Microarray Time-Series Data Clustering via Multiple Alignment of Gene Expression Profiles

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

[1] School of Computer Science, 5115 Lambton Tower, University of Windsor, 401 Sunset Avenue, Windsor, Ontario, N9B 3P4, Canada
{hoque4,angom,lrueda}@uwindsor.ca
[2] Centre for Bioinformation Science, Mathematical Sciences Institute and John Curtin School of Medical Research, The Australian National University, Canberra, ACT 0200, Australia
conrad.burden@anu.edu.au

Abstract. Genes with similar expression profiles are expected to be functionally related or co-regulated. In this direction, clustering microarray time-series data via pairwise alignment of piece-wise linear profiles has been recently introduced. We propose a k-means clustering approach based on a multiple alignment of natural cubic spline representations of gene expression profiles. The multiple alignment is achieved by minimizing the sum of integrated squared errors over a time-interval, defined on a set of profiles. Preliminary experiments on a well-known data set of 221 pre-clustered *Saccharomyces cerevisiae* gene expression profiles yields excellent results with 79.64% accuracy.

Keywords: Microarrays, Time-Series Data, Gene Expression Profiles, Profile Alignment, Cubic Spline, k-Means Clustering.

1 Introduction

Clustering microarray time-series data is an important process in functional genomic studies, where genes with similar expression profiles are expected to be functionally related [1]. Many clustering methods have been developed in recent years [2,3,4,5,6].

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) [3]. A Bayesian approach in [7], partitional clustering based on k-means in [8] and an Euclidean distance approach in [9] have been proposed for clustering time-series gene expression profiles. They have applied self-organizing maps (SOMs) to visualize and to interpret gene temporal expression profile patterns. Also, the methods proposed in [4,10] 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 [10]. Specifying expression levels for the candidate profiles in advance for these correlation-based procedures requires estimating each candidate profile, which is made using a small sample of arbitrarily

V. Kadirkamanathan et al. (Eds.): PRIB 2009, LNBI 5780, pp. 377–390, 2009.

selected genes. The resulting clusters depend upon the initially chosen template genes, because there is a possibility of missing important genes. A regression-based method, which is suitable for analyzing single or multiple microarrays was proposed in [6] to address the challenges in clustering short time-series expression datasets.

Analyzing gene temporal expression profile datasets that are non-uniformly sampled and can contain missing values has been studied in [2]. Statistical spline estimation was used to represent temporal expression profiles as continuous curves. Clustering temporal gene expression profiles was studied by identifying homogeneous clusters of genes in [5]. 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 [11]. In [12], 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 this approach, two gene's expression profiles fall into the same cluster, if they show similar profiles in terms of directions of the changes of expression ratios, regardless of how big or small the changes are. In [13], 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 method, combined with an area-based distance measure between two aligned profiles, was used to cluster microarray time-series data. We re-formulate the profile alignment problem of [13] in terms of integrals of arbitrary functions, allowing us to generalize from a piecewise linear interpolation to any type of interpolation one believes be more physically realistic. The expression measurements are basically snapshots taken at time-points chosen by the experimental biologist. The cells expressing genes do not know when the biologist is going to choose to measure gene expression, which one would guess is changing continuously and smoothly all the time. Thus, smooth spline curve through the known time-points in the cell's expression path would be a better guess. We use natural cubic spline interpolation to represent each gene expression profile; also, it gives a handy way to align profiles for which measurements were not taken at the same time-points. We generalize the pairwise expression profile alignment formulae of [13] from the case of piece-wise linear profiles to profiles which are any continuous integrable functions on a finite interval. Next, we extend the concept of pairwise alignment to multiple expression profile alignment, where profiles from a given set are aligned in such a way that the sum of integrated squared errors, over a time-interval, defined on the set is minimized. Finally, we combine k-means clustering with our multiple alignment approach to cluster microarray time-series data.

2 Pairwise Expression Profile Alignment

Clustering time-series expression data with unequal time intervals is a very special problem, as measurements are not necessarily taken at regular time points.

Taking into account the length of the interval is accomplished by means of analyzing the area between two expression profiles, joined by the corresponding measurements at subsequent time points. This is equivalent to considering the sum or average of squared errors between the infinite points in the two lines. This analysis can be easily achieved by computing the underlying integral, which is analytically resolved in advance, subsequently avoiding expensive computations during the clustering process.

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 $[0, T]$. Functions $x(t)$ and $\hat{y}(t)$ may cross each other many times, but 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 (see Fig. 1). Let a denote the amount of vertical shifting of $y(t)$. Then, we want to find 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 results of [13] generalize from the case of piece-wise linear profiles to profiles which are *any* integrable functions on a finite interval. Suppose we have two profiles, $x(t)$ and $y(t)$, defined on the time-interval $[0, T]$. The alignment process consists of finding the value a that minimizes

$$f_a(x(t), y(t)) = \int_0^T \Big[x(t) - \hat{y}(t)\Big]^2 dt = \int_0^T \Big[x(t) - [y(t) - a]\Big]^2 dt. \tag{1}$$

Differentiating yields

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

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

$$a_{\min} = -\frac{1}{T} \int_0^T \Big[x(t) - y(t)\Big] dt, \tag{3}$$

and since $\frac{d^2}{da^2} f_a(x(t), y(t)) = 2T > 0$ then $a_{\min}$ is a minimum. The integrated error between $x(t)$ and the shifted $\hat{y}(t) = y(t) - a_{\min}$ is then

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

In terms of Fig. 1, this means that the sum of all the areas where $x(t)$ is above $y(t)$ minus the sum of those areas where $y(t)$ is above $x(t)$, is zero.

Given an original profile $x(t) = [e_1, e_2, \ldots, e_n]$ (with n expression values taken at n time-points $t_1, t_2, \ldots, t_n$) 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

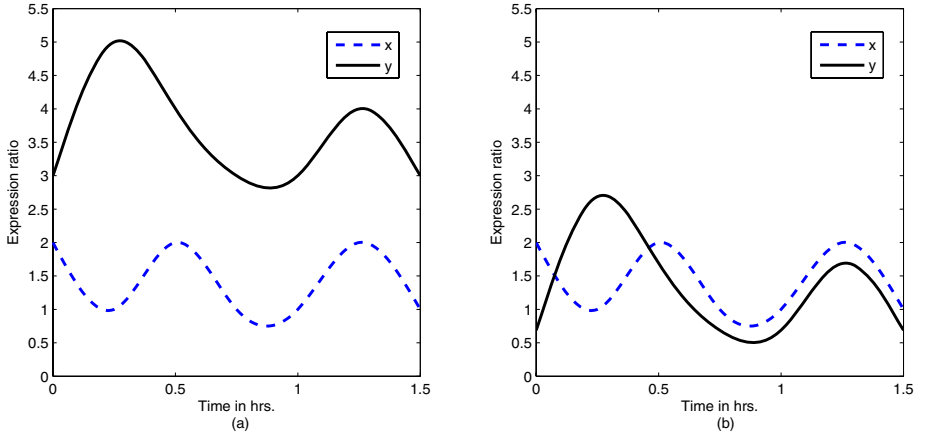

Fig. 1. (a) Unaligned profiles $x(t)$ and $y(t)$. (b) Aligned profiles $x(t)$ and $y(t)$, after applying $y(t) \leftarrow y(t) - a_{\min}$.

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

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 need only 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 Eq. (3) as

$$a_{\min} = -\frac{1}{T}\sum_{j=1}^{n-1}\int_{t_j}^{t_{j+1}} \Big[x_j(t) - y_j(t)\Big]\,dt = -\frac{1}{T}\sum_{j=1}^{n-1}\sum_{k=0}^{3}\frac{(x_{jk} - y_{jk})\,(t_{j+1} - t_j)^{k+1}}{k+1}. \tag{6}$$

Fig. 1(b) shows a pairwise alignment, of the two initial profiles in Fig. 1(a), after applying the vertical shift $y(t) \leftarrow y(t) - a_{\min}$. The two aligned profiles cross each other many times, but the integrated error, Eq. (4), is zero.

In particular, from Eq. (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 the next section, we use this property of Eq. (4) to define the multiple alignment of a set of profiles.

3 Multiple Expression Profile Alignment

Given a set $X = \{x_1(t), \ldots, x_s(t)\}$, we want to align the profiles such 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

$$f_{a_i,a_j}\left(x_i(t), x_j(t)\right) = \int_0^T \left[\hat{x}_i(t) - \hat{x}_j(t)\right]^2 dt = \int_0^T \left[\left[x_i(t) - a_i\right] - \left[x_j(t) - a_j\right]\right]^2 dt, \tag{7}$$

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 Eq. (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

$$F_{a_1,\ldots,a_s}\left(x_1(t), \ldots, x_s(t)\right) = \sum_{1 \le i < j \le s} f_{a_i,a_j}\left(x_i(t), x_j(t)\right), \tag{8}$$

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 $\int_0^T \left[\hat{x}_i(t) - \hat{x}_j(t)\right] dt = 0$.*

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

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. From Lemma 1,
$\int_0^T \left[\hat{x}_i(t) - \hat{x}_j(t)\right] dt = 0 \Rightarrow \int_0^T \left[\left[x_i(t) - a_{\min_i}\right] - \left[x_j(t) - a_{\min_j}\right]\right]^2 dt$ 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 Eq. (8), 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 Eq. (8).

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*

$$\hat{x}_i(t) = x_i(t) - a_{\min_i}, \text{ where, } a_{\min_i} = -\frac{1}{T}\int_0^T \Big[z(t) - x_i(t)\Big]dt, \tag{9}$$

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

$$\hat{x}_i(t) = x_i(t) - a_{\min_i}, \text{ where, } a_{\min_i} = \frac{1}{T}\int_0^T x_i(t)dt. \tag{10}$$

We use the multiple alignment of Eq. (10) in all subsequent discussions. Using spline interpolations, each profile $x_i(t)$, $1 \le i \le s$, is a continuous integrable profile

$$x_i(t) = \begin{cases} x_{i,1}(t) & \text{if} \quad t_1 \le t \le t_2 \\ & \vdots \\ x_{i,n-1}(t) & \text{if} \quad t_{n-1} \le t \le t_n \end{cases} \tag{11}$$

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 Eq. (10) is

$$a_{\min_i} = \frac{1}{T}\sum_{j=1}^{n-1}\sum_{k=0}^{3}\frac{x_{ijk}\,(t_{j+1}-t_j)^{k+1}}{k+1} \tag{12}$$

4 Distance Function

The distance between any two piecewise linear profiles was defined as $f(a_{\min})$ in [13]. For convenience here, we change the definition slightly to

$$d(x,y) = \frac{1}{T}f(a_{\min}) = \frac{1}{T}\int_0^T \Big[x(t) + a_{\min} - y(t)\Big]^2 dt. \tag{13}$$

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

$$\langle\phi\rangle \triangleq \frac{1}{T}\int_0^T \phi(t)dt. \tag{14}$$

Then, from Eqs. (1) and (3),

$$\begin{aligned} d(x,y) &= \frac{1}{T}\int_0^T \Big[[x(t)-y(t)]^2 + 2a_{\min}\,[x(t)-y(t)] + a_{\min}^2\Big]dt \\ &= \frac{1}{T}\int_0^T \Big[x(t)-y(t)\Big]^2 dt - 2a_{\min}^2 + a_{\min}^2 \\ &= \Big\langle [x(t)-y(t)]^2 \Big\rangle - \langle x(t)-y(t)\rangle^2. \end{aligned} \tag{15}$$

Apart from the factor $\frac{1}{T}$, this is precisely the distance $d_{\mathrm{PA}}(x, y, t)$ in [13]. By performing the multiple alignment of Eq. (10) to obtain new profiles $\hat{x}(t)$ and $\hat{y}(t)$, we have:

$$d(x, y) = \left\langle [\hat{x}(t) - \hat{y}(t)]^2 \right\rangle = \frac{1}{T} \int_0^T \left[\hat{x}(t) - \hat{y}(t)\right]^2 dt. \tag{16}$$

Thus, $d(x, y)^{\frac{1}{2}}$ is the 2-norm, satisfying all the properties we might want for a metric. On the other hand, it is easy to show that $d(x, y)$ in Eq. (16) does not satisfy the triangle inequality, and hence it is not a metric. We, however, use $d(x, y)$ in Eq. (16) as our distance function, since it is algebraically easier to work with than the metric $d(x, y)^{\frac{1}{2}}$. Eq. (16) is closer to the spirit of regression analysis, and thus, we can dispense with the requirement for the triangle inequality. Also the distance as defined in Eq. (16) is unchanged by an additive shift, and hence, is order-preserving; that is: $d(u, v) \leq d(x, y)$ if and only if $d(\hat{u}, \hat{v}) \leq d(\hat{x}, \hat{y})$. This property has important implications for distance-based clustering methods that rely on pairwise alignments of profiles; as discussed later in the experiment section.

With the spline interpolations of Eq. (5), we derived the analytical solution for $d(x, y)$ in Eq. (16), using the symbolic computational package, *Maple*[1], as follows:

$$\begin{aligned} d(x,y) = {} & \frac{P^2(n^7 - m^7)}{7} + \frac{(2PQ - 6P^2m)(n^6 - m^6)}{6} + \frac{(2PR - 10PQm + Q^2 + 15P^2m^2)(n^5 - m^5)}{5} + \\ & \frac{(-8PRm - 4Q^2m + 2PS + 20PQm^2 + 2QR - 20P^2m^3)(n^4 - m^4)}{4} + \\ & \frac{(-6QRm - 20Pm^3Q + R^2 + 6Q^2m^2 + 12Pm^2R - 6PmS + 15P^2m^4 + 2QS)(n^3 - m^3)}{3} + \\ & \frac{(10Pm^4Q + 6Qm^2R + 2RS - 8Pm^3R - 2R^2m - 6P^2m^5 + 6Pm^2S - 4QmS - 4Q^2m^3)(n^2 - m^2)}{2} \\ & -2RmS(n-m) + S^2(n-m) + P^2m^6(n-m) + Q^2m^4(n-m) + R^2m^2(n-m) - \\ & 2Qm^3R(n-m) - 2Pm^5Q(n-m) - 2Pm^3S(n-m) + 2Pm^4R(n-m) + 2Qm^2S(n-m) \end{aligned} \tag{17}$$

where, $P = (x_{j3} - y_{j3})$, $Q = (x_{j2} - y_{j2})$, $R = (x_{j1} - y_{j1})$, $S = (x_{j0} - y_{j0} + c_y - c_x)$, $m = t_j$ and $n = t_{j+1}$.

5 Centroid of a Set

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

$$\Delta[\mu] = \sum_{i=1}^{s} d\,(x_i, \mu)\,. \tag{18}$$

where, Δ plays the role of the *within-cluster-scatter* defined in [13]. Since $d(\cdot, \cdot)$ is unchanged by an additive shift $x(t) \to x(t) - a$ in either of its arguments, we have

[1] All the analytical solutions in this paper were derived by Maple.

$$\Delta[\mu] = \sum_{i=1}^{s} d\left(\hat{x}_i, \mu\right) = \frac{1}{T} \int_0^T \sum_{i=1}^{s} \left[\hat{x}_i(t) - \mu(t)\right]^2 dt, \tag{19}$$

where, $\hat{X} = \{\hat{x}_1(t), \ldots, \hat{x}_s(t)\}$ is the multiple alignment of Eq. (10). This is a *functional* of μ; that is, a mapping from the set of real valued functions defined on $[0, T]$ to the set of real numbers. To minimize with respect to μ we set the functional derivative to zero[2]. This functional is of the form

$$F[\phi] = \int L(\phi(t))dt, \tag{20}$$

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

$$\frac{\delta \Delta[\mu]}{\delta \mu(t)} = -\frac{2}{T} \sum_{i=1}^{s} \left[\hat{x}_i(t) - \mu(t)\right] = -\frac{2}{T} \left(\sum_{i=1}^{s} \hat{x}_i(t) - s\mu(t) \right). \tag{21}$$

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

$$\mu(t) = \frac{1}{s} \sum_{i=1}^{s} \hat{x}_i(t). \tag{22}$$

With the spline coefficients, x_{ijk}, of each $x_i(t)$ interpolated as in Eq. (11), the analytical solution for $\mu(t)$ in Eq. (22) is

$$\mu_j(t) = \frac{1}{s} \sum_{i=1}^{s} \left[\sum_{k=0}^{3} x_{ijk} \left(t - t_j\right)^k \right] - a_{\min_i}, \quad \text{in each interval } [t_j, t_{j+1}]. \tag{23}$$

Eq. (22) applies to aligned profiles while Eq. (23) can apply to unaligned profiles.

6 *k*-Means Clustering via Multiple Alignment

Many clustering methods have been developed, and each has its own advantages and disadvantages regarding handling noise in the measurements and the properties of the data set being clustered. None of them is considered the best method. In [13], hierarchical clustering was used and the decision rule was the *farthest neighbor* distance between two clusters; computed using an equivalent of Eq. (1) for piece-wise linear profiles. Hierarchial clustering is a greedy method that cannot be readily applied on large data sets. Our approach allows us to apply flat clustering such as k-means, which, though not optimal, provides a fast and practical solution to the problem. This also applies to fuzzy k-means or expectation maximization (EM) clustering methods.

[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.

In k-means [14], we want to partition a set of s profiles, $\mathcal{D} = \{x_1(t), \ldots, x_s(t)\}$, into k disjoint clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$, $1 \leq k \leq s$; such that (i) $\phi, i = 1, \ldots, k$ (ii) $U_{i=1}^{k} \mathcal{C}_i = \mathcal{D}$ (iii) $\mathcal{C}_i \cap \mathcal{C}_j = \phi$; $i \neq j$; $i, j = 1, \ldots, k$. Also, each profile is assigned to the cluster whose mean is the closest. It is similar to EM for mixtures of Gaussians in the sense that they both attempt to find the centers of natural clusters in the data. It assumes that the object features form a *vector space*. Let $U = \{u_{ij}\}$ be the membership matrix:

$$u_{ij} = \begin{cases} 1 & \text{if } d(x_i, \mu_j) = min_{l=1,\ldots,k} d(x_i, \mu_l) \, where \;\; i = 1, \ldots, s \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

The aim is to minimize the sum of squared distances:

$$J(\theta, U) = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij} d(x_i, \mu_j) \,. \tag{25}$$

where $\theta = \mu_1, \mu_2, \ldots, \mu_n$.

Algorithm 1. *k-MCMA: k-Means Clustering with Multiple Alignment*

Input: Set of profiles, $\mathcal{D} = \{x_1(t), \ldots, x_s(t)\}$, and desired number of clusters, k
Output: 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 $1 \leq i \leq k$ (see Section 2)
2. Multiple-align transformed $\mathcal{D}$ to obtain $\hat{\mathcal{D}} = \{\hat{x}_1(t), \ldots, \hat{x}_s(t)\}$, using Eq. (10)
3. Randomly initialize centroid $\hat{\mu}_i(t)$, for $1 \leq i \leq k$

repeat

4.a. Assign $\hat{x}_j(t)$ to cluster $\hat{\mathcal{C}}_{\hat{\mu}_i}$ with minimal $d(\hat{x}_j, \hat{\mu}_i)$, for $1 \leq j \leq s$ and $1 \leq i \leq k$

4.b. Update $\hat{\mu}_i(t)$ of $\hat{\mathcal{C}}_{\hat{\mu}_i}$, for $1 \leq i \leq k$

until Convergence: that is, no change in $\hat{\mu}_i(t)$, for $1 \leq i \leq k$
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 Eq. (10), and then cluster the multiple aligned $\hat{\mathcal{D}}$ with k-means. Recall that the process of Eq. (10) 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.a), 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 Eq. (16) can be used in step (4.a), including Euclidean distance. Also, by Theorem 2 below, there is no need to multiple-align $\hat{\mathcal{C}}_{\hat{\mu}_i}$ in step (4.b), 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, $a_{\min_{\bar{\mu}}} = \frac{1}{T}\int_0^T \bar{\mu}(t)dt = \frac{1}{T}\int_0^T \frac{1}{m}\sum_{i=1}^{m} \hat{x}_i(t) = 0$, 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 $\mathcal{D}$, and even more than this when Euclidean distance is used to assign a profile to a cluster. An important implication of Eq. (16) 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, as pointed out in Section 4. In k-means on $\mathcal{D}$, step (4.a) 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 [15]; which is, a dissimilarity function that is not metric can be made metric by using a shift operation (in our case any metric can be used in step (4.a) such as Euclidean distance). In this case the objective function of k-means does not change, and convergence is assured. Thus, this saves a lot of computations and opens the door for applications of multiple alignment methods to many distance-based clustering methods. This is a future research direction that we plan to investigate.

7 Computational Experiments

The performance of the k-MCMA method on a set of pre-clustered budding yeast, *Saccharomyces cerevisiae*, data set of [1][3] is discussed in this section. The data set contains time-series gene expression profiles of the complete characterization of mRNA transcript levels during the yeast cell cycle. These experiments measured the expression levels of the 6,220 yeast genes during the cell cycle at seventeen different points, from 0 to 160 minutes, at every 10-minute time-interval. From those gene profiles, 221 profiles were analyzed. We normalized each expression profile as in [1]; that is, we divided each transcript level by the mean value of each profile with respect to each other.

The data set 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 Fig. 2(b), and Table 1 shows the complete data set. Setting $k = 5$, we applied k-MCMA on the data set to see if k-MCMA is able to find these phases as accurately as possible. Once the clusters have been found, to compare the k-MCMA clustering with the pre-clustered dataset of [1], the next step is to label the clusters, where the labels are the "phases" in the pre-clustered dataset.

[3] `http://genomics.stanford.edu/yeast_cell_cycle/cellcycle.html`

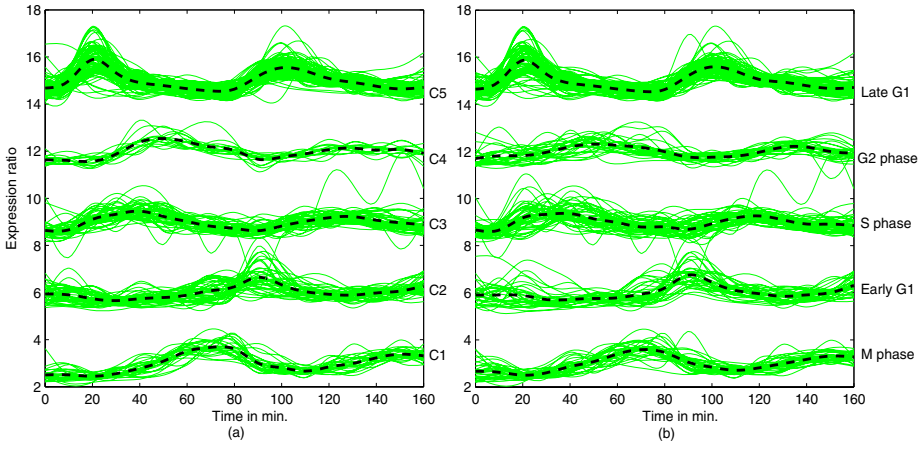

Fig. 2. (a) k-MCMA clusters and (b) Yeast phases [1], with centroids shown

Although this can be done in many different ways, we adopted the following approach. We assigned each k-MCMA cluster to a yeast phase using the *Hungarian algorithm* [16]. The Hungarian method is a combinatorial optimization algorithm which solves the assignment problem in polynomial time. Our phase assignment problem is formulated using a complete bipartite graph $G = (C, P, E)$ with k cluster vertices (C) and k phases vertices (P), and each edge in E has a nonnegative cost $c(\hat{\mathcal{C}}_{\hat{\mu}_i}, \hat{\mathcal{P}}_{\hat{\nu}_j})$, $\hat{\mathcal{C}}_{\hat{\mu}_i} \in C$ and $\hat{\mathcal{P}}_{\hat{\nu}_j} \in P$. We want to find a perfect matching with minimum cost. The cost of an edge between a cluster vertex $\hat{\mathcal{C}}_{\hat{\mu}_i}$ and a phase vertex $\hat{\mathcal{P}}_{\hat{\nu}_j}$ is the distance between their centroids $\hat{\mu}_i, \hat{\nu}_j$; that is $c(\hat{\mathcal{C}}_{\hat{\mu}_i}, \hat{\mathcal{P}}_{\hat{\nu}_j}) = d(\hat{\mathcal{C}}_{\hat{\mu}_i}, \hat{\mathcal{P}}_{\hat{\nu}_j})$, and the distances are computed using Eq. (16). In terms of such bipartite graph, the Hungarian method will select the k perfect matching pairs $(\hat{\mathcal{C}}_{\hat{\mu}_i}, \hat{\mathcal{P}}_{\hat{\nu}_j})$ with minimum cost. In Fig. 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 *C5* of k-MCMA is assigned to the *Late G1* phase of [1] by our phase assignment approach, and hence they are at the same level in the figure.

The five clusters found by k-MCMA are shown in Fig. 2(a), while the corresponding phases of [1] after the phase assignment are shown in Fig. 2(b). 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 k-MCMA (Fig. 2(a)) and the *known phase centroids* of the yeast data (Fig. 2(b)). In the figure, each cluster and phase were multiple-aligned using Eq. (10) to enhance visualization.

Fig. 2 clearly shows a high degree of similarity between the k-MCMA clusters and the yeast phases. Visually, each k-MCMA cluster on the left is *very similar* to exactly one of the yeast phases, which we show at the same level on the right. Visually also, it even *"seems"* that k-MCMA clusters are more accurate than

Table 1. Pre-clustered yeast genes from Table 1 of [1] with their actual phases and k-MCMA cluster numbers

Gene Names	Phases	k-KCMA	Gene Names	Phases	k-KCMA	Gene Names	Phases	k-KCMA
YBR202w/CDC47	Early G1	1	YER069w/ARG5,6	G2	3	YMR076C/PDS5	Late G1	5
YPL058C/PDR12	Early G1	1	YJR112W/NNF1	G2	3	YMR078C/CHL12	Late G1	5
YGL255W/ZRT1	G2	1	YLL046c/RNP1	G2	3	YNL225C/CNM1	Late G1	5
YIL106W/MOB1	G2	1	YJL092W/HPR5	G2	3	YPL209C/IPL1	Late G1	5
YBR038w/CHS2	G2	1	YCR084c/TUP1	G2	3	YPL241C/CIN2	Late G1	5
YDL048c/STP4	M	1	YKL032C/IXR1	G2 & M	3	YDR507c/GIN4	Late G1	5
YGR143W/SKN1	M	1	YLL021w/SPA2	Late G1	3	YGL027C/CWH41	Late G1	5
YGL116W/CDC20	M	1	YDR297w/SUR2	Late G1	3	YGR041W/BUD9	Late G1	5
YGR108W/CLB1	M	1	YPL127C/HHO1	Late G1	3	YGR152C/RSR1	Late G1	5
YPR119W/CLB2	M	1	YDR224c/HTB1	Late G1 & G2	3	YIL159W/BNR1	Late G1	5
YBR138c/HDR1	M	1	YDR225w/HTA1	Late G1 & G2	3	YLR286C/CTS1	Late G1	5
YGR092W/DBF2	M	1	YPR167C/MET16	M	3	YLR313C/SPH1	Late G1	5
YHR023w/MYO1	M	1	YER001w/MNN1	S	3	YNL233W/BNI4	Late G1	5
YOL069W/NUF2	M	1	YER003c/PMI40	S	3	YAR007C/RFA1	Late G1	5
YOR058C/ASE1	M	1	YIR017C/MET28	S	3	YBL035c/POL12	Late G1	5
YPL242C/IQG1	M	1	YKR001C/SPO15	S	3	YBR088c/POL30	Late G1	5
YJR092W/BUD4	M	1	YDL155W/CLB3	S	3	YBR252w/DUT1	Late G1	5
YLR353W/BUD8	M	1	YBL063w/KIP1	S	3	YBR278w/DPB3	Late G1	5
YMR001C/CDC5	M	1	YDR113c/PDS1	S	3	YDL164C/CDC9	Late G1	5
YNL053W/MSG5	M	1	YDR356w/NUF1	S	3	YER070w/RNR1	Late G1	5
YGL021W/ALK1	M	1	YEL061c/CIN8	S	3	YJL173C/RFA3	Late G1	5
YDR146c/SWI5	M	1	YGR140W/CBF2	S	3	YKL045W/PRI2	Late G1	5
YLR131c/ACE2	M	1	YHR172W/SPC97	S	3	YLR103c/CDC45	Late G1	5
YOR025W/HST3	M	1	YLR045c/STU2	S	3	YML102W/CAC2	Late G1	5
YOR229W/WTM2	M	1	YNL126W/SPC98	S	3	YNL102W/CDC17	Late G1	5
YLL040c/VPS13	Early G1	2	YPR141C/KAR3	S	3	YNL262W/POL2	Late G1	5
YDL179w/	Early G1	2	YBL002w/HTB2	S	3	YNL312W/RFA2	Late G1	5
YLR079w/SIC1	Early G1	2	YBL003c/HTA2	S	3	YOR074C/CDC21	Late G1	5
YBR200w/BEM1	Early G1	2	YJR006W/HUS2	S	3	YPR018W/RLF2	Late G1	5
YBL023c/MCM2	Early G1	2	YCR035c/RRP4	S	3	YPR175W/DPB2	Late G1	5
YEL032w/MCM3	Early G1	2	YER016w/BIM1	S	3	YLR382C/NAM2	Late G1	5
YJL194W/CDC6	Early G1	2	YER118c/SSU81	S	3	YDL227C/HO	Late G1	5
YLR274W/CDC46	Early G1	2	YAL001C/TFC3	S	3	YGL089C/MF(α_2)	Late G1	5
YPR019W/CDC54	Early G1	2	YFR037C/RSC8	S	3	YNL173C/MDG1	Late G1	5
YCL040w/GLK1	Early G1	2	YPL016W/SWI1	S	3	YBR070c/SAT2	Late G1	5
YCR005c/CIT2	Early G1	2	YJR159W/SOR1	S & G2	3	YBR073w/RDH54	Late G1	5
YDL181W/INH1	Early G1	2	YDR277c/MTH1	S & M	3	YGL200C/EMP24	Late G1	5
YGR183C/QCR9	Early G1	2	YJL137c/GLG2	G2	4	YHR153c/SPO16	Late G1	5
YLR273C/PIG1	Early G1	2	YIL050W/PCL7	G2	4	YKL101W/HSL1	Late G1	5
YLR395C/COX8	Early G1	2	YBL097w/BRN1	G2	4	YKL165C/MCD4	Late G1	5
YML110C/COQ5	Early G1	2	YCL014w/BUD3	G2	4	YLL002w/KIM2	Late G1	5
YMR256c/COX7	Early G1	2	YJL099W/CHS6	G2	4	YLR233C/EST1	Late G1	5
YHR005c/GPA1	Early G1	2	YCR073c/SSK22	G2	4	YLR457C/NBP1	Late G1	5
YJL157C/FAR1	Early G1	2	YDR389w/SAC7	G2	4	YNL272C/SEC2	Late G1	5
YKL185W/ASH1	Early G1	2	YLR210W/CLB4	S	4	YPL057C/SUR1	Late G1	5
YGR281W/YOR1	Early G1	2	YMR198W/CIK1	S	4	YPL124W/NIP29	Late G1	5
YBR083w/TEC1	Early G1	2	YJR137C/ECM17	S	4	YDL101C/DUN1	Late G1	5
YBR104w/YMC2	G2	2	YMR190C/SGS1	S	4	YDR097C/MSH6	Late G1	5
YLR014c/PPR1	G2	2	YBL052c/SAS3	S	4	YKL113C/RAD27	Late G1	5
YOR274W/MOD5	G2	2	YIL126W/STH1	S	4	YLR032w/RAD5	Late G1	5
YKL068W/NUP100	G2	2	YHR086w/NAM8	S & G2	4	YLR234W/TOP3	Late G1	5
YOR317W/FAA1	Late G1	2	YDR150w/NUM1	S & M	4	YLR383W/RHC18	Late G1	5
YPL187W/MF(α_1)	Late G1	2	YIL009W/FAA3	Early G1	5	YML021C/UNG1	Late G1	5
YOR316C/COT1	Late G1	2	YNR016C/ACC1	Early G1	5	YML060W/OGG1	Late G1	5
YHR038W/KIM4	Late G1	2	YER111c/SWI4	Early G1	5	YML061C/PIF1	Late G1	5
YAL040C/CLN3	M	2	YLR258W/GSY2	Early G1	5	YNL082W/PMS1	Late G1	5
YCL037c/SRO9	M	2	YBR067c/TIP1	Early G1	5	YOL090W/MSH2	Late G1	5
YDL138W/RGT2	M	2	YGL055W/OLE1	Early G1	5	YPL153C/SPK1	Late G1	5
YIL162W/SUC2	M	2	YKL092C/BUD2	Early G1 & G2	5	YML027W/YOX1	Late G1	5
YKL130C/SHE2	M	2	YOR373W/NUD1	Early G1 & S	5	YMR179W/SPT21	Late G1	5
YHR152w/SPO12	M	2	YJL196C/ELO1	Late G1	5	YBR160w/CDC28	Late G1 & M	5
YIL167W/	M	2	YJR148W/TWT2	Late G1	5	YAR008W/SEN34	S	5
YKL129C/MYO3	M	2	YDL127w/PCL2	Late G1	5	YDL093W/PMT5	S	5
YCR042c/TSM1	M	2	YGR109C/CLB6	Late G1	5	YDL095W/PMT1	S	5
YNL073W/MSK1	S	2	YJL187C/SWE1	Late G1	5	YDR488c/PAC11	S	5
YER017c/AFG3	S	2	YMR199W/CLN1	Late G1	5	YOR026W/BUB3	S	5
YML091C/RPM2	S & G2	2	YNL289W/PCL1	Late G1	5	YIL140W/SRO4	S	5
YKL049C/CSE4	G2	3	YPL256C/CLN2	Late G1	5	YKL067W/YNK1	S	5
YPR111W/DBF20	G2	3	YPR120C/CLB5	Late G1	5	YKL127W/PGM1	S	5
YJR076C/CDC11	G2	3	YDL003W/RHC21	Late G1	5	YBR275c/RIF1	S	5
YKL048C/ELM1	G2	3	YFL008W/SMC1	Late G1	5	YCR065w/HCM1	S	5
YOR188W/MSB1	G2	3	YJL074C/SMC3	Late G1	5	YDL197C/ASF2	S	5
YDL198C/YHM1	G2	3	YKL042W/SPC42	Late G1	5	YJL115W/ASF1	S	5
YDR464w/SPP41	G2	3	YLR212C/TUB4	Late G1	5			

the yeast phases; which suggest that k-MCMA can also correct manual phase assignment errors, if any.

To show the biological significance of the results, the 221 yeast genes are listed in Table 1, where, for each gene, the cluster number that k-MCMA assigns to a gene and the actual yeast phase in [1] of that same gene is shown.

An objective measure for comparing k-MCMA clusters with the yeast phases was computed as follows. For each k-MCMA cluster $\hat{\mathcal{C}}_{\hat{\mu}_c}$ $(1 \leq c \leq k = 5)$, we find the shortest distance between each profile $x_i(t)$, $1 \leq i \leq |\hat{\mathcal{C}}_{\hat{\mu}_c}|$, and all five phase centroids $\nu_j(t)$, $1 \leq j \leq k = 5$, using Eq. (16). Profile $x_i(t)$ will be assigned the *correct* label (i.e., assigned to phase label of $\hat{\mathcal{P}}_{\hat{\nu}_j}$) whenever $x_i(t) \in \hat{\mathcal{P}}_{\hat{\nu}_j}$ and $\left(\hat{\mathcal{C}}_{\hat{\mu}_c}, \hat{\mathcal{P}}_{\hat{\nu}_j}\right) \in \mathcal{S}$ 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 $\hat{\mathcal{P}}_{\hat{\nu}_j}$ by our pair-assignment method. The percentage of *correct* assignments over the 221 profiles was used as our measure of accuracy, resulting in 79.64%. That is

$$\text{Accuracy} = \frac{\sum_{c=1}^{k} \sum_{i=1}^{|\hat{\mathcal{C}}_{\hat{\mu}_c}|} E\left(\ c\ ,\ \arg\min_{1 \leq j \leq k} d\left(x_i, \nu_j\right)\ \right)}{221}, \tag{26}$$

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 [1], the 5 phases were determined using biological information, including genomic and phenotypic features observed in the yeast cell cycle experiments. This 79.64% accuracy is quite high considering that k-MCMA is an *unsupervised* learning method.

8 Conclusion

We proposed k-MCMA, a method that combines k-means with multiple alignment of gene expression profiles to cluster microarray time-series data. The profiles are represented as natural cubic splines functions, to compare profiles, where expression measurements were not taken at the same time-intervals. Multiple alignment is based on minimizing the sum of integrated squared errors over a time-interval, defined on a set of profiles. k-MCMA was able to find the 5 yeast cell-cycle phases of [1] with an accuracy of about 80%. k-MCMA can also be used to correct manual phase assignment errors. In the future, we plan to study other distance-based clustering approaches using our multiple alignment method. It will be also interesting to study the effectiveness of any such clustering methods in dose-response microarray data sets. Cluster validity indices based on multiple alignment will also be investigated. We argue that in real applications data can be very noisy, and the use of cubic spline interpolation could lead to some problems. The use of splines has the advantage of being tractable, however, although we also plan to study interpolation methods that incorporate noise. Currently, we are also carrying out experiments with larger data sets.

Acknowledgements. This research has been partially funded by Canadian NSERC Grant #RGPIN228117-2006 and CFI grant #9263.

References

1. Cho, R., Campbell, M., Winzeler, E., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T., Gareilian, A., Lockhart, D., Davis, R.: A genome-wide transactional analysis of the mitotic cell cycle. Molecular Cell 2(1), 65–73 (1998)
2. Bar-Joseph, Z., Gerber, G., Jaakkola, T., Gifford, D., Simon, I.: Continuous representations of time series gene expresion data. Journal of Comp. Biology 10(3-4) (2003)
3. Bréhélin, L.: Clustering gene expression series with prior knowledge. In: Casadio, R., Myers, G. (eds.) WABI 2005. LNCS (LNBI), vol. 3692, pp. 27–38. Springer, Heidelberg (2005)
4. Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P., Herskowitz, I.: The transcriptional program of sporulation in budding yeast. Science 282, 699–705 (1998)
5. Djean, S., Martin, P., Baccini, A., Besse, P.: Clustering time-series gene expression data using smoothing spline derivatives. EURASIP Journal on Bioinformatics and Systems Biology 70561, 705–761 (2007)
6. Ernst, J., Nau, G., Bar-Joseph, Z.: Clustering short time series gene expression data. Bioinformatics 21(suppl. 1), i159–i168 (2005)
7. Ramoni, M., Sebastiani, P., Kohane, I. (eds.): Cluster analysis of gene expression dynamics. Proc. Natl. Acad. Sci. USA 99 (2002)
8. Tavazoie, S., Hughes, J., Campbell, M., Cho, R., Church, G.: Systematic determination of genetic network architecture. Nature Genetics 22, 281–285 (1999)
9. Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E., Golub, T. (eds.): Interpreting patterns of gene expression with SOMs: Methods and application to hematopoietic differentiation, vol. 96 (1999)
10. Heyer, L., Kruglyak, S., Yooseph, S.: Exploring expression data: identification and analysis of coexpressed genes. Genome Research 9, 1106–1115 (1999)
11. Moller-Levet, C., Klawonn, F., Cho, K., Wolkenhauer, O.: Clustering of unevenly sampled gene expression time-series data. Fuzzy sets and Systems 152(1-16), 49–66 (2005)
12. Peddada, S., Lobenhofer, E., Li, L., Afshari, C., Weinberg, C., Umbach, D.: Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference. Bioinformatics 19(7), 834–841 (2003)
13. Rueda, L., Bari, A., Ngom, A.: Clustering time-series gene expression data with unequal time intervals. In: Priami, C., Dressler, F., Akan, O.B., Ngom, A. (eds.) Transactions on Computational Systems Biology X. LNCS (LNBI), vol. 5410, pp. 100–123. Springer, Heidelberg (2008)
14. Xu, R., Wunsch, D.: Clustering. Wiley-IEEE Press, Chichester (2008)
15. Roth, V., Laub, J., Kawanabe, M., Buhmann, J.: Optimal cluster preserving embedding of nonmetric proximity data. IEEE Trans. on Pattern Analysis and Machine Intelligence 25(12), 1540–1551 (2003)
16. Kuhn, H.: The hungarian method for the assignment problem. Naval Research Logistics 52(1), 7–21 (2005)