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

Ataul Bari[1], Luis Rueda[1] and Alioune Ngom[1]

School of Computer Science, University of Windsor, 401 Sunset Avenue, Windsor, Ontario, N9B 3P4, Canada, {bari1, lrueda, angom}@uwindsor.ca

Abstract. Pairwise alignment approaches for time-varying gene expression profiles have been recently developed for the detection of co-expressions in time-series microarray data sets. In this paper, we analyze *multiple expression profile alignment* (MEPA) methods for classifying microarray time-course data. We apply a nearest centroid classification technique, in which the centroid of each class is computed by means of a MEPA algorithm. MEPA aligns the expression profiles in such a way to minimize the total area between all aligned profiles. We propose four MEPA approaches whose effectiveness are demonstrated on the well-known budding yeast, *S. cerevisiae*, data set.

Key words: Pattern analysis, time-series analysis, gene expression profiles, classification

1 Introduction

Finding patterns on genes based on the similarity of their temporal profile expressions is important for many studies, such as those genes that are functionally related or co-regulated [7]. Analyzing gene expression data given in terms of time series is different from a general pattern analysis problem, because exchanging two time points delivers quite different results, while it may not be biologically meaningful.

Many unsupervised methods for gene pattern analysis based on the similarity (or dissimilarity) of their microarray temporal profiles have been proposed in the past few years [1–3, 7, 9, 12–17]. One of the methods for clustering microarray time-series data is based on a hidden phase model (similar to a hidden Markov model) to define the parameters of a mixture of normal distributions in a Bayesian-like manner, which are estimated by using expectation maximization (EM) [3]. Pattern analysis of time-series data has also been studied using a Bayesian approach in [13], and a hidden Markov model (HMM) in [14]. A partitional clustering based on k-means and Euclidian distance has been studied in [17], and in [16], self-organizing maps (SOMs) have been applied to visualize and interpret gene temporal expression profiles patterns. Other methods based on correlation measures have been proposed for finding patterns of genes using microarray time-series data [4, 9]. The method proposed in [4] requires computing the mean expression levels of some candidate profiles using some pre-identified, arbitrarily selected profiles. In [9], a method for clustering microarray time-series data employing a jack-knife correlation coefficient with or without using the seeded candidate profiles is proposed. 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 selected genes. This makes it vulnerable to the possibility of missing important genes, since the resulting clusters depend upon the initially chosen template genes. Another method is to select and classify genes using

M. Chetty, S. Ahmad, A. Ngom, S. W. Teng (Eds.): PRIB2008,
Supplementary Conf. Proc., pp.25-36, 2008.

the ideas of order-restricted inference, where estimation makes use of known inequalities among parameters [12]. In this method, at first, potential candidate profiles of interest are defined and expressed in terms of inequalities between the expected gene expression levels at various time points. For a given candidate profile, the estimated mean expression level of each gene is computed and the best fitting profile for a given gene is selected using the goodness-of-fit criterion and the bootstrap test procedure. In this approach, two genes expression profiles $\mathbf{x}_1$ and $\mathbf{x}_2$ fall into the same cluster if they show similar profiles in terms of direction of the changes of expression ratios (e.g. up-up-up-down-down), regardless how big/small the change is. In [1], a minimum-square-error profile alignment approach to find patterns in microarray time-series data was proposed. The idea is to pairwisely align two temporal profiles in such a way that the sum of square errors between two aligned vectors is minimized. The alignment procedure, however, does not consider the length of the interval between the time points at which individual measurements are taken. In [18], a pairwise gene expression profile alignment approach to find patterns in temporal microarray data that minimizes the area between two aligned profiles has been proposed. The profile alignment proposed in [18] is different from that of [1] in the sense that: (i) the approach proposed in [18] considers unequal time intervals, which is usually the case in microarray time-series experiments, and (ii) the alignment is performed by minimizing the error between two continuous functions and not the "knot" points. Recently, the authors of [20] proposed a variation-based co-expression detection algorithm, which finds clusters of co-expressed genes. Their method is based on calculating the degree of change of the expressions between adjacent time points and evaluating their trend similarities over time.

The mean of a cluster or class in time-series profiles is called the *centroid*, which represents the common features or patterns of the elements in the class. The centroid is also used by validity indices [6, 11], to evaluate the quality of the result obtained by a clustering algorithm. It is still an open problem to find the exact number of clusters, and therefore, determining the *correct* centroids is also a difficult problem. In this paper, we propose to compute the centroids of microarray time-series data by means of *multiple expression profile alignment* (MEPA) methods. Specifically, we focus on the problem of classifying the time-series data rather than clustering them. Our motivation is only to test and evaluate the effectiveness of our MEPA methods (we are currently working on characterizing the clustering problem by means of MEPA methods), and hence, our contribution in this paper is a proof of concepts. We assume that we are given microarray time-course data with well-defined classes. We use the nearest centroid classifier [8], where each class is represented by a single centroid. During the testing phase, the distance between a test profile and each class centroid is found and the test profile is assigned to the label of the nearest centroid.

2 Area Based Profile Alignment

Profile alignment that minimizes the area between two aligned gene expression temporal profiles was introduced in [1]. In [18], the approach was extended to include the effect of unequal time intervals between the measurements. The idea of profile alignment proposed in [18] is discussed in the following. Given a dataset with n samples $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, where $\mathbf{x}_i = [x_{i_1}, x_{i_2}, ..., x_{i_m}]^t$ is an m-dimensional feature vector that represents the expression ratio of gene i at m different time points, $\mathbf{t} = [t_1, t_2, ..., t_m]^t$. The approach of [18] aims to partition $\mathcal{D}$ into k disjoint subsets $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_k$, where $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup ... \cup \mathcal{D}_k$, and $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$, for $\forall i, j, i \neq j$, in such a way that a similarity (dissimilarity) cost function $\phi : \{0, 1\}^{n \times k} \rightarrow \Re$ is maximized (minimized).

The alignment algorithm proposed in [18] takes two features vectors, and produces two new vectors in such a way that the area between "aligned" vectors is minimized. The idea is described in Figure 1. In Figure 1(a) two vectors are shown before alignment. Figure 1(b) shows the "aligned" vectors such that the area between the profiles is minimized, i.e. they were aligned in such a way that the total area covered by the triangle $\{u, v, z\}$ and the polygon $\{z, w, q, e, r, k, h, g, s\}$ is minimized.

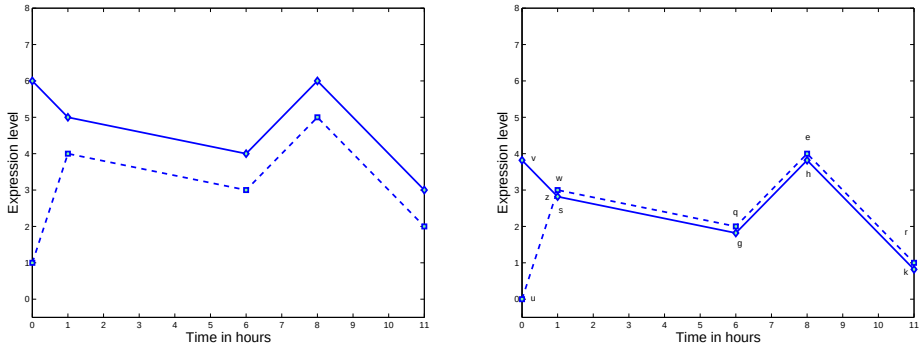


Fig. 1. (a) Two unaligned profiles. (b) The two "aligned" profiles obtained after applying the pairwise alignment, (1)–(4), such that the area between each pair of lines is minimized.

Let, $\mathbf{t} = [t_1, t_2, ..., t_m]^t$ be the vector representing the time points, and the two profiles, $\mathbf{x} = [x_1, x_2, ..., x_m]^t$ and $\mathbf{y} = [y_1, y_2, ..., y_m]^t$, be two profiles, whose expression ratios were measured at time points as given in $\mathbf{t}$, which are to be aligned. The aim is to find a scalar a that minimizes the total area between the two profiles, e.g., between the lines that join the expression ratios. To do this, first, profile $\mathbf{x}$ is aligned in a transformed space and a new vector, $\mathbf{x}'$, is obtained as follows:

$$\mathbf{x}' = [x'_1, x'_2, ..., x'_m]^t \leftarrow \mathbf{x} - x_1\,. \tag{1}$$

Now, let the straight line that joins points $\left(t_{i-1}, x'_{i-1}\right)$ and (t_i, x'_i) be given by $x'_{i-1} + \frac{\left(x'_i - x'_{i-1}\right)}{t_i - t_{i-1}} u$, and for points (t_{i-1}, y_{i-1}) and (t_i, y_i) is given by $y_{i-1} - a + \frac{(y_i - y_{i-1})}{t_i - t_{i-1}} u$, where u corresponds to the x–*axis*. Also, let $\hat{y}_i = y_i - y_{i-1}$, $\hat{x}'_i = x'_i - x'_{i-1}$ and $\hat{t}_i = t_i - t_{i-1}$. A scalar a that minimizes the sum of square errors between $\mathbf{x}$ and $\mathbf{y}$ (aligned), for all $t_1, t_2, ..., t_m$ is obtained using the following sum of integrals:

$$f(a) = \sum_{i=2}^{m} \int_{t_{i-1}}^{t_i} \left[x'_{i-1} + a - y_{i-1} + \frac{\hat{x}'_i - \hat{y}_i}{\hat{t}_i} u \right]^2 du\,, \tag{2}$$

and by means of the first and second order conditions, resulting in:

$$a = -\frac{\sum_{i=2}^{m} \left[\left(x'_{i-1} - y_{i-1} \right) \hat{t}_i + \frac{\hat{x}'_i - \hat{y}_i}{\hat{t}_i} \frac{\hat{t}_i^2}{2} \right]}{\sum_{i=2}^{m} \hat{t}_i}\,. \tag{3}$$

Then, a new vector, $\mathbf{y}'$, is computed as follows:

$$\mathbf{y}' = \mathbf{y} - a\,. \tag{4}$$

Let f_i be defined as $f_i = \frac{(x'_i - x'_{i-1}) - (y'_i - y'_{i-1})}{t_i - t_{i-1}}$. By computing the integrals, the distance between the two new vectors $\mathbf{x}'$ and $\mathbf{y}'$, $d(\mathbf{x}', \mathbf{y}')$, is obtained as:

$$d(\mathbf{x}', \mathbf{y}') = \sum_{i=2}^{m} \left(x'_{i-1} - y'_{i-1}\right)^2 \hat{t}_i + \left(x'_{i-1} - y'_{i-1}\right) f_i \hat{t}_i^2 + f_i^2 \frac{\hat{t}_i^3}{3}. \quad (5)$$

In this paper, we shall denote this distance $d(\mathbf{x}', \mathbf{y}')$, computed by Equation (5), which is the distance between two aligned profile vectors, $\mathbf{x}$ and $\mathbf{y}$, where the time points are given in $\mathbf{t}$, as $d_{\text{PA}}(\mathbf{x}, \mathbf{y}, \mathbf{t})$.

3 Multiple Expression Profile Alignment Algorithms

In this section, we present four heuristic algorithms for multiple gene expression profile alignment. Given a class of gene temporal expression profiles, the goal of each algorithm is to find the centroid, that is a representative profile, for the class such that all profiles in the class remain as close as possible to the centroid after alignment. Each algorithm makes use of the algorithm *pairwise expression profile alignment* (PEPA), which is given in Heuristic 1. This algorithm takes two profile vectors $\mathbf{x}_1 = [x_{1_1}, x_{1_2}, \ldots, x_{1_m}]^t$ and $\mathbf{x}_2 = [x_{2_1}, x_{2_2}, \ldots, x_{2_m}]^t$, and a vector $\mathbf{t} = [t_1, t_2, ..., t_m]^t$, containing m time points at which the measurements of each gene are taken. Applying equations (1)–(4) on $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{t}$, the algorithm obtain two aligned vectors $(\mathbf{x}'_1,\ \mathbf{x}'_2)$. Finally, the algorithm computes and returns the mean profile by taking the averages for each of the m time points of $\mathbf{x}'_1$ and $\mathbf{x}'_2$.

Heuristic 1 ***Pairwise Expression Profile Alignment***

Input: $\mathbf{x}_1 = [x_{1_1}, x_{1_2}, \ldots, x_{1_m}]^t$, $\mathbf{x}_2 = [x_{2_1}, x_{2_2}, \ldots, x_{2_m}]^t$, and $\mathbf{t} = [t_1, t_2, ..., t_m]^t$.
Output: Mean profile, $\boldsymbol{\mu}$, of $\mathbf{x}_1$ and $\mathbf{x}_2$.
$(\mathbf{x}'_1, \mathbf{x}'_2) \leftarrow$ Two aligned vectors obtained after applying equations (1)–(4) on $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{t}$.
$\boldsymbol{\mu} \longleftarrow \frac{1}{2}(\mathbf{x}'_1 + \mathbf{x}'_2)$
return $\boldsymbol{\mu}$

3.1 Random MEPA Algorithm

The algorithm, *random MEPA* (RMEPA) is given in heuristic 2. This algorithm takes a class of n gene expression temporal profiles, $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, where each profile $\mathbf{x}_i,\ 1 \leq i \leq n$, is given as an m-feature vector $\mathbf{x}_i = [x_{i_1}, x_{i_2}, \ldots, x_{i_m}]^t$. The m features are the measurements of gene expression ratios at m different time points. The time points at which the measurements are taken are given in vector $\mathbf{t}$. In the first step, the algorithm finds a pair of genes, $(\mathbf{x}^0, \mathbf{y}^0)$, where $\mathbf{x}^0, \mathbf{y}^0 \in \mathcal{D}^2, \mathbf{x}^0 \neq \mathbf{y}^0$, among all pairs of genes in $\mathcal{D}^2$ such that the distance $d_{\text{PA}}(\mathbf{x}^0, \mathbf{y}^0, \mathbf{t})$, computed using equations (1) - (5), is the minimum among the distances between all pair of profiles in $\mathcal{D}^2$. The algorithm then computes the initial mean, $\boldsymbol{\mu}$, from the profiles $\mathbf{x}^0$ and $\mathbf{y}^0$, using PEPA (Heuristic 1), and remove the profiles $\mathbf{x}^0$ and $\mathbf{y}^0$ from $\mathcal{D}$. Then, in every iteration of the repeat loop, the algorithm randomly selects a profile, $\mathbf{x}$, from $\mathcal{D}$, uses PEPA to obtain a new mean from $\boldsymbol{\mu}$ and $\mathbf{x}$, and assigns the newly computed mean to $\boldsymbol{\mu}$. The process keeps on repeating as long as $\mathcal{D}$ in not empty. The final value of $\boldsymbol{\mu}$ is returned as the centroid of the class $\mathcal{D}$.

Heuristic 2 ***Random MEPA***

Input: $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, and $\mathbf{t}$.
Output: Mean profile, $\boldsymbol{\mu}$, of $\mathcal{D}$.
  $(\mathbf{x}^0, \mathbf{y}^0) \leftarrow \arg\min_{(\mathbf{x},\mathbf{y})\in\mathcal{D}^2, \mathbf{x}\neq\mathbf{y}} d_{\mathrm{PA}}(\mathbf{x}, \mathbf{y}, \mathbf{t})$.
  $\boldsymbol{\mu} \leftarrow$ PEPA($\mathbf{x}^0$, $\mathbf{y}^0$, $\mathbf{t}$).
  $\mathcal{D} \leftarrow \mathcal{D} \setminus \{\mathbf{x}^0, \mathbf{y}^0\}$.
  repeat
    Randomly select $\mathbf{x} \in \mathcal{D}$.
    $\boldsymbol{\mu} \leftarrow$ PEPA($\boldsymbol{\mu}$, $\mathbf{x}$, $\mathbf{t}$).
    $\mathcal{D} \leftarrow \mathcal{D} \setminus \mathbf{x}$.
  until $\mathcal{D} = \emptyset$
return $\boldsymbol{\mu}$.

3.2 Nearest-to-Mean MEPA Algorithm

The *nearest-to-mean MEPA* (NMEPA) algorithm is shown in Heuristic 3. The initialization is same as in RMEPA. However, NMEPA always selects the closest profile, $\mathbf{x} \in \mathcal{D}$, to the current mean to obtain a new mean $\boldsymbol{\mu}$. The last mean is the centroid of $\mathcal{D}$.

Heuristic 3 ***Nearest-to-Mean MEPA***

Input: $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, and $\mathbf{t}$.
Output: Mean profile, $\boldsymbol{\mu}$, of $\mathcal{D}$.
  $(\mathbf{x}^0, \mathbf{y}^0) \leftarrow \arg\min_{(\mathbf{x},\mathbf{y})\in\mathcal{D}^2, \mathbf{x}\neq\mathbf{y}} d_{\mathrm{PA}}(\mathbf{x}, \mathbf{y}, \mathbf{t})$.
  $\boldsymbol{\mu} \leftarrow$ PEPA($\mathbf{x}^0$, $\mathbf{y}^0$, $\mathbf{t}$).
  $\mathcal{D} \leftarrow \mathcal{D} \setminus \{\mathbf{x}^0, \mathbf{y}^0\}$.
  repeat
    $\mathbf{x}' \leftarrow \arg\min_{\mathbf{x}\in\mathcal{D}}, d_{\mathrm{PA}}(\boldsymbol{\mu}, \mathbf{x}, \mathbf{t})$.
    $\boldsymbol{\mu} \leftarrow$ PEPA($\boldsymbol{\mu}$, $\mathbf{x}'$, $\mathbf{t}$).
    $\mathcal{D} \leftarrow \mathcal{D} \setminus \mathbf{x}'$.
  until $\mathcal{D} = \emptyset$
return $\boldsymbol{\mu}$.

3.3 Tree-Based MEPA Algorithm

The algorithm *tree-based MEPA* (TMEPA) algorithm, given in Heuristic 4, applies a tree based approach for the purpose of multiple profile alignment. The entire algorithm consists of an iterative process, which begins with the input argument $\mathcal{D}$. In every iteration, the algorithm first finds a pair of genes, $(\mathbf{x}', \mathbf{y}')$, $\mathbf{x}', \mathbf{y}' \in \mathcal{D}^2, \mathbf{x}' \neq \mathbf{y}'$, among all pairs of genes in $\mathcal{D}^2$ such that the distance $d_{\mathrm{PA}}(\mathbf{x}', \mathbf{y}', \mathbf{t})$ is minimum among the distances between all pairs of profiles in $\mathcal{D}^2$. The algorithm then computes the mean, $\boldsymbol{\mu}'$, of the profiles $\mathbf{x}'$ and $\mathbf{y}'$, using PEPA. Afterwards, it removes the profiles $\mathbf{x}'$ and $\mathbf{y}'$ from $\mathcal{D}$, and adds the newly obtained mean, $\boldsymbol{\mu}'$, to class $\mathcal{D}$. This process continues with the updated $\mathcal{D}$ until there is only one profile left in $\mathcal{D}$. This last profile, $\boldsymbol{\mu}$, is the centroid of $\mathcal{D}$.

Heuristic 4 ***Tree-Based MEPA***

Input: $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, and $\mathbf{t}$.
Output: Mean profile, $\boldsymbol{\mu}$, of $\mathcal{D}$.

repeat
  $(\mathbf{x}', \mathbf{y}') \leftarrow \arg\min_{(\mathbf{x},\mathbf{y})\in\mathcal{D}^2, \mathbf{x}\neq\mathbf{y}} d_{\mathrm{PA}}(\mathbf{x}, \mathbf{y}, \mathbf{t})$.
  $\boldsymbol{\mu}' \leftarrow \mathrm{PEPA}(\mathbf{x}', \mathbf{y}', \mathbf{t})$.
  $\mathcal{D} \leftarrow (\mathcal{D} \cup \{\boldsymbol{\mu}'\}) \setminus \{\mathbf{x}', \mathbf{y}'\}$.
until $|\mathcal{D}| = 1$
$\boldsymbol{\mu} \leftarrow$ The only profile left in $\mathcal{D}$.
return $\boldsymbol{\mu}$.

3.4 Cover-Based MEPA Algorithm

The *cover-based MEPA* (CMEPA) algorithm, shown in Heuristic 5, first aligns all profiles in $\mathcal{D}$ against an arbitrary reference profile, $\mathbf{x} \in \mathcal{D}$. The resulting aligned profiles can be viewed as a *"compact" profile* in such a way that the area enclosed by all profiles is minimized. Once aligned, the mean is obtained by applying PEPA on the *upper-cover* profile and the *lower-cover* profile of the aligned profiles. The upper-cover, $\mathbf{u}$, (lower-cover, $\mathbf{l}$) is a profile vector with m features corresponding to $\mathbf{t}$, such that, at each time point $i, 1 \leq i \leq m$, the value u_i (l_i) is the maximum (minimum) of all aligned profiles at the same time point i. The algorithm starts by randomly selecting a profile $\mathbf{x} \in \mathcal{D}$, as the reference profile, and aligns all profiles in $\mathcal{D} \setminus \mathbf{x}$ to $\mathbf{x}$. Let $\mathcal{D}'$ contain the aligned profiles from $\mathcal{D}$. The algorithm computes the upper-cover and lower-cover, $\mathbf{u}$ and $\mathbf{l}$, of $\mathcal{D}'$. Then, the centroid, $\boldsymbol{\mu}$, of $\mathcal{D}$ is the mean of $\mathbf{u}$ and $\mathbf{l}$ using PEPA.

Heuristic 5 ***Cover-Based MEPA***

Input: $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, and $\mathbf{t}$.
Output: Mean profile, $\boldsymbol{\mu}$, of $\mathcal{D}$.
  $\mathcal{D}' \leftarrow \emptyset$.
  Randomly select $\mathbf{x} \in \mathcal{D}$.
  $\mathcal{D}' \leftarrow \mathcal{D}' \cup \{\mathbf{x}\}$.
  $\mathcal{D} \leftarrow \mathcal{D} \setminus \{\mathbf{x}\}$.
  repeat
    Randomly select $\mathbf{y} \in \mathcal{D}$.
    $\mathbf{y}' \leftarrow$ The aligned profile of $\mathbf{y}$, obtained after aligning $\mathbf{y}$ to $\mathbf{x}$, using equations (1)–(4).
    $\mathcal{D}' \leftarrow \mathcal{D}' \cup \{\mathbf{y}'\}$.
    $\mathcal{D} \leftarrow \mathcal{D} \setminus \{\mathbf{y}\}$.
  until $\mathcal{D} = \emptyset$
  / Compute the upper and lower cover,* $\mathbf{u}$ *and* $\mathbf{l}$*, of* $\mathcal{D}'$ **/*
  for $i = 1$ **to** m **do**
    $u_i \leftarrow \max_{\mathbf{x}\in\mathcal{D}'}(x_i)$
    $l_i \leftarrow \min_{\mathbf{x}\in\mathcal{D}'}(x_i)$
  end for
  $\boldsymbol{\mu} \leftarrow \mathrm{PEPA}(\mathbf{u}, \mathbf{l}, \mathbf{t})$.
return $\boldsymbol{\mu}$.

In Figure 2, we show the results of applying CMEPA on the budding yeast data set. The first column shows the aligned profiles in each of the five classes; the upper-cover

and the lower-cover are shown in the middle column; the two covers are then aligned by PEPA, in the last column, to obtain the centroid of each class.

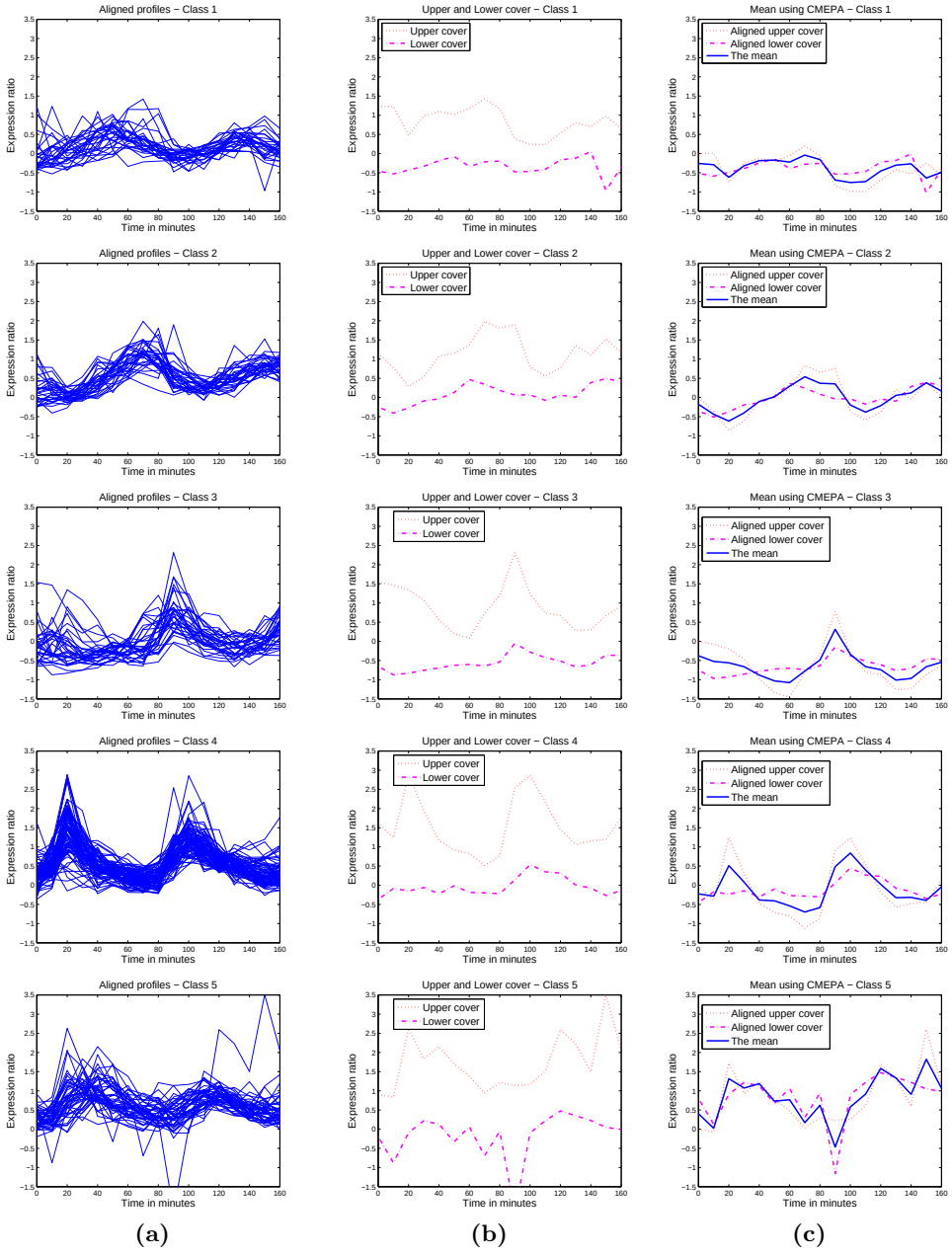


Fig. 2. (a) Aligned profiles in classes 1-5 using CMEPA. (b) The upper and lower covers in classes 1-5 using CMEPA. (c) The centroids of classes 1-5 using CMEPA.

Figure 3 shows the centroids obtained by each method on each class of the budding yeast data set. For AMEPA, in the figure, the centroid is simply the average (arithmetic mean) of all profiles in a given class; no alignment is performed.

4 Nearest Centroid Classification with MEPA Centroids

A fast and simple algorithm for classification is the *nearest centroid* (NC) method [8]. This algorithm assumes that the target classes correspond to individual (single) clusters and uses the cluster means (or centroids) to determine the class of an unlabeled sample point. In other words, in this paper, we take the gene expression profile of a new sample, $\mathbf{x}$, and compare it to each of the class centroids. The centroids (or, the prototype patterns) are computed by a MEPA method. The class, C_i, whose centroid, c_i, that $\mathbf{x}$ is closest to, according to our integral distance function, d_{PA}, is the predicted class for $\mathbf{x}$. That is, during classification, the class label of $\mathbf{x}$ is determined as $\arg\min_{1\leq i\leq n} d_{\mathrm{PA}}(\mathbf{x}, \mathbf{c}_i, \mathbf{t})$, where n is the number of classes.

5 Computational Experiments and Discussions

We performed experiments on the budding yeast, *S. cerevisiae* [19], data set to compare our different MEPA algorithms. We used the nearest centroid classification (NC) method to evaluate the effectiveness of a centroid in representing a class accurately. The effectiveness of a centroid, which determines the quality of its associated MEPA method, is measured by the classification accuracy of NC on the data set. If a centroid represents a given class correctly then it should be at minimum distance (after alignment) with each member of the class than members of the other classes, assuming that the classes are well separated (i.e., the centroids are far from each other).

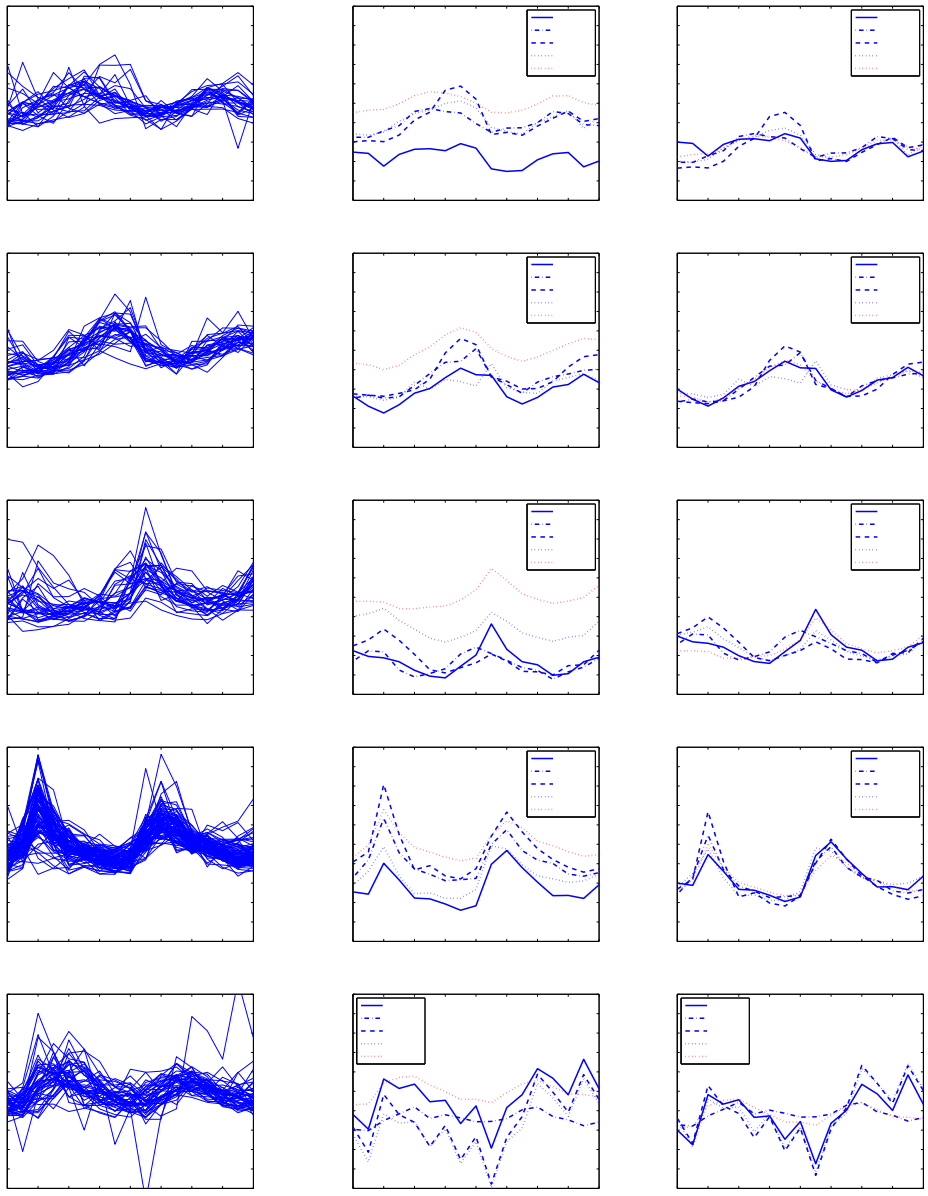
The *S. cerevisiae* data set contains 221 time-varying gene expression profiles distributed in 5 classes of size 32 (class 1: *G2* phase), 84 (class 2: *M* phase), 46 (class 3: *Early G1* phase), 28 (class 4: *Late G1* phase) and 31 (class 5: *S* phase), respectively. In our experiments, however, each class is equally represented in the test set and the training set. That is, we randomly selected 21 samples from each class to form a training set of 105 samples, then randomly selected another 7 samples from each class to form a test set of 35 samples. Therefore, we used 140 distinct samples out of the original set of 221 samples. Since we have a very small training set, we used 3-fold cross-validation on the 105 samples from the training set and returned the accuracy on the test set.

The results of all MEPA methods are shown in Table 1. Column "Acc" is the accuracy on the test set and the remaining columns, afterwards, are the accuracies on each class. The accuracy is simply the percentage of correctly classified samples.

Table 1. Effectiveness of MEPA methods on *S. cerevisiae* data

MEPA Methods	Acc	Class#1	Class#2	Class#3	Class#4	Class#5
NMEPA	54.3%	0%	85.7%	42.9%	71.4%	71.4%
TMEPA	68.6%	42.9%	85.7%	71.4%	71.4%	71.4%
CMEPA	77.1%	28.6%	100%	85.7%	85.7%	85.7%
RMEPA	80.0%	71.4%	71.4%	85.7%	85.7%	85.7%
AMEPA	80.0%	57.1%	100	85.7%	85.7%	71.4%

In Figures 4(a)-(e), we show the unaligned means for all the heuristics and for the five classes. For class 1, we observe that the means for CMEPA, NMEPA, RMEPA



(a) **(b)** **(c)**

Fig. 3. (a) Non-aligned profiles in classes 1-5. (b) Centroids from all MEPA methods. (c) Aligned centroids from all MEPA methods.

and TMEPA are quite dissimilar (in an unaligned sense). It is clearly observed, on the other hand, that the means for these four heuristics are quite similar for the other four classes, 2, 3, 4, and 5. Also, the mean for NMEPA is quite dissimilar to the others, and the profiles' shapes for class 1. That makes the classification task very difficult for class 1, even worse for NMEPA for which the accuacy is 0%. Additionally, class 1 has the smallest cardinality, and a visual inspection of class 1 in Figure 3(a) shows many noisy profiles (outliers). Note also that we randomly selected only 140 profiles from the original data set of 221 profile; in the case of class 5, the centroids in Figure 3(c) and Figure 4(e) are quite dissimilar since the outliers may not have been selected for the training and test set.

Cell cycle is a continuous biological process of *G2/M-M/G1* phases rather than strict *G2, M* and *G1* classes. Therefore, MEPAs were able to perform reasonably well for classifying *G1* and *S* phases. On the other hand, several *S-G2* genes are co-regulated with rather small shifts of time, which can be attributed to the poor results observed in class 1.

In general, more variable centroid estimates can be obtained from the NMEPA since the last profile used would be the one among the most distant profiles from the others, and hence, add a high degree of random variability to the estimate. The TMEPA initially forms multiple well separated profiles but the final estimate is produced by the combination of two highly separated profiles. CMEPA uses both extremes and will counterbalance each other. The centroid estimates by all methods are highly influenced by a few profiles, which remains as an open issue that can be investigated. A weighting function can be utilized to counterbalance the effect of distant profiles. The centroid profile can be defined as a profile that minimizes the sum of the distance measures between the centroid and all the other profiles in a class. The effect of using the sum of absolute distances from the centroid to the profiles can be investigated as well.

6 Conclusions

We have proposed four multiple expression profile alignment (MEPA) methods and evaluated their effectiveness through the class centroids they generate. We applied the nearest centroid classification method to evaluate the quality of all MEPA methods. We plan to use a much larger data set than the *S. cerevisiae* to truly assess the differences between the different MEPA methods. Finally, in [1, 18], the authors used pairwise expression profile alignment algorithms to detect co-expressions in time-varying microarray data; this was achieved by using the distance function, d_{PA}, to group genes into clusters of similar profiles. We are investigating clustering approaches that are based on MEPA methods; in such a case, a characterization of the clustering problem via MEPA is required. A robust analysis and treatment of noisy profiles by any MEPA methods is also needed.

We plan to apply our method to predict the temporal order of gene expressions in the context of gene function, response to signal and gene networks. We also plan to apply the nearest shrunken centroid method [21], the nearest subspace method, and the distance weighted nearest centroid method, and to study the resulting effects of the classification.

Acknowledgements

The authors would like to thank the reviewers for their thoughtful comments and valuable suggestions, which have contributed to the improvement of this manuscript

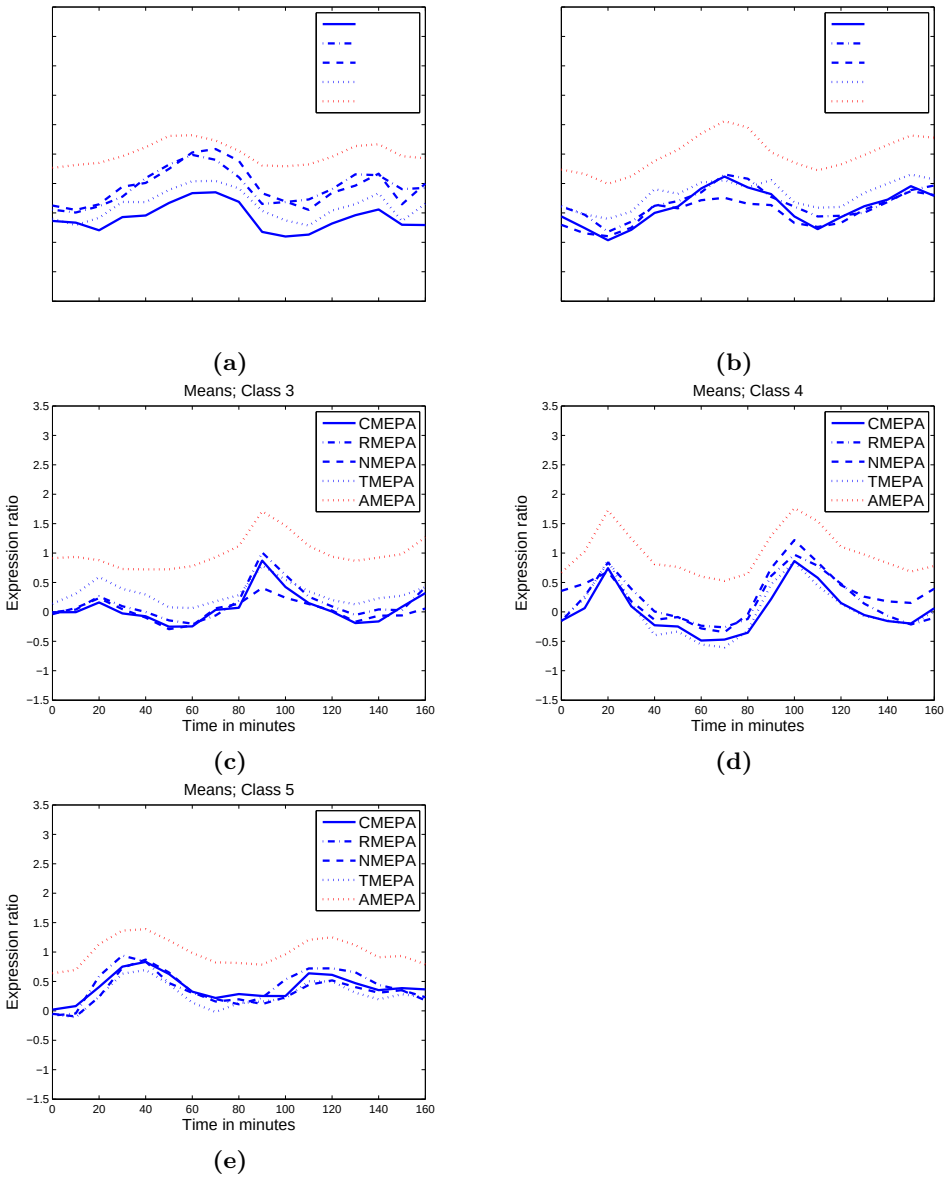


Fig. 4. (a) - (e) Non-aligned centroids of each MEPA method on each class.

in many ways. The work of L. Rueda was partially supported by the Chilean National Council for Technological and Scientific Research, FONDECYT grant No. 1060904. Research of A. Ngom was supported by NSERC.

References

1. A. Bari and L. Rueda. A New Profile Alignment Method for Clustering Gene Expression Data. In proceedings of *19th Conference of the Canadian Society for Computational Studies of Intelligence (AI 2006)*, Quebec City, LNCS, Vol 4013:86–97, Springer, 2006.
2. A. Brazma and J. Vilo. Gene expression data analysis. *FEBS Lett.*, 480:17–24, 2000.
3. L. Bréhélin. Clustering Gene Expression Series with Prior Knowledge. In *Lecture Notes in Computer Science*, volume 3692, pages 27–38, October 2005.
4. S. Chu, J. DeRisi, M. Eisen, J. Mulholland, D. Botstein, P. Brown, and I. Herskowitz. The transcriptional program of sporulation in budding yeast. *Science*, 282:699–705, 1998.
5. S. Drăghici. *Data Analysis Tools for DNA Microarrays*. Chapman & Hall, 2003.
6. R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, Inc., New York, NY, 2nd edition, 2000.
7. M. Eisen, P. Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. In *Proc. Natl Acad. Sci.*, volume 95, pages 14863–14868, USA, 1998.
8. T. Hasti, R. Tibshiran and J. Friedman. The Elements of Statistical Learning Springer Series in Statistics. Springer Verlag, New York; 2001
9. L. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: identification and analysis of coexpressed genes. *Genome Res.*, 9:1106–1115, 1999.
10. V. Iyer, M. Eisen, D. Ross, G. Schuler, T. Moore, J. Lee, J. Trent, L. Staudt, Jr. J. Hudson, and M. Boguski. The transcriptional program in the response of human fibroblasts to serum. *Science*, 283:83–87, 1999.
11. U. Maulik and S. Bandyopadhyay. Performance Evaluation of Some Clustering Algorithms and Validity Indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1650–1654, 2002.
12. S. Peddada, E. Lobenhofer, L. Li, C. Afshari, C. Weinberg, and D. Umbach. Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference. *Bioinformatics*, 19(7):834–841, 2003.
13. M. Ramoni, P. Sebastiani and I. Kohane. Cluster analysis of gene expression dynamics. *Proc. Natl Acad. Sci. USA*, 99(14), 9121–9126, 2002.
14. A. Schliep, A. Schonhuth and C. Steinhoff. Using hidden Markov models to analyze gene expression time course data. *Bioinformatics*, 2003, 19, I264–I272.
15. G. Sherlock. Analysis of large-scale gene expression data. *Curr. Opin. Immunol.*, 12:201–205, 2000.
16. P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. Lander, and T. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci.* 96(6): 2907–2912. 1999.
17. S. Tavazoie, J. Hughes, M. Campbell, R. Cho, and G. Church. Systematic determination of genetic network architecture. *Nat. Genet.* 22: 281–285. 1999.
18. L. Rueda and A. Bari. Clustering Temporal Gene Expression Data with Unequal Time Intervals. In proceedings of 2*nd International Conference on Bio-Inspired Models of Network, Information, and Computing Systems (BIONETICS 2007)*, Bioinformatics Track. Budapest, Hungary, ICST 978-963-9799-11-0, 2007.
19. R. J. Cho, M.J.Campbell, E. A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. G. Wolfsberg, A. E. Gabrielian, D. Landsman, D. J. Lockhart, R. W. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2(1):65–73, 1998.
20. Z-X. Yin and J-H. Chiang. Novel Algorithm for Coexpression Detection in Time-Varying Microarray Data Sets. *IEEE TCBB*, 5(1):120–135, 2008.
21. R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. PNAS Vol. 99:6567-6572, 2002.