BIOINFORMATICS ORIGINAL PAPER Vol. 26 no. 18 2010, pages 2281–2288
doi:10.1093/bioinformatics/btq422

Gene expression

Advance Access publication July 16, 2010

Multiple gene expression profile alignment for microarray time-series data clustering

Numanul Subhani[1], Luis Rueda[1,*], Alioune Ngom[1] and Conrad J. Burden[2]

[1]School of Computer Science, 5115 Lambton Tower, University of Windsor, 401 Sunset Avenue, Windsor, Ontario N9B 3P4, Canada and [2]Mathematical Sciences Institute, The Australian National University, Canberra ACT 0200, Australia

Associate Editor: Martin Bishop

ABSTRACT

Motivation: Clustering gene expression data given in terms of time-series is a challenging problem that imposes its own particular constraints. Traditional clustering methods based on conventional similarity measures are not always suitable for clustering time-series data. A few methods have been proposed recently for clustering microarray time-series, which take the temporal dimension of the data into account. The inherent principle behind these methods is to either define a similarity measure appropriate for temporal expression data, or pre-process the data in such a way that the temporal relationships between and within the time-series are considered during the subsequent clustering phase.

Results: We introduce *pairwise gene expression profile alignment*, which vertically shifts two profiles in such a way that the area between their corresponding curves is minimal. Based on the pairwise alignment operation, we define a new distance function that is appropriate for time-series profiles. We also introduce a new clustering method that involves *multiple expression profile alignment*, which generalizes pairwise alignment to a set of profiles. Extensive experiments on well-known datasets yield encouraging results of at least 80% classification accuracy.

Contact: lrueda@uwindsor.ca

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on May 7, 2010; revised on July 13, 2010; accepted on July 15, 2010

1 INTRODUCTION

An important process in functional genomic studies is clustering microarray time-series data, where genes with similar expression profiles are expected to be functionally related (Cho *et al.*, 1998). A Bayesian approach (Ramoni *et al.*, 2002), a partitional clustering based on *k*-means (Tavazoie *et al.*, 1999) and a Euclidean distance approach (Tamayo *et al.*, 1999) have been proposed for clustering time-series gene expression profiles. They have applied self-organizing maps (SOMs) to visualize and to interpret the gene temporal expression profile patterns. A hidden phase model was used for clustering time-series data to define the parameters of a mixture of normal distributions in a Bayesian-like manner that are estimated by using expectation maximization (EM; Bréhélin, 2005). Also, the methods proposed in Chu *et al.* (1998) and Heyer *et al.* (1999) are based on correlation measures. A method that uses jack–knife correlation with or without using seeded candidate profiles was proposed for clustering time-series microarray data as well (Heyer *et al.*, 1999), where the resulting clusters depend upon the initially chosen template genes, because there is a possibility of missing important genes. A regression-based method was proposed in Ernst *et al.* (2005) to address the challenges in clustering short time-series expression data. Analyzing gene temporal expression profile data that are non-uniformly sampled and can contain missing values has been studied in Bar-Joseph *et al.* (2003). Clustering temporal gene expression profiles was studied by identifying homogeneous clusters of genes in Déjean *et al.* (2007). The *shapes of the curves* were considered instead of the *absolute expression ratios*. Fuzzy clustering of gene temporal profiles, where the similarities between co-expressed genes are computed based on the rate of change of the expression ratios across time, has been studied in Moller-Levet *et al.* (2005). In Peddada *et al.* (2005), the idea of order-restricted inference levels across time has been applied to select and cluster genes, where the estimation makes use of known inequalities among parameters. In Rueda *et al.* (2008), pairs of profiles represented by piece-wise linear functions are aligned in such a way to minimize the integrated squared area between the profiles. An agglomerative clustering method, combined with an area-based distance measure between two aligned profiles, was used to cluster microarray time-series data. Using natural cubic spline interpolations, we re-formulated the pairwise gene expression profile alignment problem of Rueda *et al.* (2008) in terms of arbitrary functions that are continuously integrable on a finite interval, and extended the concept of pairwise alignment to multiple expression profile alignment. Finally, we combined *k*-means and EM clustering with multiple alignment to cluster microarray time-series data, yielding at least 80% classification accuracy on well-known data.

2 SYSTEM AND METHODS

2.1 Pairwise expression profile alignment

Given two profiles, $x(t)$ and $y(t)$ (either piece-wise linear or continuously integrable functions), where $y(t)$ is to be aligned to $x(t)$, the basic idea of alignment is to *vertically shift* $y(t)$ towards $x(t)$ in such a way that the *integrated squared errors* between the two profiles is minimal. Let $\hat{y}(t)$ be the result of shifting $y(t)$. Here, the *error* is defined in terms of the areas between $x(t)$ and $\hat{y}(t)$ in interval $[0,T]$. While $x(t)$ and $\hat{y}(t)$ may cross each other many times, we want that the sum of all the areas where $x(t)$ is above $\hat{y}(t)$ minus the sum of those areas where $\hat{y}(t)$ is above $x(t)$ is minimal (Fig. 1). Let a denote the amount of vertical shifting of $y(t)$. Then, we want to find

*To whom correspondence should be addressed.

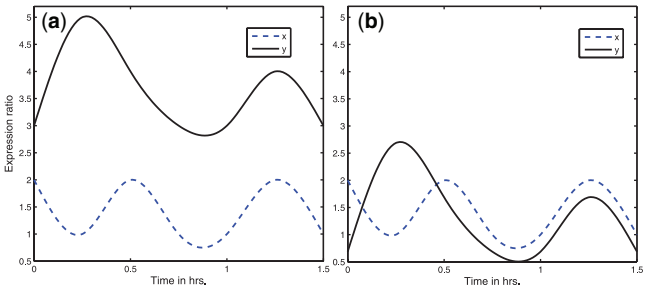

Fig. 1. (**a**) Unaligned profiles $x(t)$ and $y(t)$. (**b**) Aligned profiles $x(t)$ and $y(t)$, after applying $y(t) \leftarrow y(t) - a_{\min}$.

the value $a_{\min}$ of a that minimizes the integrated squared error between $x(t)$ and $\hat{y}(t)$. Once we obtain $a_{\min}$, the alignment process consists of performing the shift on $y(t)$ as $\hat{y}(t) = y(t) - a_{\min}$.

The pairwise alignment that we propose here applies to profiles, that are *any* integrable functions on a finite interval. Suppose that we have two profiles, $x(t)$ and $y(t)$, defined on the time-interval $[0, T]$. The alignment of $x(t)$ and $y(t)$ consists of finding the value a that minimizes

$$f_a(x,y) = \int_0^T [x(t) - [y(t) - a]]^2 \mathrm{d}t, \tag{1}$$

which is a quadratic function involving a vertical shift factor, a. Differentiating yields

$$\frac{\mathrm{d}}{\mathrm{d}a} f_a(x,y) = 2\int_0^T [x(t) - y(t)]\mathrm{d}t + 2aT. \tag{2}$$

Setting $\frac{\mathrm{d}}{\mathrm{d}a} f_a(x,y) = 0$ and solving for a gives

$$a_{\min} = -\frac{1}{T}\int_0^T [x(t) - y(t)]\mathrm{d}t. \tag{3}$$

Since $\frac{\mathrm{d}^2}{\mathrm{d}a^2} f_a(x,y) = 2T > 0$ then $a_{\min}$ is a minimum. Thus, there is only a single vertical shift factor that minimizes the integrated squared error. The integrated error between $x(t)$ and the shifted $\hat{y}(t) = y(t) - a_{\min}$ is then

$$\int_0^T [x(t) - \hat{y}(t)]\mathrm{d}t = \int_0^T [x(t) - y(t)]\mathrm{d}t + a_{\min} T = 0. \tag{4}$$

Given an original profile $x(t) = [e_1, e_2, \ldots, e_n]$ (with n expression values taken at n time-points $t_1, t_2, \ldots, t_n$), in our approach, we use *natural cubic spline* interpolation, with n knots, $(t_1, e_1), \ldots, (t_n, e_n)$, to represent $x(t)$ as a continuously integrable function

$$x(t) = \begin{cases} x_1(t) & \text{if} \quad t_1 \le t \le t_2 \\ \cdots & \\ x_j(t) & \text{if} \quad t_j \le t \le t_{j+1} \\ \cdots & \\ x_{n-1}(t) & \text{if} \quad t_{n-1} \le t \le t_n \end{cases} \tag{5}$$

where $x_j(t) = x_{j3}(t - t_j)^3 + x_{j2}(t - t_j)^2 + x_{j1}(t - t_j)^1 + x_{j0}(t - t_j)^0$ interpolates $x(t)$ in interval $[t_j, t_{j+1}]$, with spline coefficients $x_{jk} \in \Re$, for $1 \le j \le n-1$ and $0 \le k \le 3$.

For practical purposes, given the coefficients, $x_{jk} \in \Re$, associated with $x(t) = [e_1, e_2, \ldots, e_n] \in \Re^n$, we only need to transform $x(t)$ into a new space as $x(t) = [x_{13},\ x_{12},\ x_{11},\ x_{10}, \ldots,\ x_{j3},\ x_{j2},\ x_{j1},\ x_{j0}, \ldots,\ x_{(n-1)3},\ x_{(n-1)2},\ x_{(n-1)1},\ x_{(n-1)0}] \in \Re^{4(n-1)}$. We can add or subtract polynomials given their coefficients, and the polynomials are continuously differentiable. This yields an analytical solution for $a_{\min}$ in Equation (3) as follows:

$$a_{\min} = -\frac{1}{T}\sum_{j=1}^{n-1}\sum_{k=0}^{3} \frac{(x_{jk} - y_{jk})(t_{j+1} - t_j)^{k+1}}{k+1} \tag{6}$$

Figure 1b shows a pairwise alignment, of the two initial profiles in Figure 1a, after applying the vertical shift $y(t) \leftarrow y(t) - a_{\min}$. The two aligned profiles cross each other many times, and the integrated error, Equation (4), is zero. In this example, from Equation (4), the horizontal t-axis will bisect a profile $x(t)$ into two halves with equal areas, when $x(t)$ is aligned to the t-axis. In Section 2.2, we use this property of Equation (4) to define the multiple alignment of a set of profiles.

2.2 Multiple expression profile alignment

Given a set $X = \{x_1(t), \ldots, x_s(t)\}$, we want to align the profiles in such a way that the integrated squared error between any two *vertically shifted* profiles is minimal. Thus, for any $x_i(t)$ and $x_j(t)$, we want to find the values of a_i and a_j that minimize

$$f_{a_i,a_j}\left(x_i(t), x_j(t)\right) = \int_0^T \{[x_i(t) - a_i] - [x_j(t) - a_j]\}^2 \mathrm{d}t,$$

where *both* $x_i(t)$ and $x_j(t)$ are shifted vertically by an amount a_i and a_j, respectively, in possibly different directions, whereas in the pairwise alignment of Equation (1), profile $y(t)$ is shifted towards a *fixed* profile $x(t)$. The multiple alignment process consists then of finding the values of $a_1, \ldots, a_s$ that minimize

$$F_{a_1,\ldots,a_s}\left(x_1(t), \ldots, x_s(t)\right) = \sum_{1 \le i < j \le s} f_{a_i,a_j}\left(x_i(t), x_j(t)\right), \tag{7}$$

We use Lemma 1 to find the values a_i and a_j, $1 \le i < j \le s$, that minimize $F_{a_1,\ldots,a_s}$.

LEMMA 1. *If $x_i(t)$ and $x_j(t)$ are pairwise-aligned each to a fixed profile, $z(t)$, then the integrated error $\int_0^T [\hat{x}_i(t) - \hat{x}_j(t)]\mathrm{d}t = 0$.*

PROOF. If $x_i(t)$ and $x_j(t)$ are pairwise-aligned each to $z(t)$, then from Equation (3), we have $a_{\min_i} = -\frac{1}{T}\int_0^T [z(t) - x_i(t)]\mathrm{d}t$ and $a_{\min_j} = -\frac{1}{T}\int_0^T [z(t) - x_j(t)]\mathrm{d}t$. Then, $\int_0^T [\hat{x}_i(t) - \hat{x}_j(t)]\mathrm{d}t$
$= \int_0^T \left[[x_i(t) - a_{\min_i}] - [x_j(t) - a_{\min_j}]\right]\mathrm{d}t$
$= \int_0^T x_i(t)\mathrm{d}t + \int_0^T [z(t) - x_i(t)]\mathrm{d}t - \int_0^T x_j(t)\mathrm{d}t - \int_0^T [z(t) - x_j(t)]\mathrm{d}t = 0.$ ■

In other words, $\hat{x}_j(t)$ is automatically aligned relative to $\hat{x}_i(t)$, given $z(t)$ is fixed.

COROLLARY 1. *If $x_i(t)$ and $x_j(t)$ are pairwise-aligned each to a fixed profile, $z(t)$, then $f_{a_{\min_i}, a_{\min_j}}\left(x_i(t), x_j(t)\right)$ is minimal.*

PROOF. This follows immediately from Lemma 1, which shows the property implied by the single vertical shift minimizing the integrated squared error, i.e. $\int_0^T [\hat{x}_i(t) - \hat{x}_j(t)]\mathrm{d}t = 0 \Rightarrow$
$\int_0^T \left[[x_i(t) - a_{\min_i}] - [x_j(t) - a_{\min_j}]\right]^2 \mathrm{d}t$ is minimal. ■

LEMMA 2. *If profiles $x_1(t), \ldots, x_s(t)$ are pairwise-aligned each to a fixed profile, $z(t)$, then $F_{a_{\min_1},\ldots,a_{\min_s}}\left(x_1(t), \ldots, x_s(t)\right)$ is minimal.*

PROOF. From Corollary 1,
$f_{a_i,a_j}\left(x_i(t), x_j(t)\right) \ge f_{a_{\min_i}, a_{\min_j}}\left(x_i(t), x_j(t)\right)$, with equality holding when $a_k = a_{\min_k}$, which is attained by aligning each $x_k(t)$ independently with $z(t)$, $1 \le k \le s$. From the definition of Equation (7), it follows that $F_{a_1,\ldots,a_s}\left(x_1(t), \ldots, x_s(t)\right) \ge \sum_{1 \le i < j \le s} f_{a_{\min_i}, a_{\min_j}}\left(x_i(t), x_j(t)\right)$
$= F_{a_{\min_1},\ldots,a_{\min_s}}\left(x_1(t), \ldots, x_s(t)\right)$, with equality holding when $a_k = a_{\min_k}$, $1 \le k \le s$. ■

Thus, given a fixed profile $z(t)$, applying Corollary 1 to all pairs of profiles minimizes $F_{a_1,\ldots,a_s}\left(x_1(t), \ldots, x_s(t)\right)$ in Equation (7).

THEOREM 1. *Given a fixed profile, $z(t)$, and a set of profiles, $X = \{x_1(t), \ldots, x_s(t)\}$, there always exists a multiple alignment, $\hat{X} = \{\hat{x}_1(t), \ldots, \hat{x}_s(t)\}$, such that*

$$\hat{x}_i(t) = x_i(t) - a_{\min_i},\ \textit{where, } a_{\min_i} = -\frac{1}{T}\int_0^T [z(t) - x_i(t)]\mathrm{d}t, \tag{8}$$

and, in particular, for profile $z(t)=0$, defined by the horizontal t-axis, we have

$$\hat{x}_i(t) = x_i(t) - a_{\min_i},\ \textit{where, } a_{\min_i} = \frac{1}{T}\int_0^T x_i(t)\mathrm{d}t. \tag{9}$$

PROOF. The proof follows from Corollary 1. Since each profile is aligned to a fixed profile, $z(t)$, it implies that we can either align each profile and $z(t)$ individually, or all profiles at a time, implying a 'universal' multiple alignment. ■

We use the multiple alignment of Equation (9) in all subsequent discussions. Using spline interpolations, each profile $x_i(t)$, $1 \le i \le s$, is a continuously integrable profile

$$x_i(t) = \begin{cases} x_{i,1}(t) & \text{if} \quad t_1 \le t \le t_2 \\ \cdots & \\ x_{i,j}(t) & \text{if} \quad t_j \le t \le t_{j+1} \\ \cdots & \\ x_{i,n-1}(t) & \text{if} \quad t_{n-1} \le t \le t_n \end{cases} \tag{10}$$

where, $x_{i,j}(t) = x_{ij3}(t-t_j)^3 + x_{ij2}(t-t_j)^2 + x_{ij1}(t-t_j)^1 + x_{ij0}(t-t_j)^0$ represents $x_i(t)$ in interval $[t_j, t_{j+1}]$, with spline coefficients x_{ijk} for $1 \le i \le s$, $1 \le j \le n-1$ and $0 \le k \le 3$. Thus, the analytical solution for $a_{\min_i}$ in Equation (9) is

$$a_{\min_i} = \frac{1}{T}\sum_{j=1}^{n-1}\sum_{k=0}^{3}\frac{x_{ijk}\left(t_{j+1}-t_j\right)^{k+1}}{k+1}. \tag{11}$$

2.3 Distance function

The distance between any two piecewise linear profiles is defined as follows:

$$d(x,y) = \frac{1}{T}f(a_{\min}) = \frac{1}{T}\int_0^T \{x(t) + a_{\min} - y(t)\}^2\mathrm{d}t. \tag{12}$$

For any function $\phi(t)$ defined on $[0,T]$, we also define

$$\langle\phi\rangle \triangleq \frac{1}{T}\int_0^T \phi(t)\mathrm{d}t. \tag{13}$$

Then, from Equations (1) and (3) we have

$$\begin{aligned} d(x,y) &= \frac{1}{T}\int_0^T \{[x(t)-y(t)]^2 + 2a_{\min}[x(t)-y(t)] + a_{\min}^2\}\mathrm{d}t \\ &= \frac{1}{T}\int_0^T \{[x(t)-y(t)]\}^2\mathrm{d}t - 2a_{\min}^2 + a_{\min}^2 \\ &= \left\langle [x(t)-y(t)]^2 \right\rangle - \langle x(t)-y(t)\rangle^2. \end{aligned} \tag{14}$$

By performing the multiple alignment of Equation (9) to obtain new profiles $\hat{x}(t)$ and $\hat{y}(t)$, we have

$$d(x,y) = \left\langle \left[\hat{x}(t)-\hat{y}(t)\right]^2 \right\rangle = \frac{1}{T}\int_0^T \{\hat{x}(t)-\hat{y}(t)\}^2\mathrm{d}t. \tag{15}$$

Thus, $d(x,y)^{\frac{1}{2}}$ is the 2-norm, satisfying all the properties of a metric. On the other hand, it is easy to show that $d(x,y)$ in Equation (15) does not satisfy the triangle inequality, and hence it is not a metric. We, however, use $d(x,y)$ in Equation (15) as our distance function, since it is algebraically easier to work with than the metric $d(x,y)^{\frac{1}{2}}$. Equation (15) is closer to the spirit of regression analysis, and thus, we can dispense with the requirement for the triangle inequality.

With the spline interpolations of Equation (5), we derived the analytical solution for $d(x,y)$ in Equation (15), using the symbolic computational package *Maple*[1]. Full details can be found in Subhani *et al.* (2009).

2.4 Centroid of a cluster

Given a set of profiles $X = \{x_1(t), \ldots, x_s(t)\}$, we aim to find a representative *centroid profile* $\mu(t)$, that well represents X. An obvious choice is the function that minimizes

$$\Delta[\mu] = \sum_{i=1}^{s} d\left(x_i, \mu\right), \tag{16}$$

where Δ plays the role of the *within-cluster-scatter* defined in Rueda *et al.* (2008), and the distance between two profiles, $x(t)$ and $y(t)$, is defined in Equation (15). The distance $d(\cdot,\cdot)$ as defined in Equation (15) is unchanged by an additive shift $x(t) \to x(t) - a$ in either of its arguments, and hence, is order-preserving; i.e.: $d(u,v) \le d(x,y)$ if and only if $d\left(\hat{u},\hat{v}\right) \le d\left(\hat{x},\hat{y}\right)$. Therefore, we have

$$\Delta[\mu] = \sum_{i=1}^{s} d\left(\hat{x}_i, \mu\right) = \frac{1}{T}\int_0^T \sum_{i=1}^{s}\{\hat{x}_i(t) - \mu(t)\}^2\mathrm{d}t, \tag{17}$$

where $\hat{X} = \left\{\hat{x}_1(t), \ldots, \hat{x}_s(t)\right\}$ is the multiple alignment of Equation (9). This is a *functional* of μ; i.e. a mapping from the set of real valued functions defined in $[0,T]$ to the set of real numbers. To minimize Δ with respect to μ, we set the functional derivative to zero[2]. This functional is of the form

$$F[\phi] = \int L(\phi(t))\mathrm{d}t, \tag{18}$$

for some function L, for which the functional derivative is simply $\frac{\delta F[\phi]}{\delta\phi(t)} = \frac{\mathrm{d}L(\phi(t))}{\mathrm{d}\phi(t)}$. In our case, we have

$$\frac{\delta\Delta[\mu]}{\delta\mu(t)} = -\frac{2}{T}\sum_{i=1}^{s}\left[\hat{x}_i(t) - \mu(t)\right] = -\frac{2}{T}\left(\sum_{i=1}^{s}\hat{x}_i(t) - s\mu(t)\right) \tag{19}$$

Setting $\frac{\delta\Delta[\mu]}{\delta\mu(t)} = 0$ gives

$$\mu(t) = \frac{1}{s}\sum_{i=1}^{s}\hat{x}_i(t). \tag{20}$$

With the spline coefficients, x_{ijk}, of each $x_i(t)$ interpolated as in Equation (10), the analytical solution for $\mu(t)$ in Equation (20) is

$$\mu_j(t) = \frac{1}{s}\sum_{i=1}^{s}\left[\sum_{k=0}^{3}x_{ijk}\left(t-t_j\right)^k\right] - a_{\min_i}, \tag{21}$$

in each interval $[t_j, t_{j+1}]$. Equation (20) applies to aligned profiles whereas Equation (21) can also apply to unaligned profiles.

3 ALGORITHMS

3.1 *k*-means clustering via multiple alignment

Our approach allows us to apply a clustering algorithm such as k-means or EM, which, though not optimal, provide a fast and practical solution to the problem. In k-means (Xu and Wunsch, 2008), we want to partition a set of s profiles, $\mathcal{D} = \{x_1(t), \ldots, x_s(t)\}$, into k disjoint clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$, $1 \le k \le s$; such that (i) $\mathcal{C}_i \ne \emptyset, i = 1, \ldots, k$; (ii) $U_{i=1}^{k}\mathcal{C}_i = \mathcal{D}$; and (iii) $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$; $i \ne j$; $i,j = 1, \ldots, k$.

[1] All the analytical solutions in this article were verified with Maple.

[2] For a functional $F[\phi]$, the functional derivative is defined as $\frac{\delta F[\phi]}{\delta\phi(t)} = \lim_{\epsilon\to 0}\frac{(F[\phi+\epsilon\delta_t] - F[\phi])}{\epsilon}$, where $\delta_t(\tau) = \delta(\tau - t)$ is the Dirac delta function centered at t.

Also, each profile is assigned to the cluster whose mean is the closest. It assumes that the object features form a *vector space*. Let $U=\{u_{ij}\}$ be the membership matrix

$$u_{ij}=\begin{cases}1 & \text{if } d(x_i,\mu_j)=\min_{l=1,\ldots,k} d(x_i,\mu_l), \text{where } i=1,\ldots,s\\ 0 & \text{otherwise}\end{cases} \tag{22}$$

The aim is to minimize the sum of squared distances: $J(\theta,U)=\sum_{i=1}^{n}\sum_{j=1}^{k}u_{ij}d(x_i,\mu_j)$, where $\theta=\mu_1,\mu_2,\ldots,\mu_n$.

Algorithm 1 *k-MCMA: k-means clustering with multiple alignment*

Require: Set of profiles, $\mathcal{D}=\{x_1(t),\ldots,x_s(t)\}$, and desired number of clusters, k

Ensure: Clusters $\hat{\mathcal{C}}_{\hat{\mu}_1},\ldots,\hat{\mathcal{C}}_{\hat{\mu}_k}$

1. apply natural cubic spline interpolation on $x_i(t)\in\mathcal{D}$, for $1\le i\le k$ (see Section 2.1)
2. multiple-align transformed $\mathcal{D}$ to obtain $\hat{\mathcal{D}}=\{\hat{x}_1(t),\ldots,\hat{x}_s(t)\}$, using Equation (9)
3. randomly initialize centroid $\hat{\mu}_i(t)$, for $1\le i\le k$
4. **repeat**
5.
6. assign $\hat{x}_j(t)$ to cluster $\hat{\mathcal{C}}_{\hat{\mu}_i}$ with minimal $d(\hat{x}_j,\hat{\mu}_i)$, for $1\le j\le s$ and $1\le i\le k$
7. update $\hat{\mu}_i(t)$ of $\hat{\mathcal{C}}_{\hat{\mu}_i}$, for $1\le i\le k$
8. **until** Convergence: that is, no change in $\hat{\mu}_i(t)$, for $1\le i\le k$

return Clusters $\hat{\mathcal{C}}_{\hat{\mu}_1},\ldots,\hat{\mathcal{C}}_{\hat{\mu}_k}$

In k-MCMA (see Algorithm 1), we first multiple-align the set of profiles $\mathcal{D}$, using Equation (9), and then cluster the multiple aligned $\hat{\mathcal{D}}$ with k-means. Recall that the process of Equation (9) is to *pairwise-align* each profile with the t-axis. The k initial centroids are found by randomly selecting k pairs of profiles in $\hat{\mathcal{D}}$, and then take the centroid of each pair. In Step (4.6), we do not use pairwise alignment to find the centroid $\hat{\mu}_i(t)$ closest to a $\hat{x}_j(t)$; since, by Lemma 1, they are automatically aligned relative to each other. When profiles are multiple-aligned, any arbitrary distance function other than Equation (15) can be used in Step (4.6), including the Euclidean distance. Also, by Theorem 2 below, there is no need to multiple-align $\hat{\mathcal{C}}_{\hat{\mu}_i}$ in Step (4.7), to update its centroid $\hat{\mu}_i(t)$.

THEOREM 2. *Let $\bar{\mu}(t)$ be the centroid of a cluster of m multiple-aligned profiles. Then $\hat{\bar{\mu}}(t)=\bar{\mu}(t)$.*

PROOF. We have $\hat{\bar{\mu}}(t)=\bar{\mu}(t)-a_{\min_{\bar{\mu}}}$. However, $a_{\min_{\bar{\mu}}}=\frac{1}{T}\int_0^T\bar{\mu}(t)\mathrm{d}t=\frac{1}{T}\int_0^T\frac{1}{m}\sum_{i=1}^{m}\hat{x}_i(t)=0$, since each $\hat{x}_i(t)$ is aligned with the t-axis. ■

Thus, Lemma 1 and Theorem 2 make k-MCMA much faster than applying k-means directly on the non-aligned dataset $\mathcal{D}$. An important implication of Equation (15) is that applying k-means on the non-aligned dataset $\mathcal{D}$ (i.e. clustering on $\mathcal{D}$), without any multiple alignment, is equivalent to k-MCMA (i.e. clustering on $\hat{\mathcal{D}}$). That is, if a profile $x_i(t)$ is assigned to a cluster $\mathcal{C}_{\mu_i}$ by k-means on $\mathcal{D}$, its shifted profile $\hat{x}_i(t)$ will be assigned to cluster $\hat{\mathcal{C}}_{\hat{\mu}_i}$ by k-MCMA (k-means on $\hat{\mathcal{D}}$). This can be easily shown by the fact that multiple alignment is order-preserving. In k-means on $\mathcal{D}$, Step (6) would require $O(sk)$ *pairwise alignments* to assign s profiles to k clusters; whereas no pairwise alignment is needed in k-MCMA. In other words, we show that we can multiple-align *once*, and obtain the *same* k-means clustering results, provided that we initialize the means in the same manner. This also, reinforces a known fact demonstrated in Roth *et al.* (2003); which is a dissimilarity function that is not a metric can be made metric by using a shift operation. In this case, the objective function of k-means does not change, and convergence is assured. Thus, this saves a lot of computations.

3.2 EM clustering via multiple alignment

In this section, we present the EM clustering algorithm combined with the alignment methods. EM is used for clustering in the context of mixture models (Dempster *et al.*, 1977). The goal of EM clustering is to estimate the means and covariances for each cluster so as to maximize the likelihood of the observed data distribution. In EM, we want to partition a set of s profiles, $\mathcal{D}=\{x_1(t),\ldots,x_s(t)\}$, into k disjoint clusters $\mathcal{C}_1,\ldots,\mathcal{C}_k$, $1\le k\le s$, such that: (i) $\mathcal{C}_i\neq\emptyset, i=1,\ldots,k$; (ii) $\bigcup_{i=1}^{k}\mathcal{C}_i=\mathcal{D}$; and (iii) $\mathcal{C}_i\cap\mathcal{C}_j=\emptyset, i,j=1,\ldots,k$ and $i\neq j$. Let $\mathcal{D}$ be the complete-data space drawn independently from the mixture density

$$\textbf{E-step: } p(x|\theta)=\sum_{i=1}^{k}p(x|\mathcal{C}_i,\theta_i)P(\mathcal{C}_i) \tag{23}$$

where parameter $\theta=[\theta_1,\ldots,\theta_k]^t$ is fixed but unknown and $P(\mathcal{C}_i)$ the known posterior probability of class $\mathcal{C}_i$. The aim is to maximize the likelihood

$$\textbf{M-step: } p(D|\theta)=\prod_{e=1}^{s}p(x_e|\theta) \tag{24}$$

We consider normal distributions, $p(x_k|\mathcal{C}_i,\theta_i)\sim N(\mu_i,\Sigma_i)$, where $\theta_i=[\mu_i,\Sigma_i]^t$; μ_i and Σ_i are the means and the covariances of classes, respectively. Both steps iterate until the log-likelihood reaches a maximum. Thus, EM assigns profiles to multiple clusters, as in *fuzzy* clustering. Also, unlike in k-means, EM assigns each profile to the cluster that finds the maximum posterior probability.

Algorithm 2 *EMMA: EM clustering with multiple alignment*

Require: Set of profiles, $\mathcal{D}=\{x_1(t),\ldots,x_s(t)\}$, and desired number of clusters, k

Ensure: Clusters $\hat{\mathcal{C}}_{\hat{\mu}_1},\ldots,\hat{\mathcal{C}}_{\hat{\mu}_k}$

1. apply natural cubic spline interpolation on $x_i(t)\in\mathcal{D}$, for $1\le i\le s$ (see Section 2.1)
2. multiple-align transformed $\mathcal{D}$ to obtain $\hat{\mathcal{D}}=\{\hat{x}_1(t),\ldots,\hat{x}_s(t)\}$, using Equation (9)
3. initialize centroid $\hat{\mu}_i(t)$, for $1\le i\le k$
4. compute the initial log-likelihood (see Equation (24))
5. **repeat**
6. **E-step:** $p(x|\theta)=\sum_{i=1}^{k}p(x|\hat{\mathcal{C}}_{\hat{\mu}_i},\theta_i)P(\hat{\mathcal{C}}_{\hat{\mu}_i})$
7. assign $\hat{x}_j(t)$ to cluster $\hat{\mathcal{C}}_{\hat{\mu}_i}$ with maximum log-likelihood, for $1\le j\le s$ and $1\le i\le k$
8. **M-step:** $p(D|\theta)=\prod_{e=1}^{s}p(x_e|\theta)$
9. **until** The log-likelihood reaches its maximum
10. **return** Clusters $\hat{\mathcal{C}}_{\hat{\mu}_1},\ldots,\hat{\mathcal{C}}_{\hat{\mu}_k}$

In EMMA (see Algorithm 2), we first multiple-align the set of profiles $\mathcal{D}$, using Equation (9), and then cluster the multiple-aligned $\hat{\mathcal{D}}$ with EM. Recall that the process of Equation (9) is to *pairwise-align* each profile with the t-axis. The k centroids can be initialized randomly in Step (3) of EMMA, or by any initialization approach. However, to obtain better clustering results with EMMA, it is necessary to start with near-optimal centroids; thus, we applied

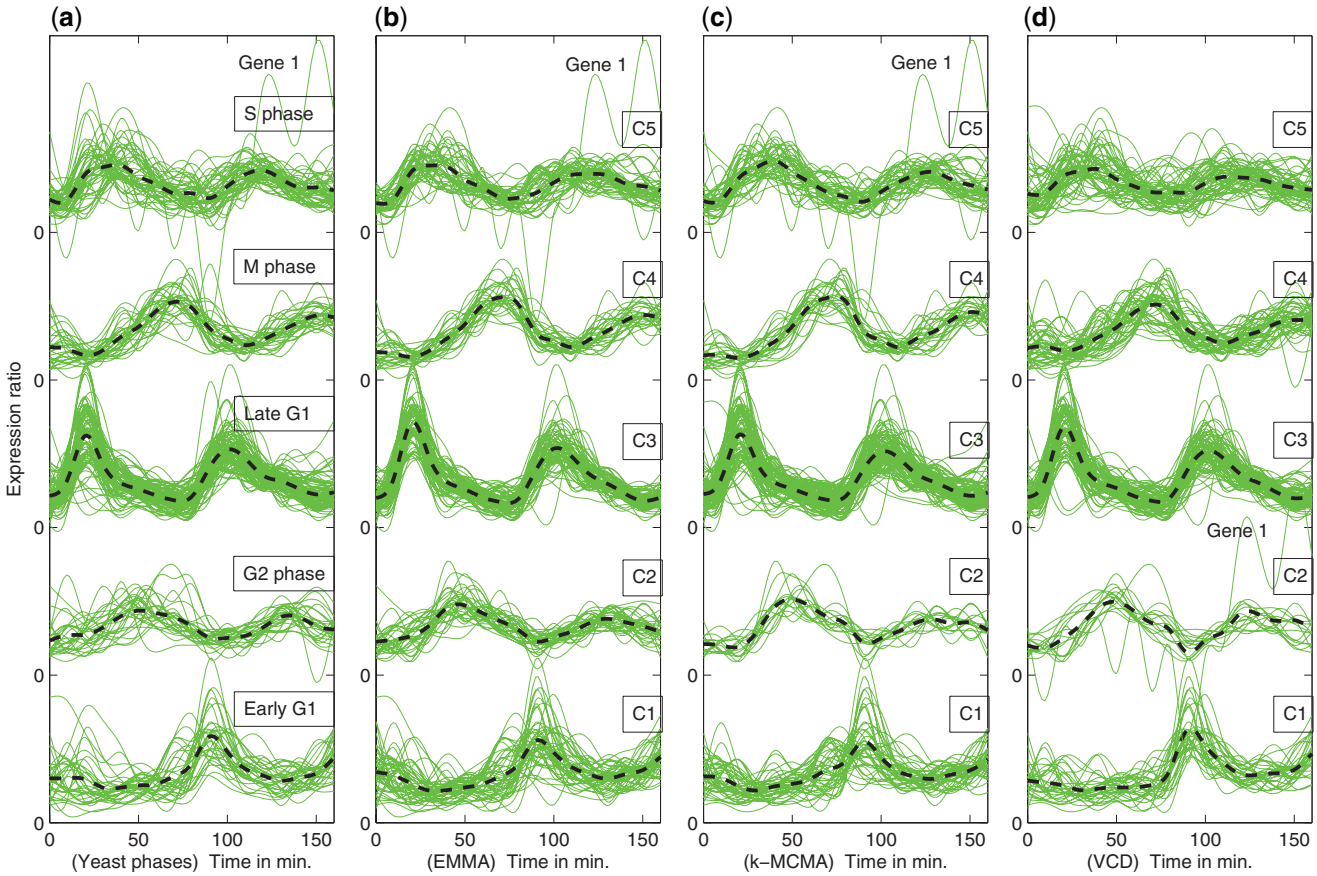

Fig. 2. (**a**) EMMA clusters, (**b**) *S.cerevisiae* phases (Cho *et al.*, 1998), (**c**) *k*-MCMA clusters and (**d**) VCD clusters, with centroids shown.

k-MCMA to generate the k initial centroids in Step (3) (we have also tried different initialization methods but with less success).

By Theorem 2, there is also no need to multiple-align a cluster $\hat{\mathcal{C}}_{\hat{\mu}_i}$ in Step (6) of EMMA, for updating its centroid $\hat{\mu}_i(t)$. Likewise, any arbitrary distance function can be used in Step (6), for computing the centroids. EMMA is not a distance-based clustering method. Nevertheless, the quantities $p(x|\theta)$, $p(x|\hat{\mathcal{C}}_{\hat{\mu}_i}, \theta_i)$, $P(\hat{\mathcal{C}}_{\hat{\mu}_i})$ and $p(D|\theta)$ are also preserved when the distances are preserved.

To conclude this section, we note that all the above theoretical results on Natural Cubic Spline representation of profiles and clustering algorithms that are proposed in the previous section also apply to piecewise linear representations of time-series profiles.

4 RESULTS AND DISCUSSION

One of the datasets, the pre-clustered budding yeast genes of Cho *et al.* (1998), contains time-series gene expression profiles of the complete characterization of mRNA transcript levels during the yeast cell cycle. These experiments measure the expression levels of the 6220 yeast genes during the cell cycle at 17 time points, from 0 to 160 min. From those gene profiles, 221 profiles were analyzed, and normalized as in Cho *et al.* (1998), i.e. dividing each transcript level by the mean value of the profile. The dataset contains five *known* clusters called *phases*: early G1 phase (32 genes), late G1 phase (84 genes), S phase (46 genes), G2 phase (28 genes) and M phase (31 genes); the phases are visualized in Figure 2a. Another dataset used in the experiments is the *Pseudomonas aeruginosa*, which contains expressions of the transcriptomes from planktonic clusters at eight different points, from 0 to 48 h (Waite *et al.*, 2006). The clustering methods were also tested on the dataset containing the expressions of the cell-cycle progressions of the fission yeast, *Schizosaccharomyces pombe* (Peng *et al.*, 2005). This dataset contains 747 genes, representing the expression ratios measured at 14 different time points, for two types of cells, namely, wild-type and cdc25 mutant cells.

Setting $k=5$, we applied *k*-MCMA and EMMA on the yeast dataset to see if *k*-MCMA and EMMA are able to find these phases as accurately as possible. Once the clusters have been found, to compare the clustering with the pre-clustered dataset of Cho *et al.* (1998), the next step is to label the clusters, where the labels are the 'phases' in the pre-clustered dataset. To measure the performance of *k*-MCMA and EMMA, we assigned each EMMA cluster to a yeast phase using the *Hungarian algorithm* (Kuhn, 2005). The Hungarian method is a combinatorial optimization algorithm, which solves the assignment problem in polynomial time. Our phase assignment problem and the complete discussion of the solution can be found in Subhani *et al.* (2009). In Figure 2, the cluster and the phase of each of the five selected pairs, found by the Hungarian algorithm, are shown at the same level; e.g. cluster *C3* of Figure 2b–d is assigned to the *late G1* phase of Cho *et al.* (1998) by our phase assignment

approach, and hence they are at the same level in the figure. The same procedure by using k-MCMA of Subhani *et al.* (2009) and the results are in Figure 2c.

The five clusters found by EMMA are shown in Figure 2b and those found by k-MCMA are shown in Figure 2c, while the corresponding phases of Cho *et al.* (1998) after the phase assignment are shown in Figure 2a. The horizontal axis represents the time points in minutes and the vertical axis represents the expression values. The dashed black lines are the *cluster centroids* learned by EMMA (Fig. 2b) and the *known phase centroids* of the yeast data (Fig. 2a). In the figure, each cluster and phase were multiple-aligned using Equation (9) to enhance visualization. Figure 2 clearly shows a high degree of similarity between the EMMA clusters and the yeast phases. Visually, each EMMA cluster on the left is *very similar* to exactly one of the yeast phases, which we show at the same level on the right. Also visually, it even *'seems'* that EMMA clusters are more accurate than the yeast phases and k-MCMA clusters, which suggests that EMMA can also correct manual phase assignment errors, if any.

An objective measure for comparing EMMA and k-MCMA clusters with the yeast phases was computed as follows. For each EMMA or k-MCMA cluster, $\hat{\mathcal{C}}_{\hat{\mu}_c}$ $(1 \le c \le k = 5)$, we find the shortest distance between each profile $x_i(t)$, $1 \le i \le |\hat{\mathcal{C}}_{\hat{\mu}_c}|$, and all five-phase centroids $v_j(t)$, $1 \le j \le k = 5$, using Equation (16) of Subhani *et al.* (2009). Profile $x_i(t)$ will be assigned the *correct* label (i.e. assigned to phase label of $\hat{\mathcal{P}}_{\hat{v}_j}$) whenever $x_i(t) \in \hat{\mathcal{P}}_{\hat{v}_j}$ and $\left(\hat{\mathcal{C}}_{\hat{\mu}_c}, \hat{\mathcal{P}}_{\hat{v}_j}\right) \in \mathcal{S}$ the set of selected cluster-phase pairs; otherwise, $x_i(t)$ will be assigned the *incorrect* label, if cluster $\hat{\mathcal{C}}_{\hat{\mu}_c}$ was not paired with phase $\hat{\mathcal{P}}_{\hat{v}_j}$ by our pair-assignment method. The percentage of *correct* assignments over the 221 profiles was used as our measure of accuracy, resulting in 83.26% for EMMA and 79.64% for k-MCMA. That is

$$\text{Acc} = \frac{\sum_{c=1}^{k} \sum_{i=1}^{|\hat{\mathcal{C}}_{\hat{\mu}_c}|} E\left(c\ ,\ \text{argmin}_{1 \le j \le k}\, d\left(x_i, v_j\right)\right)}{221}, \qquad (25)$$

where $E(a, b)$ returns 1 when $a = b$, and zero otherwise. This criterion is reasonable, as k-MCMA is an unsupervised learning approach that does not know the phases beforehand, and hence the aim is to *'discover'* the phases. In Cho *et al.* (1998), the five phases were determined using biological information, including genomic and phenotypic features observed in the yeast cell-cycle experiments. EMMA's accuracy of 83.26% is quite high considering that it is an *unsupervised* learning method.

Comparison with previous approaches We have compared our approaches with the following two previously published approaches: (i) a clustering method that uses piecewise linear profiles in Rueda *et al.* (2008); and (ii) the variation-based coexpression detection (VCD) algorithm, which is described in Zong-Xian and Jung-Hsien (2008).

We used an objective measure for comparing EMMA clusters with the yeast phases. The measurement was computed by taking the average classification accuracy, as the number of genes that EMMA *correctly* assigned to one of the phases. Considering each EMMA cluster as a class, $\hat{\mathcal{C}}_{\hat{\mu}_c}$ $(1 \le c \le k = 5)$, we trained a c-nearest neighbor (c-NN) classifier with clusters to classify the data with a 10-fold cross validation procedure, where c is the number of nearest profiles from the centroids. We found that $k = 5$ is the best number of clusters for the dataset, and we used the distance function of Equation 15 to measure the distance between the centroids and the nearest profiles. We applied the same procedure for k-MCMA clusters too. In Cho *et al.* (1998), the five phases were determined using biological information, including genomic and phenotypic features observed in the yeast cell-cycle experiments. EMMA's average classification accuracy is 91.03%, whereas for k-MCMA it is 89.51%.

We also applied the same objective measure as described above for comparing the EMMA clusters with the *P.aeruginosa* dataset, obtaining an average classification accuracy of 91.40%. In Waite *et al.* (2006) and Rueda *et al.* (2008), the correlation coefficient is used as the distance measure between gene profiles while here, we used the distance as defined in Equation (15). Figure 3 shows that EMMA yields better results than k-MCMA and the methods used in Waite *et al.* (2006). The same objective measure as described above was applied in order to compare the k-MCMA clusters, using piecewise linear profiles (an approach of Rueda *et al.*, 2008) with the yeast phases, which yielded an average classification accuracy of 86.12%. For the *P. aeruginosa* dataset, we obtained an average classification accuracy of 90.90%. Table 1 shows the average classification accuracies of our approaches and the approach of Rueda *et al.* (2008).

From Table 1 and Supplementary Figure 1, we observe that natural cubic spline profiles performed better than piecewise linear profiles. k-MCMA and EMMA clusters using natural cubic spline profiles on both datasets obtained over 90% classification accuracy, which is very high considering that they are both *unsupervised* methods, while EMMA yields better results than k-MCMA. The same comparison was carried out against the VCD algorithm (Zong-Xian and Jung-Hsien, 2008). In that approach, gene expressions are translated into gene variation vectors whose cosine values are then used to evaluate the variation vector similarities over time. EMMA and VCD are compared on two datasets: *Saccharomyces cerevisiae* and *S.pombe* datasets (Table 2).

In Figure 2, cluster number 5 of EMMA and k-MCMA are similar to the corresponding S phase, whereas VCD assigns many differentially expressed genes to it—the same situation occurs for cluster number 2 as well. From the figure, we observe that EMMA's clusters are more compact than those of all other methods. EMMA's clusters are even more well-separated than pre-characterized phases, at least visually. In Figure 4, VCD identified three clusters that contain only two genes and many genes are assigned to incorrect clusters. In this dataset, k-MCMA's clusters are more well-separated than EMMA's.

On the *S.cerevisiae* dataset, both EMMA and VCD found five clusters, whereas EMMA clusters obtained over 90% classification accuracy. On the *S.pombe* dataset, we ran EMMA in conjunction with four validity indices. We found eight meaningful clusters in this dataset. EMMA was applied by setting of $k = 8$, and yielded 89.53% classification accuracy. Setting $\lambda = 0.59$ and $z_p = 7$, we applied VCD on *S.pombe* as well to find the clusters. VCD also identified eight clusters and yielded 70.46% classification accuracy. VCD identified 33 unique genes that do not belong to any cluster. According to Peng *et al.* (2005), 71 clusters were obtained in the *S.pombe* dataset with parameters $\lambda = 0.75$ and $z_p = 1.96$. In their method, λ covers the similarity between sets and z_p determines the number of clusters. The eight EMMA and k-MCMA clusters on *S.pombe* yielded 86.94% and 87.63% classification accuracies, respectively, which shows that *S.pombe* contains eight meaningful clusters. In fact, EMMA and VCD are both *unsupervised* learning methods, while EMMA performs much better than VCD.

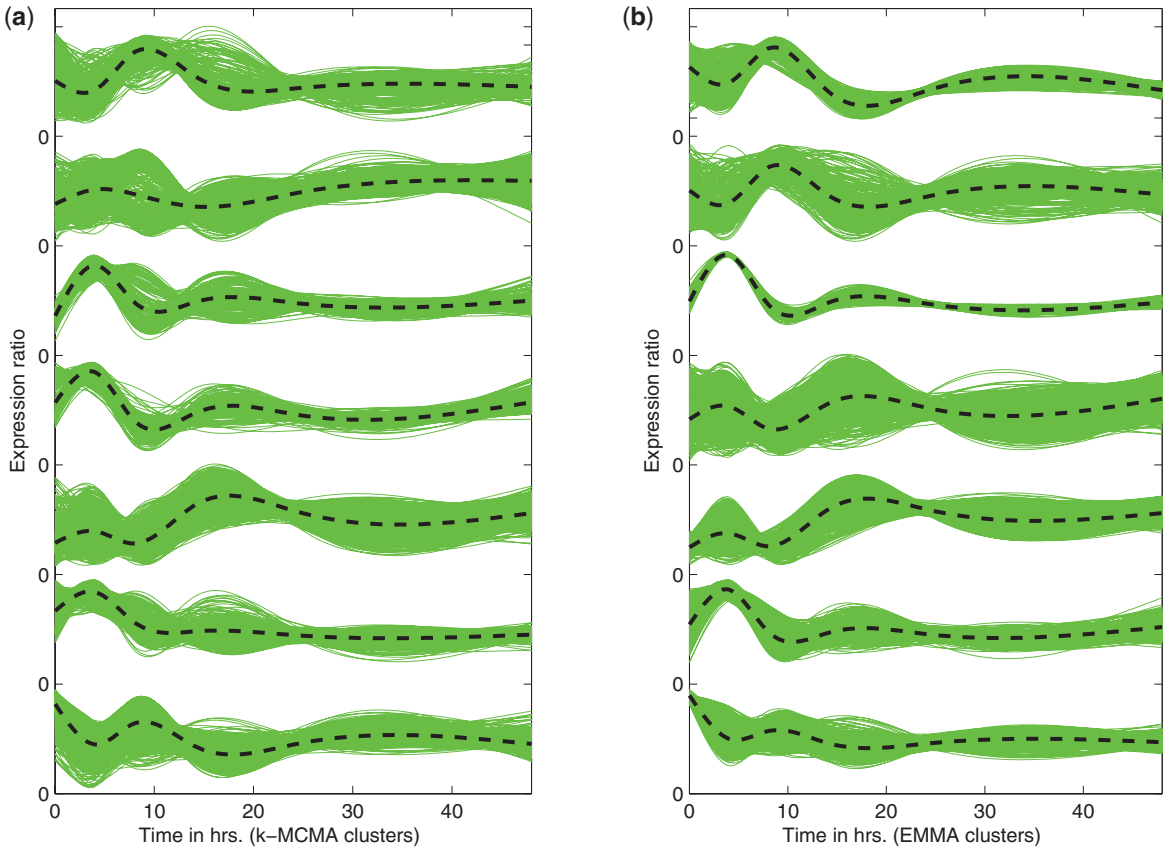

Fig. 3. (**a**) *k*-MCMA clusters and (**b**) EMMA clusters of *P.aeruginosa* dataset, with centroids shown.

Table 1. Experiment results overview of *k*-MCMA and EMMA with piecewise linear profile of Rueda *et al.* (2008)

Profiles	Approaches	*Saccharomyces cerevisiae* (%)	*Pseudomonas aeruginosa* (%)
NCS	*k*-MCMA	89.51	91.40
PL	*k*-MCMA	86.12	90.90
NCS	EMMA	91.03	92.71
PL	EMMA	86.43	89.37

NCS: natural cubic spline; PL: piecewise linear profiles.

Table 2. Experiment results overview of EMMA approach and the VCD method of Zong-Xian and Jung-Hsien (2008)

Approaches	*Saccharomyces cerevisiae* (%)	*Schizosaccharomyces pombe* (%)
k-MCMA	89.51	87.63
EMMA	91.03	86.94
VCD	80.68	70.46

5 CONCLUSION

We propose *k*-MCMA and EMMA, two methods that combine *k*-means and EM with multiple profile alignment of gene expression profiles to cluster microarray time-series data. The profiles are represented as natural cubic spline functions, where the expression measurements are not necessarily taken at regular time-intervals. Four cluster validity indices are used in conjunction with the above methods to determine the appropriate number of clusters and also the validity of the clusters. An objective measure for comparing the *k*-MCMA and EMMA clusters with the yeast phases is computed by taking the average classification accuracy. EMMA combined with natural cubic spline profiles performs better than piecewise linear profiles, and also outperformed VCD. Our experiments also show that EMMA is able to find better clusters than biologically characterized yeast phases. We finally note that our vertical alignment method is different from the temporal alignment of Bar-Joseph *et al.* (2003) and Ernst *et al.* (2005), where the alignment is horizontal, i.e. it is performed along the time axis to match the time points of one profile to the time points of the other profile, in such a way that the integrated squared error between the horizontally aligned profiles is minimal. Temporal alignment is used for profiles that are either sampled at different time points, have different number of time points, or have different time extents.

In the future, we plan to study other distance-based clustering approaches using our multiple alignment method. Other clustering algorithms with multiple alignment, cluster validity indices based on multiple alignment, phase detection by aligning over a portion of the time series expression and studying the effectiveness of our clustering methods in dose–response microarray datasets can also

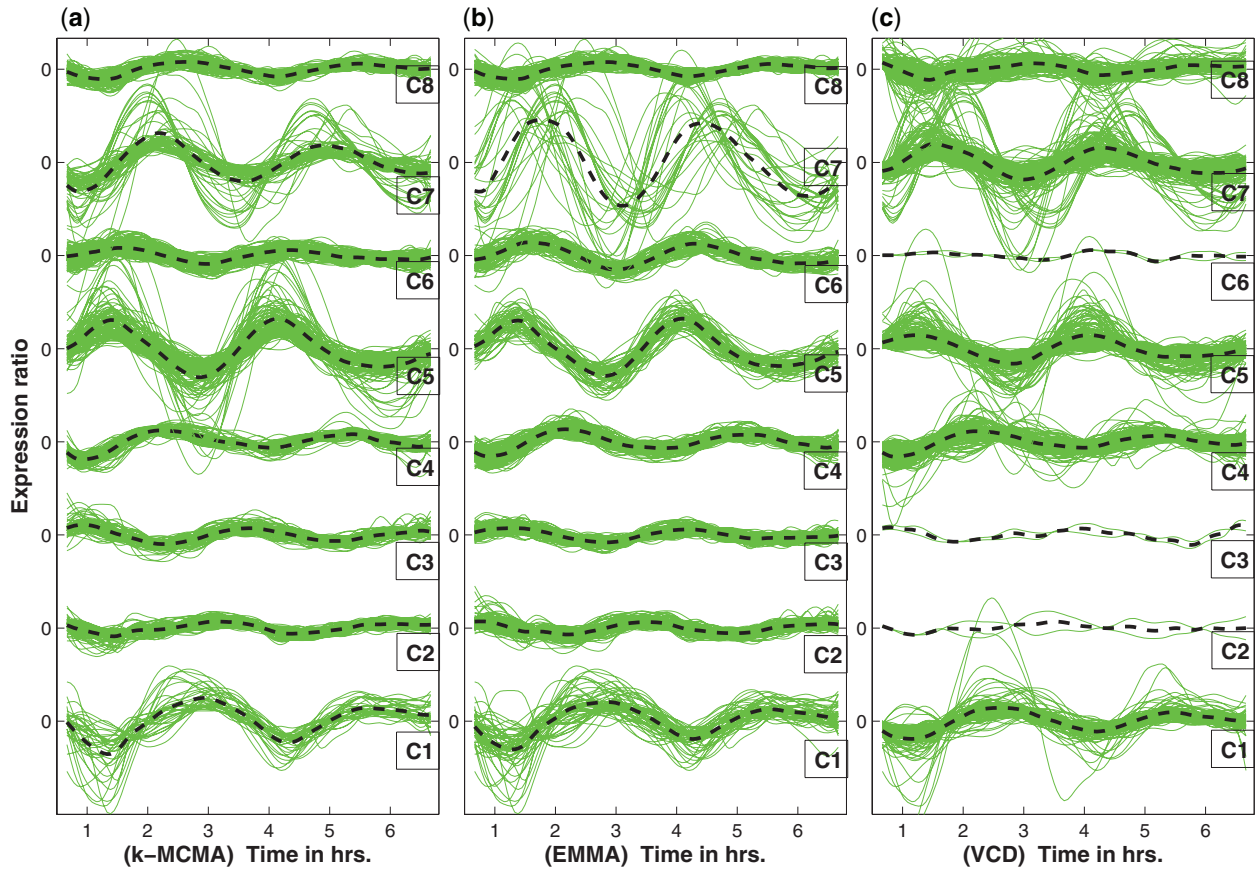

Fig. 4. (**a**) *k*-MCMA clusters, (**b**) EMMA clusters and (**c**) VCD clusters on *S.pombe* dataset, with centroids shown.

be interesting to investigate. Though our main focus is clustering, the effect of using different imputation methods rather than natural cubic spline in representing the profiles are also worth investigating.

Funding: Natural Sciences and Engineering Research Council of Canada, Grants #RGPIN228117-2006 and #RGPIN261360-2009 (partial).

Conflict of Interest: none declared.

REFERENCES

Bar-Joseph,Z. *et al*. (2003) Continuous representations of time series gene expression data. *J. Comput. Biol.*, **10**, 341–356.

Bréhélin,L. (2005) Clustering gene expression series with prior knowledge. *Lect. Notes Comput. Sci.*, **3692**, 27–38.

Cho,R. *et al*. (1998) A genome-wide transactional analysis of the mitotic cell cycle. *Mol. Cell*, **2**, 65–73.

Chu,S. *et al*. (1998) The transcriptional program of sporulation in budding yeast. *Science*, **282**, 699–705.

Déjean,S. *et al*. (2007) Clustering time-series gene expression data using smoothing spline derivatives. *EURASIP J. Bioinform. Syst. Biol.*, **70561**, 705–761.

Dempster,A. *et al*. (1977) Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc.*, **39**, 1–38.

Ernst,J. *et al*. (2005) Clustering short time series gene expression data. *Bioinformatics*, **21** (Suppl. 1), i159–i168.

Heyer,L. *et al*. (1999) Exploring expression data: identification and analysis of coexpressed genes. *Genome Res.*, **9**, 1106–1115.

Kuhn,H. (2005) The hungarian method for the assignment problem. *Nav. Res. Logist.*, **52**, 7–21.

Moller-Levet,C. *et al.* (2005) Clustering of unevenly sampled gene expression time-series data. *Fuzzy sets Syst.*, **152**, 49–66.

Peddada,S. *et al.* (2005) Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference. *Bioinformatics*, **19**, 834–841.

Peng,X. *et al*. (2005) Identification of cell cycle-regulated genes in fission yeast. *Mol. Biol. Cell*, **16**, 1026–1042.

Ramoni,M. *et al.* (2002) Cluster analysis of gene expression dynamics. *Proc. Natl Acad. Sci. USA*, **99**, 9121–9126.

Roth, V., Laub, J., Kawanabe, M., and Buhmann, J. (2003). Optimal cluster preserving embedding of nonmetric proximity data. *IEEE Trans. Pattern Anal. Mach. Intell.*, **25**, 1540–1551.

Rueda,L. *et al*. (2008) Clustering time-series gene expression data with unequal time intervals. *Springer Trans. Comput. Syst. Biol. X, LNBI*, **5410**, 100–123.

Subhani,N. *et al.* (2009) Microarray time-series data clustering via multiple alignment of gene expression profiles. In *Fourth IAPR International Conference on Pattern Recognition in Bioinformatics*. Vol. 5780 of *Lecture Notes in Bioinformatics*, Springer, New York, USA, pp. 377–390.

Tamayo,P. *et al.* (1999) Interpreting patterns of gene expression with soms: methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA*, **96**, 9121–9126.

Tavazoie,S. *et al.* (1999) Systematic determination of genetic network architecture. *Nat. Genet.*, **22**, 281–285.

Waite,R. *et al.* (2006) Clustering of pseudomonas aeruginosa transcriptomes from planktonic cultures, developing and mature biofilms reveals distinct expression profiles. *BMC Genomics*, **7**, 162–175.

Xu,R. and Wunsch,D. (2008) *Clustering*. Wiley-IEEE Press, New Jersey.

Zong-Xian,Y. and Jung-Hsien,C. (2008) Novel algorithm for coexpression detection in time-varying microarray data sets. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **5**, 120–135.