of sampling interval, $\rho_o(a,b) = \rho_o(a,c) = 0.953$, $\forall i$.

Table 1. Co-expression coefficient (ce), and correlation coefficient (ρ) of modeled profiles a, b and c in Fig. 7 for various values of i. As i increases, $ce(a,c)$ increases, while $\rho(a,c)$ decreases.

i	$ce(a,b)$	$ce(a,c)$	$\rho(a,b)$	$\rho(a,c)$
4.5	0.984	0.980	0.946	0.960
5	0.981	0.981	0.945	0.945
5.5	0.981	0.984	0.946	0.938
6	0.981	0.985	0.949	0.936
7	0.981	0.991	0.950	0.936
8	0.981	0.993	0.949	0.920

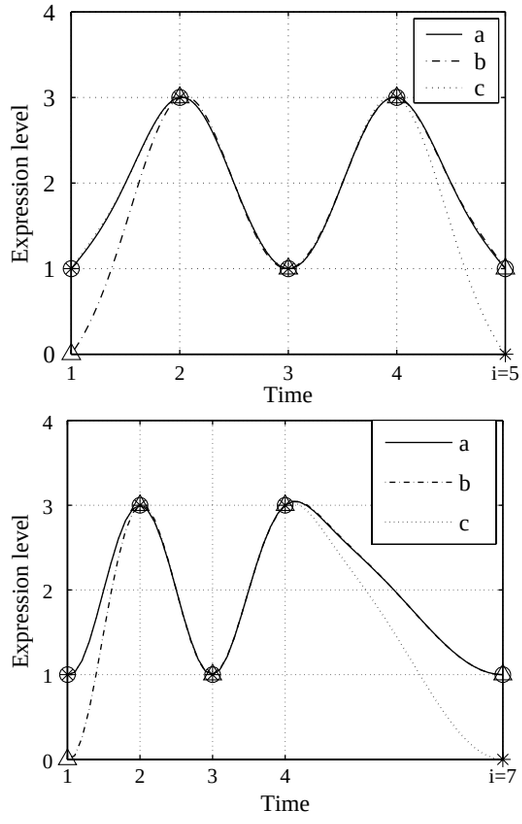

Fig. 7. The figures show the influence of the sampling interval on the shape of the profiles. Top figure shows evenly sampled profiles ($i = 5$) and bottom figure shows unevenly sampled profiles ($i = 7$).

4. Fuzzy Clustering Based on Co-Expression

The objective function, which measures the desirability of partitions in fuzzy c-means clustering (FCM)[17]

is described by,

$$J(x, v, u) = \sum_{i=1}^{n_c} \sum_{j=1}^{n_g} u_{ij}^m d^2(x_j, v_i), \tag{8}$$

where n_c is the number of clusters, n_g is the number of vectors to cluster, u_{ij} is the membership degree of the vector x_j to the cluster i, $d^2(x_j, v_i)$ is the squared distance between vector x_j and prototype v_i, and m is the parameter that determines the degree of overlap of fuzzy clusters. The optimization of the FCM process is operated on the inner product induced norms, the use of a different metric requires the recalculation of v and u to minimize the objective function.

In order to derive a fuzzy clustering algorithm based on the co-expression coefficient, the dot-product fuzzy clustering algorithm is implemented and the input data is normalized to unit length. Considering that the higher the *ce* value, the higher the similarity between the profiles and $-1 \leq ce \leq 1$, the dissimilarity function can be defined as:

$$D(x_j, v_i) = 1 - ce(x_j, v_i). \tag{9}$$

If vectors are normalized to unit length, $D(x_j, v_i)$ can be simplified to

$$D(x_j, v_i) = 1 - \Delta x_j^{\mathrm{T}} \Delta v_i \tag{10}$$

In Ref. 18, the dot-product fuzzy clustering was developed by calculating the updating expression for v that minimizes (8), when the distance is defined as $d(x_j, v_i) = 1 - x_j^{\mathrm{T}} v_i$, and the vectors are normalized to unit length. The updating rule for the prototype v of cluster i at time point t is,

$$v_{it} = \frac{\sum_{j=1}^{n_g} u_{ij}^m x_{jt}}{\sqrt{\sum_{k=1}^{n_t} (\sum_{j=1}^{n_g} u_{ij}^m x_{jk})^2}} \tag{11}$$

where n_t is the number of time points.

The number of clusters n_c and the fuzziness parameter m are user-defined parameters in the FCM algorithm. In this paper, the PBM-index (described in Sec. 5.2) is used to validate the number of clusters.[19] In Ref. 3, the selection of m for an optimal performance of fuzzy clustering for microarray data is addressed. The selection can also be aided by validity measures and membership plots.[c]

5. Yeast Cell Cycle Dataset

In Spellman *et al.*,[2] cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* were identified by microarray hybridization. Three independent methods: α factor arrest, elutriation and arrest of a cdc15 temperature sensitive mutant were used to synchronized the yeast cultures. The yeast cells were sampled every 7 minutes for 119 minutes. We utilized the temporal expression of the yeast culture synchronized by α factor arrest to illustrate the proposed method.

Eight hundred cell cycle-regulated genes were identified in the analysis of Spellman *et al.*[2] using Fourier analysis of combined data of all three experiments. Among the 800 genes, 511 with no missing values for the α experiment are available from their web site.[d] In Ref. 20, the data is re-analyzed using a shape-invariant model together with a false discovery rate procedure to identify periodically expressed genes. The authors identified 297 cell cycle-regulated genes in the α experiment. Out of these 297 genes, 208 have no missing values. In addition, there are 104 genes determined to be cell cycle-regulated by traditional methods. Out of these 104 genes, 71 with no missing values are available from the Spellman *et al.*[2] dataset. In Ref. 2, 95 of the 104 genes were identified as cell cycle-regulated, while in Ref. 20, 47 were identified. The 511, 208 and 71 genes, corresponding to the cell cycle-regulated genes identified by Refs. 2 and 20 and traditional methods, respectively, are used here to form three test datasets.

5.1. *Modeling*

The expression series are modeled using the RBF neural networks. The moving average smoothing technique is used to further smooth the modeled profiles. Figure 8 presents three examples of modeled profiles from the Spellman *et al.* dataset and their corresponding smoothed expressions.

The cell cycle period for the α dataset has been identified with different methods, producing different results. A time-frequency analysis using wavelet transforms is used to identify the period in Ref. 21. The authors conclude that the dominant period is not that of the cell cycle but, the higher frequency 30–40 min submultiples of the cycle period. In

[c]These are plots of membership degrees into grayscales in a hierarchical fashion.[10]
[d]Dataset available from http://cellcycle-www.stanford.edu.

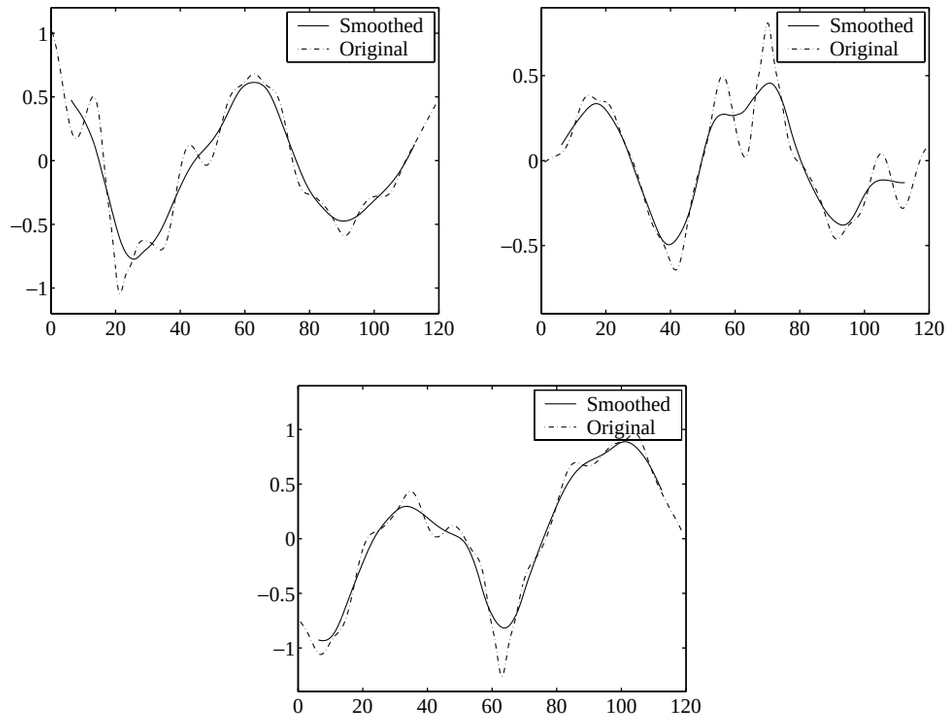

Fig. 8. Example genes (YAL040C, YCR077W and YDL055C) and their corresponding smoothed expression. The horizontal axis denotes time [mins] and the vertical axis denotes the expression level ($\log_2$(ratio)).

Ref. 4, the authors analyze the similarity of a time-series curve with itself and deduce that the period is 70 minutes. Similar to Ref. 21, the authors in Ref. 4 observe that there are very strong indications of underlying oscillatory phenomena with periods smaller than the observed cell cycles, around 20 and 40 minutes. In Ref. 20, five clusters are identified and the times between peaks for four clusters are estimated to be 67, 63.4, 54.2, and 61.4 minutes. The modeling and smoothing of the profiles proposed in this paper allow the estimation of times between peaks. With higher occurrences at periods between 55 and 65 minutes, the estimated times are comparable to the previously identified results. The summery of the estimated times between two peaks for the three datasets is shown in Table 2, and the corresponding histograms are presented in Fig. 9.

5.2. *Clustering*

To facilitate the validation of the results, the clustering is not initialized randomly, instead, the Mountain method[22] is used to initialize the algorithm. The Mountain method is a simple and effective approach for obtaining the initial values of the clusters that are required by more complex cluster algorithms. As a first step a discrete grid of points, which will be the candidate cluster centers, are assign over the hyper-rectangle containing the dataset. At each vertex in the grid, the mountain function, which measures local density of data points in its proximity, is calculated. The mountain candidate with the highest mountain function (peak) is identified and set as a cluster center. The mountain function is then flatten at each candidate by subtracting an inversely proportional amount to its distance from the peak candidate. The process is iterated until the mountain function is less than a threshold for every candidate, or until a desired number of clusters centers have being generated.

Table 2. Estimated times (in minutes) between two peaks, for the 71, 208 and 511 genes datasets using the proposed RBF modeling.

Dataset	Mean	Median	Std. Dev.
71 genes	60.2289	59.2500	10.1421
208 genes	59.1010	58.7500	5.7688
511 genes	57.0186	57.7500	13.6742

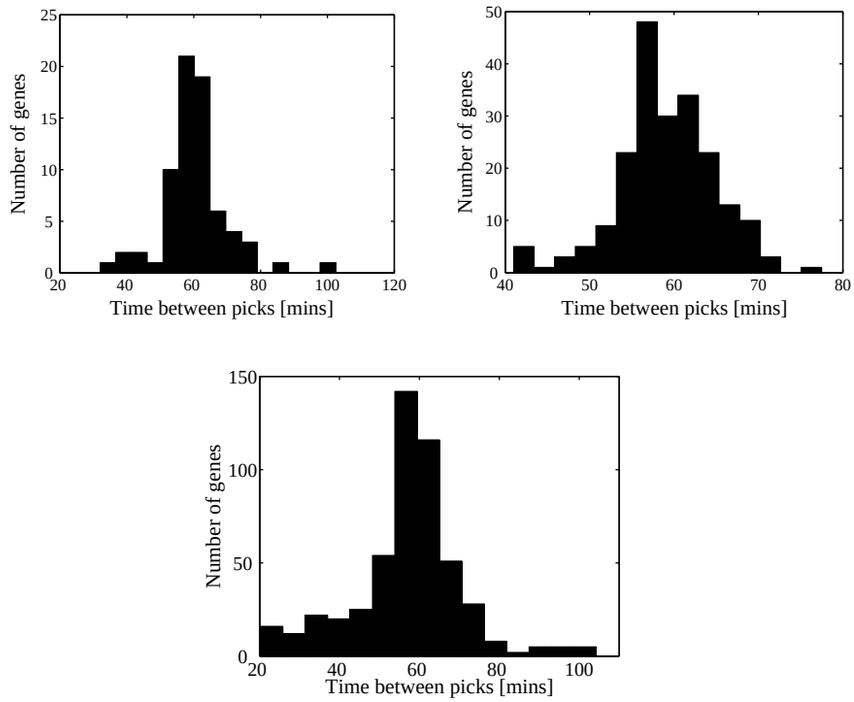

Fig. 9. Histograms of the estimated periods between peaks of genes in the three datasets used. Top left 71 genes, top right 208 genes and center 511 genes.

The number of clusters was validated using the PBM index.[19] The index is a product of three factors, and its maximization ensures the formation of a small number of compact clusters with large separation between at least two clusters. The PBM-index is defined as follows:

$$PBM(n_c) = \left[\left(\frac{1}{n_c} \right) \left(\frac{E_1}{E_{n_c}} \right) D_{n_c} \right]^2, \qquad (12)$$

where n_c is the number of clusters, $E_{n_c} = \sum_{c=1}^{n_c} E_c$, $E_c = \sum_{g=1}^{n_g} u_{cg} d(x_g, z_c)$, z_c is the center of the cth cluster and $D_{n_c} = \max_{i,j=1}^{n_c} d(z_i, z_j)$, where d is the distance function, the total number of genes is n_g, $U(X) = [u_{cg}]$, $1 \leq c \leq n_c$ and $1 \leq g \leq n_g$ is the partition matrix for the data.

Figure 10 plots the PBM-index as a function of the number of clusters for the three datasets. The genes are assigned to the clusters according to their highest membership degree. A hierarchical clustering of the partition matrix is performed to order the genes and obtain a complete visualization of the results. The results are shown in Fig. 11. The membership plot can be utilized to identify genes with similar distribution of membership degree across all the clusters.

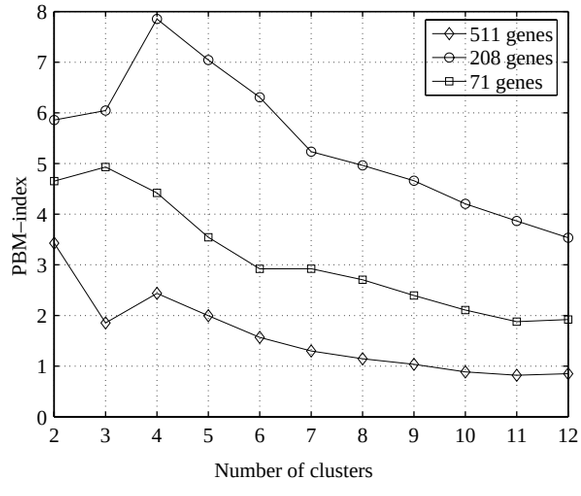

Fig. 10. High values of the PBM-index indicate the formation of small number of compact clusters with large separation.

The genes are classified according to the different cell cycle phases used in Ref. 2. The classification of the genes is based on the minimum error, comparing their peak times to the peak times described in Ref. 2 for each phase. Table 3 presents the distributions of the genes in each cluster among the cell cycle

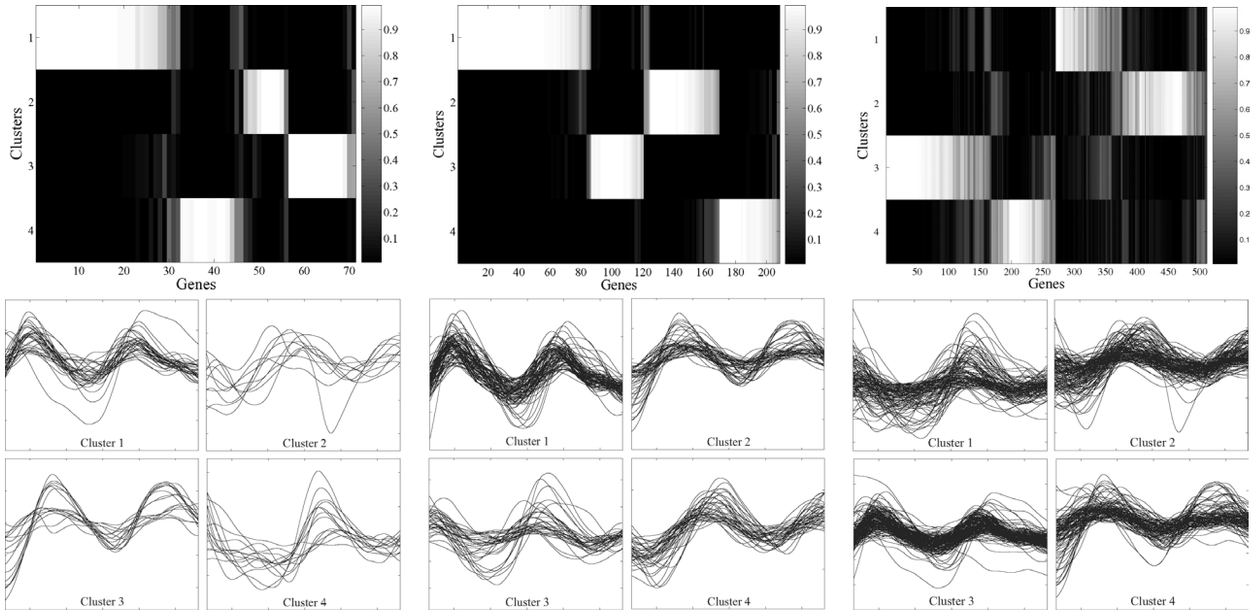

Fig. 11. Clustering results (membership plot and time-series plots) for the 71 genes (left), 208 (center) and 511 (right) datasets. In time-series plots the horizontal axis denotes time [mins] and the vertical axis denotes the expression level ($\log_2$(ratio)).

phases. In the 71 and 208 genes datasets, the genes in each cluster belong mainly to one phase or the two neighboring phases. In the 511 genes dataset, the genes are more spread mainly because of the inclusion of genes which are not cell cycle-regulated and have a low membership degree to all the clusters, for example genes in Fig. 11 (right-bottom), cluster 3.

The dependency between the obtained clusterings and the known cell cycle phases can be quantified using a Chi-squared (χ^2) based measure,

$$\chi^2 = \sum_{i=1}^{p} (O_i - E_i)^2 / E_i, \tag{13}$$

where E and O are the expected and observed frequencies of genes respectively and p is the number of cells (clusters $\times$ phases). The measure is used to compare the performance of RBF modeling and co-expression coefficient against other approaches such as RBF modeling and correlation coefficient, original time points and correlation coefficient, and random clustering. The random clustering is randomly grouping of the data into a predefined number of clusters.[23] Fuzzy clustering based on correlation coefficient can be obtained by using the standard FCM and sphere standardized (zero mean and unit length) profiles.[24] Table 4 presents the χ^2 values for all three datasets for the different clusterings.[e] The

Table 3. Distribution of the 71 genes (top), 208 genes (middle), and 511 genes (bottom) datasets, among the five different cell cycle phases over the four clusters obtained using the proposed method.

Cluster	M/G1	G1	S	G2	M
1(32)	3	29	0	0	0
2(10)	1	0	0	3	6
3(15)	0	1	13	1	0
4(14)	13	1	0	0	0
1(82)	2	80	0	0	0
2(39)	0	7	27	5	0
3(34)	30	2	0	0	2
4(53)	0	0	2	36	15
1(103)	74	13	3	3	10
2(133)	10	5	4	77	37
3(169)	13	152	4	0	0
4(106)	1	28	48	29	0

[e]To obtain the expected value, E, phases G2 and M are merged, and the total number of genes in each phase (M/G1, G1, S G2/M) is calculated. The expected value for cells out of the diagonal is zero, therefore, a value of one was added to all the cells to void division by zero.

Table 4. χ^2 values for the three datasets for RBF modeling and co-expression coefficient (RBF-ce), RBF modeling and correlation coefficient (RBF-ρ), original time points and correlation coefficient (O-ρ), and random clustering. For random clustering the mean of 50 runs is reported. RBF-ce consistently outperforms the other approaches.

Dataset	RBF-ce	RBF-ρ	O-ρ	Random
71 genes	14.1	236.3	229.4	372.7
208 genes	91.9	109.5	143.8	2657.3
511 genes	2328.7	2777.9	3204.7	14995.0

Table 5. Clustering results of the 71 genes dataset. The column "Gene" corresponds to the gene with the highest membership to the cluster, and "Phase", corresponds to the phase of the gene identified by traditional methods.

Cluster	Gene	Phase
1	YKL113C	G1
2	YMR001C	G2/M
3	YNL030W	S
4	YNL192W	M/G1

Table 6. Genes with high similarity to YDR150W, in the 208 genes dataset, according to their membership distribution. "Phase" corresponds to the phase identified by traditional methods, "U" stands for uncharacterized gene.

Membership	Gene	Phase
0.9957	YAR007C	G1
0.9951	YDL156W	U
0.9970	YLR103C	G1
0.9985	YDL163W	U
0.9988	**YDR150W**	**G1**
0.9966	YDL164C	G1
0.9975	YBR088C	G1
0.9939	YPL153C	G1
0.9946	YPL128C	U

RBF modeling with co-expression coefficient has the lowest χ^2 value for the three datasets, showing a higher relevancy between the obtained clusters and the true cell cycle phases.

For the 71 genes identified by traditional methods, Table 5 shows that the phase of the genes with the highest membership degree to a cluster coincides with the phase represented by the cluster (shown in Table 3). The clustering results of these previously known genes, indeed show biological relevance, indicating that the proposed method is a useful technique for gene expression time-series analysis.

In the case of the dataset formed by the 208 genes identified as cell cycle-regulated in Ref. 20, the gene with the highest membership degree to the first cluster, YDR150W, has been identified as G1 phase gene by traditional biological methods. Table 6, presents the genes with high similarity to YDR150W according to their membership distribution. YPR019W, essential for initiation of DNA replication, has the highest membership degree to the third cluster. YNL030W, a core histone required for chromatin assembly and chromosome function, holds the highest membership degree to the second cluster, which mainly represents genes of the S phase. The membership plot in Fig. 11 (middle-top) shows a sharp boundary between clusters 2 and 3. The corresponding time-series plots in Fig. 11 (middle-bottom) show that these clusters are shifted by almost a half cell cycle, thus, possessing opposite shapes. Finally, YGL021W, has the highest membership degree to the fourth cluster.

In the dataset formed by the 511 genes identified as cell cycle-regulated in Ref. 2, YHR005C, possesses the highest membership degree to cluster 1. Cluster 2 is mainly formed by genes belonging to phases G2 and M. YGL021W, has the highest membership degree to this cluster. This gene, is also the highest member of cluster 4 (formed by G2 and M phase genes) in the 208 genes dataset. Cluster 3 mainly contains genes belonging to phase G1, and the gene with the highest membership degree to cluster 3, YLR103C, was identified to peak in the G1 phase by traditional methods. YOR248W, uncharacterized, has the highest membership degree to cluster 4, which has genes associated to the G1, S and G2 phases.

The membership and time-series plots indicate that the co-expression based fuzzy clustering of the modeled profiles allows the grouping of expressions based on their temporal shapes. The resulting distribution of the genes from the three datasets among the cell cycle phases indicates that the proposed method is able to extract the meaningful groups similar to biological definitions.

6. Conclusions

The modeling of gene expression profiles using RBF neural networks can lead to a more generalized, smooth characterization of gene expressions. An extended OLS method is proposed to optimize the network parameters, both centers and individual widths. The obtained models are smoothed to reduce noise and differentiated to characterize the shapes of the expression profiles. Then, the use of the uncentered correlation coefficient on the derivatives of the modeled profiles as a shape similarity metric, termed the co-expression coefficient, allows the comparison of profiles based on their temporal shapes. Considering the advantages of fuzzy sets, a fuzzy clustering algorithm based on the proposed co-expression coefficient has been derived for clustering temporal profiles.

The proposed RBF modeling with the co-expression coefficient approach is compared to RBF modeling with the correlation coefficient, original time points with the correlation coefficient and random clustering. The results show that RBF modeling with the co-expression coefficient produces clusters with the highest correspondence with the known cell cycle phases.

Acknowledgments

The authors would like to thank the reviewers for their helpful comments. This research was supported by an Overseas Research Studentship (ORS) award from Universities U.K. and Consejo Nacional de Ciencia y Tecnologia (CONACYT).

References

1. P. Brown and D. Botstein, Exploring the new world of the genome with DNA microarrays, *Nature Genetics Supplement* **21** (1999) 33–37.
2. P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein and B. Futcher, Comprehensive identification of cell cycle-regulated genes of yeast *Saccharamyces cerevisiae* by microarray hybridization, *Mol. Biol. Cell* **9** (1998) 3273–3297.
3. D. Dembélé and P. Kastner, Fuzzy c-means method for clustering micoarray data, *Bioinformatics* **19** (2003) 973–980.
4. V. Filkov, S. Skiena and J. Zhi, Analysis techniques for microarray time-series data, *J. Computational Biology* **9** (2002) 317–330.
5. C. S. Möller-Levet, K.-H. Cho and O. Wolkenhauer, Microarray data clustering based on temporal variation: FCV with TSD preclustering, *Applied Bioinformatics* **2** (2003) 35–45.
6. J. Park and I. Sandberg, Approximation and radial basis function networks, *Neural Computing* **5** (1993) 305–316.
7. Z. Bar-Joseph, G. Gerber, D. K. Gifford, T. S. Jaakkola and I. Simon, A new approach to analyzing gene expression time series data, in *Proc. of RECOMB* (Washington DC, USA, 2002), pp. 39–48.
8. Y. Luan and H. Li, Clustering of time-course gene expression data using a mixed-effects model with B-splines, *Bioinformatics* **19** (2003) 474–482.
9. X. Zhou, M.-C. J. Kao and W. H. Wong, Transitive functional annotation by shortest-path analysis of gene expression data, *PNAS* **99** (2002) 12783–12788.
10. C. S. Möller-Levet, F. Klawonn, K.-H. Cho, H. Yin and O. Wolkenhauer, Clustering of unevenly sampled gene expression time-series data, *Fuzzy Sets and Systems* **152** (2005) 49–66.
11. C. S. Möller-Levet and H. Yin, Modelling and clustering of gene expressions using RBFs and a shape similarity metric, in *Proc. of the Fifth Int. Conf. Intelligent Data Engineering and Automated Learning* (Exeter, UK, 2004), pp. 1–10.
12. S. Chen, C. F. N. Cowan and P. M. Grant, Orthogonal least squares learning algorithm for radias basis function networks, *IEEE Transactions on Neural Networks* **2** (1991) 302–309.
13. C. S. Möller-Levet, H. Yin, K.-H. Cho and O. Wolkenhauer, Modelling gene expression time-series with radial basis function neural networks, in *Proc. of the Int. Joint Conf. Neural Networks*, **II** (Budapest, Hungary, 2004), pp. 1191–1196.
14. M. B. Eisen, P. T. Spellman, P. O. Brown and D. Botstein, Cluster analysis and display of genome-wide expression patterns, in *Proc. Natl. Acad. Sci. USA* **95** (1998), pp. 14863–14868.
15. S. D. Peddada, E. K. Lobenhofer, L. Li, C. A. Afshari, C. R. Weinberg and D. M. Umbach, Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference, *Bioinformatics* **19** (2003) 834–841.
16. E. D. Feigelson and G. J. Babu, *Statistical Challenges in Modern Astronomy* (New York: Springer-Verlag, 1992).
17. J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms* (Plenum Press: New York, 1981).
18. F. Klawonn and A. Keller, Fuzzy clustering based on modified distance measures, in *Proc. of the Third Symp. Intelligent Data Analysis* (Amsterdam, The Netherlands, 1999), pp. 291–301.

19. M. K. Pakhira, S. Bandyopedhyay and U. Maulik, Validity index for crisp and fuzzy clusters, *Pattern Recognition* **37** (2003) 487–501.
20. H. Luan and H. Li, Model-based methods for identifying periodically expressed genes based on time course microrray gene expression data, *Bioinformatics* **20** (2004) 332–339.
21. R. R. Klevecz and H. B. Dowse, Tuning in the transcriptome: Basins of attraction in the yeast cell cycle, *Cell Prolif* **33** (2000) 209–218.
22. R. R. Yager and D. P. Filev, Approximate clustering via the mouintain method, *IEE Transactions on Systems, Man, and Cybernetics* **24** (1994) 1279–1284.
23. K. Y. Yeung, D. R. Haynor and W. L. Ruzzo, Validating clustering for gene expression data, *Bioinformatics* **17** (2001) 309–318.
24. X. Golay, S. Kollias, G. Stoll, D. Meier, A. Valavanis and P. Boesiger, A new correlation-based fuzzy logic clustering algorithm for fMRI, *Magnetic Resounance Medicine* **40** (1998) 249–260.