Clustering Temporal Gene Expression Data with Unequal Time Intervals

Luis Rueda
Department of Computer Science
University of Concepción, Edmundo Larenas 215
Concepción, 4030000, Chile
lrueda@inf.udec.cl

Ataul Bari
School of Computer Science
University of Windsor, 401 Sunset Avenue
Windsor, ON, N9B 3P4, Canada
bari1@uwindsor.ca

ABSTRACT

We have focused on the problem of clustering time-series gene expression data. We present a novel algorithm for clustering gene temporal expression profile microarray data, which is fairly simple but powerful enough to find an efficient distribution of genes over clusters. Using a variant of a clustering index can effectively decide upon the optimal number of clusters for a given dataset. The clustering method is based on a profile-alignment approach, which we propose and that minimizes the (square) area between two aligned vector profiles, to hierarchically cluster microarray time series data. The effectiveness of the proposed approach is demonstrated on two well-known, yeast and serum.

Keywords

Gene expression, clustering, time-series profiling

1. INTRODUCTION

Clustering 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 [6]. Clustering gene expression data given in terms of time series is different from a general clustering problem, because exchanging two time points delivers quite different results, while it may not be biologically meaningful. Many unsupervised methods for gene clustering based on the similarity (or dissimilarity) of their microarray temporal profiles have been proposed in the past few years [1, 2, 6, 11]. 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 [2]. Other methods based on correlation measures have been proposed for clustering genes using microarray time series data [3, 7]. The method proposed in [3] requires computing the mean expression levels of some candidate profiles using some pre-identified, arbitrarily selected profiles. In [7], 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 cluster genes using the ideas of order-restricted inference, where estimation makes use of known inequalities among parameters [10]. 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 $\mathbf{x}_1$ and $\mathbf{x}_2$ fall into the same cluster if they show similar profiles in terms of directions of the changes of expression ratios (e.g. up-up-up-down-down), regardless how big/small is the change.

In [12], a minimum-square-error profile alignment approach to cluster 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 two time points at which individual measurements are taken.

In this paper, we propose a profile alignment approach to cluster temporal microarray data that minimizes the area between two aligned profiles. The hierarchical clustering algorithm uses a variant of a well-known clustering validity index that optimizes the number of clusters [5, 9]. The profile alignment that we propose in this paper is different from that of [12] in the sense that: (i) the approach proposed in this paper 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. Experiments on serum data and on pre-clustered yeast data show the effectiveness of the proposed method.

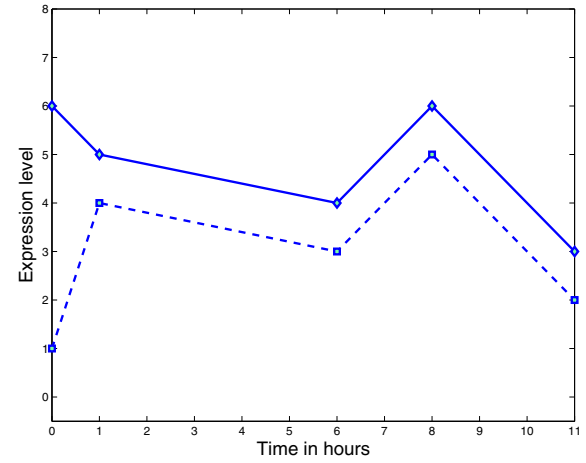
2. AREA-BASED PROFILE ALIGNMENT

Consider a dataset with n samples $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$,

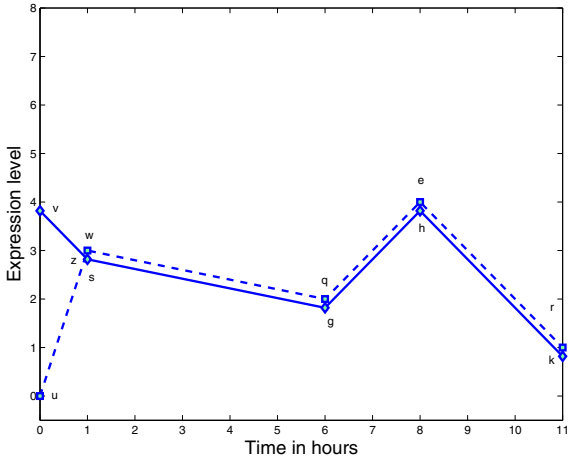
Bionetics '07, December 10-13, 2007, Budapest, Hungary

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 aim is 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).

We propose an efficient alignment algorithm that 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.



(a)



(b)

Figure 1: (a) Two unaligned profiles. (b) The two "aligned" profiles obtained after applying (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 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, we first "slide" down $\mathbf{x}$ and obtain a new vector, $\mathbf{x}'$, as follows:

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

Now, assume that the straight line that joins points (t_{i-1}, x'_{i-1}) and (t_i, x'_i) is given by $x'_{i-1} + \frac{(x'_i - x'_{i-1})}{t_i - t_{i-1}} u$, 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". Let us use the following notation: $\hat{y}_i = y_i - y_{i-1}$, $\hat{x}'_i = x'_i - x'_{i-1}$ and $\hat{t}_i = t_i - t_{i-1}$. Since we want to minimize the area between $\mathbf{x}$ and $\mathbf{y}$ (aligned), for all $t_1, t_2, ..., t_m$, we need to find a that minimizes the sum of square errors between each pair of lines, equivalent to the following sum of integrals:

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

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

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

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

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

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

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

The second order condition is satisfied, i.e. $\frac{\partial^2 f}{\partial^2 a} = \frac{\partial}{\partial a} 2 \sum_{i=1}^{m} (x_{1_i} - x_{2_i} + a) = 2m > 0$. This indicates that a minimum is reached at $a = \frac{1}{m} \sum_{i=1}^{m} (x_{2_i} - x_{1_i})$. Using (1) - (4) to obtain the value of a, we first align two profile vectors and then compute a distance function in the usual manner. It is not difficult to show that this alignment, used in conjunction with any metric d, is also a metric [12]. Note that once the alignment is applied, any metric d can be used. In this paper, we have used the distance given in (5), since we meant to minimize the area between two aligned profiles.

The justification of the proposed alignment is depicted in Figure 2 with examples of temporal profiles for three pairs of genes. Figures 2(a), (b) and (c) show three pairs of genes without alignment. Using the Pearson correlation distance [10], genes in Figure 2(b) are most likely to be clustered together (as they produce the largest value for correlation coefficient among all three pairs of genes, which is 0.9053). Note that the prime interest is to cluster genes based on the variations of the expression ratios at the different time points. The aim is to find variations in terms of changes (or not) between different time points, independently of the "scale" in which the ratios lie. Then, genes from Figure 2(c)

would be better candidates to be clustered together than the genes in Figs. 2(a) and (b). However, the value of the correlation coefficient between the pairs of genes in Figure 2(c) is the minimum (0.8039) among all three pairs of genes. Figures 2 (d), (e) and (f) show the pairs of genes after aligning the genes from Figures 2(a), (b) and (c), respectively. A simple visual inspection shows that the genes in Figure 2(f) are closer to each other compared to the genes from the other two figures, Figures 3(d) and (e).

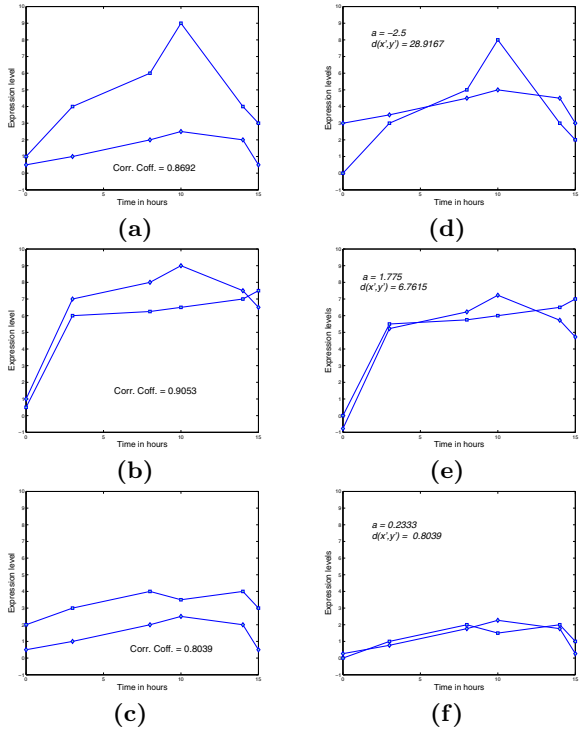


Figure 2: (a) Two genes that are likely to be clustered as in [10], although the difference among them in terms of rate of expression ratio changes between different time points is large. (b) Two genes with different profiles that are likely to be clustered together by correlation-based methods. (c) Two genes with similar profiles in terms of rate of expression ratio changes between different time points that may not be clustered together by the method proposed in [10] and the correlation-based methods. (d) Result after aligning the two genes from (a). (e) Result after aligning two genes from (b). (f) Result after aligning two genes from (c). After applying the proposed profile alignment, the differences between the genes in (d) and (e) are more notorious than in (f).

3. THE CLUSTERING ALGORITHM

Hierarchical agglomerative clustering is the method used in this paper. We apply *complete linkage* or *furthest neighbors* [4], which computes the distance between the furthest pair of points for each pair of clusters and merges the pair of clusters that has the minimum distance among all such distances between the pair of clusters under consideration. The generalized algorithm of *hierarchical agglomerative clustering* is slightly modified to obtain the desired number of clusters instead of a hierarchy of clusters, which is given in Algorithm 1, *Agglomerative-Clustering*.

The *Agglomerative-Clustering* algorithm receives two parameters as input, a complete microarray temporal dataset, $\mathcal{D}$, and the desired number of clusters k, and returns the dataset after partitioning it into k clusters. The best number of clusters k^* is chosen to maximize the $\mathcal{I}$-index as follows:

$$\mathcal{I}(k) = \left(\frac{1}{k}\right)^q \times \left(\frac{E_1}{E_k} \times \mathcal{D}_k\right)^p \quad (6)$$

where, n is the total number of samples in the dataset, $E_k = \sum_{i=1}^{k}\sum_{j=1}^{n} u_{ij} d(\mathbf{x}_j, \boldsymbol{\mu}_i)$, $\mathcal{D}_k = max_{i,j=1}^{k} d(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)$, $\{u_{ij}\}_{k\times n}$ is the partition (or membership) matrix for the data, $\boldsymbol{\mu}_i$ is the center of cluster $\mathcal{D}_i$, k is the number of clusters, and $d(.,.)$ is the distance computed as in (5).

The decision rule is based on the *furthest-neighbor* distance between two clusters, which is computed using formula (5). The latter involves the alignment of each pair of profiles before applying a conventional (formula (5) in our case) distance function.

Algorithm *Agglomerative-Clustering*

Input: The dataset, $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, and k, the desired number of clusters.
Output: k disjoint subsets $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_k$.

Create n clusters, $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_n$, where $\mathcal{D}_i = \{\mathbf{x}_i\}$
$\mathcal{D}_{currentClustersSet} \longleftarrow \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_n\}$
for $q \leftarrow n$ **down to** k **do**
  For each pair of cluster $(\mathcal{D}_i, \mathcal{D}_j)$, find furthest neighbor $(\mathbf{x}_i, \mathbf{x}_j)$, using the (3) and (4) alignment and computing the distance as in (5).
  Select $(\mathcal{D}_k, \mathcal{D}_l)$ as the pair of clusters with the closest furthest neighbors.
  $\mathcal{D}_{mergedClusters} \leftarrow \{\mathcal{D}_k \cup \mathcal{D}_l\}$.
  $\mathcal{D}_{currentClustersSet} \leftarrow$
    $\{\mathcal{D}_{currentClustersSet} \cup \mathcal{D}_{mergedClusters}\} \setminus \{\mathcal{D}_k, \mathcal{D}_l\}$
end for
return $\mathcal{D}_{currentClustersSet}$.

The partition matrix $\{u_{ij}\}$ is defined as a membership function such that $u_{ij} = 1$, if $\mathbf{x}_j$ belongs to cluster $\mathcal{D}_i$, and zero otherwise. To compute the mean of cluster $\mathcal{D}_i$, $\boldsymbol{\mu}_i$, we follow a greedy approach that computes the mean and scatter for each cluster. It picks the first profile from the cluster and assigns it to be the current mean. Then, it picks the next gene profile from the cluster, applies the pairwise alignment with this profile to the current mean, and updates the current mean by taking the average of these two aligned profiles. The process continues until all the gene profiles are aligned and the current mean is updated correspondingly for each profile. Note that it is done in this way, otherwise it should be done like "multiple alignment", which is a non-trivial issue that remains open.

The cluster mean is used to compute the scatter of a cluster. Given a cluster $\mathcal{D}_i = \{\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, ..., \mathbf{x}_{i_n}\}$, using (1) - (4) each $\mathbf{x}_{i_j}$ is aligned with the mean $\boldsymbol{\mu}_i$ producing a set of aligned profiles $\mathcal{D}_i' = \{\mathbf{y}_{i_1}, \mathbf{y}_{i_2}, ..., \mathbf{y}_{i_n}\}$. The scatter is computed as $E_i = \sum_{j=1}^{i_n} d(\mathbf{y}_{i_j}, \boldsymbol{\mu}_i)$, where $d(.,.)$ is the distance computed as in (5).

4. EXPERIMENTAL RESULTS

We have tested the performance of our clustering method

that performs profile alignment combined with agglomerative clustering (PAAC) on two well-known datasets, the serum [8] and yeast [3] datasets. For the serum dataset, we conducted experiments on a subset containing 517 out of 8,613 genes[1]. We considered a range for potential numbers of clusters, which includes values of k that lie between $\lceil\sqrt{n/2}\rceil$ and $\lfloor\sqrt{3n/2}\rfloor$, i.e. $k = 16$ to 27. For each k, the $\mathcal{I}$-index was computed using values of q from 0.3 to 1.0. We have selected $q = 0.7$, since we want to favor a large number of clusters, as $q = 0.7 < 2.0 = p$. The value 0.7 was found experimentally and confirmed with pre-clustered yeast data for which we found our method provides a high classification accuracy, as seen later. For $q = 0.7$, the value of the index reaches to a maximum level when $k = 21$. The plots are shown in Figure 4, where each plot represents a cluster. The x-axis in each plot represents the time in hours and the y-axis represents the expression ratio.

We have tested the results of our method with the results obtained by the Pearson correlation distance and the Spearman correlation distance methods. Clustering using the Pearson correlation distance is given in Figure 4. The comparison among the plots for PAAC and Pearson reveals the effectiveness of the method. For example, PAAC left clusters 1 to 5 containing a single gene each (IDs 328692, 470934, 361247, 147050 and 310406, respectively). The Pearson correlation method, however, placed these genes in clusters 19, 2, 2, 11 and 19, respectively . By visual inspection of all the temporal expression profiles, we noticed that these genes are *differentially expressed* and should be left alone in separate clusters, which is clearly done by PAAC. Also, PAAC produced four clusters containing only two profiles each, clusters 9 (IDs 356635 and 429460), 11 (IDs 26474 and 254436), 13 (IDs 280768 and 416842) and 16 (IDs 130476 and 130482). The Pearson correlation method clustered these genes as follows: 356635 and 429460 in cluster 16, 26474 and 254436 in cluster 21, 280768 and 416842 in cluster 2 and 130476 and 130482 in cluster 16 (Figure 4). Although the Pearson correlation method placed each pair of genes in the same cluster, it also placed some other genes with them. By looking at the plots of the profiles of the clusters produced by the Pearson correlation method and comparing them to the plots of the clusters of the corresponding genes produced by PAAC, it is clear that these pairs of genes are differentially expressed. For the Spearman correlation method, though non-linear, we observed that it outputs results comparable to those of the Pearson correlation, and is not able to identify and separate differentially expressed genes properly as good as PAAC (plots not shown).

In order to provide a biological significance of our results, we applied our method to a dataset containing the changes in gene expression during the cell cycle of the budding yeast, *S. cerevisiae*[2] [13], in which expression ratios were measured at seventeen different time points, from 0 min. to 160 min. with an interval of 10 mins. The experiment monitored 6,220 transcripts for cell cycle-dependent periodicity and 221 functionally characterized genes with periodic fluctuation were listed in Table 1 of [13]. We applied PAAC to these 221 genes, using the same parameters as those for the serum dataset, obtaining the best number of clusters $k = 28$. The clusters obtained using PAAC are shown in Figure 5. We also used the Pearson correlation coefficient to cluster the dataset (plots not shown). We observed that PAAC separates the genes by profiles in a wise manner, while the Pearson correlation method is not able to capture all variations in the time series.

Finally, we list in Tables 1-3, the 221 genes, where for each gene it is shown the cluster number PAAC assigns and the class (phase) that the gene is categorized as in Table 1 of [13]. An objective measure for comparing the two clusterings has been taken by computing the overall classification accuracy, which is computed as the number of genes that PAAC *correctly* assigned to one of the phases. The *correct* class (phase) is the one that PAAC assigns the largest number of genes. The overall classification accuracy was computed as the average of the individual accuracies for each cluster, resulting in 83.47%, which is very high considering the fact that PAAC is an *unsupervised* classification algorithm.

5. CONCLUSIONS

We have proposed a method to cluster gene expression temporal profile microarray data. On two well-known real-life datasets, we have demonstrated that using hierarchical clustering with our method for similarity measure produced superior results when compared to that of the Pearson and Spearman correlation similarity measures.

We have applied a variant of the $\mathcal{I}$-index that can make a trade-off between minimizing the number of useful clusters and keeping the distinctness of individual clusters. We have also shown the biological significance of the results obtained by computing the classification accuracy of our method in pre-clustered yeast data - the accuracy was over 83%.

PAAC can be used for effective clustering of gene expression temporal profile microarray data. Although we have shown the effectiveness of the method in microarray time-series datasets, we are planning to investigate the effectiveness of the method as well in dose-response microarray datasets, and other time series microarray data.

Acknowledgments:.

This research has been partially supported by the Chilean National Council for Technological and Scientific Research, FONDECYT grant No. 1060904.

6. REFERENCES

[1] A. Brazma and J. Vilo. Gene expression data analysis. *FEBS Lett.*, 480:17–24, 2000.

[2] L. Bréhélin. Clustering Gene Expression Series with Prior Knowledge. In *Lecture Notes in Computer Science*, volume 3692, pages 27–38, October 2005.

[3] 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.

[4] S. Drăghici. *Data Analysis Tools for DNA Microarrays*. Chapman & Hall, 2003.

[5] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, Inc., New York, NY, 2nd edition, 2000.

[6] 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.

[1]http://genome-www.stanford.edu/serum/.

[2]http://genome-www.stanford.edu/cellcycle/data/rawdata/individual.html.

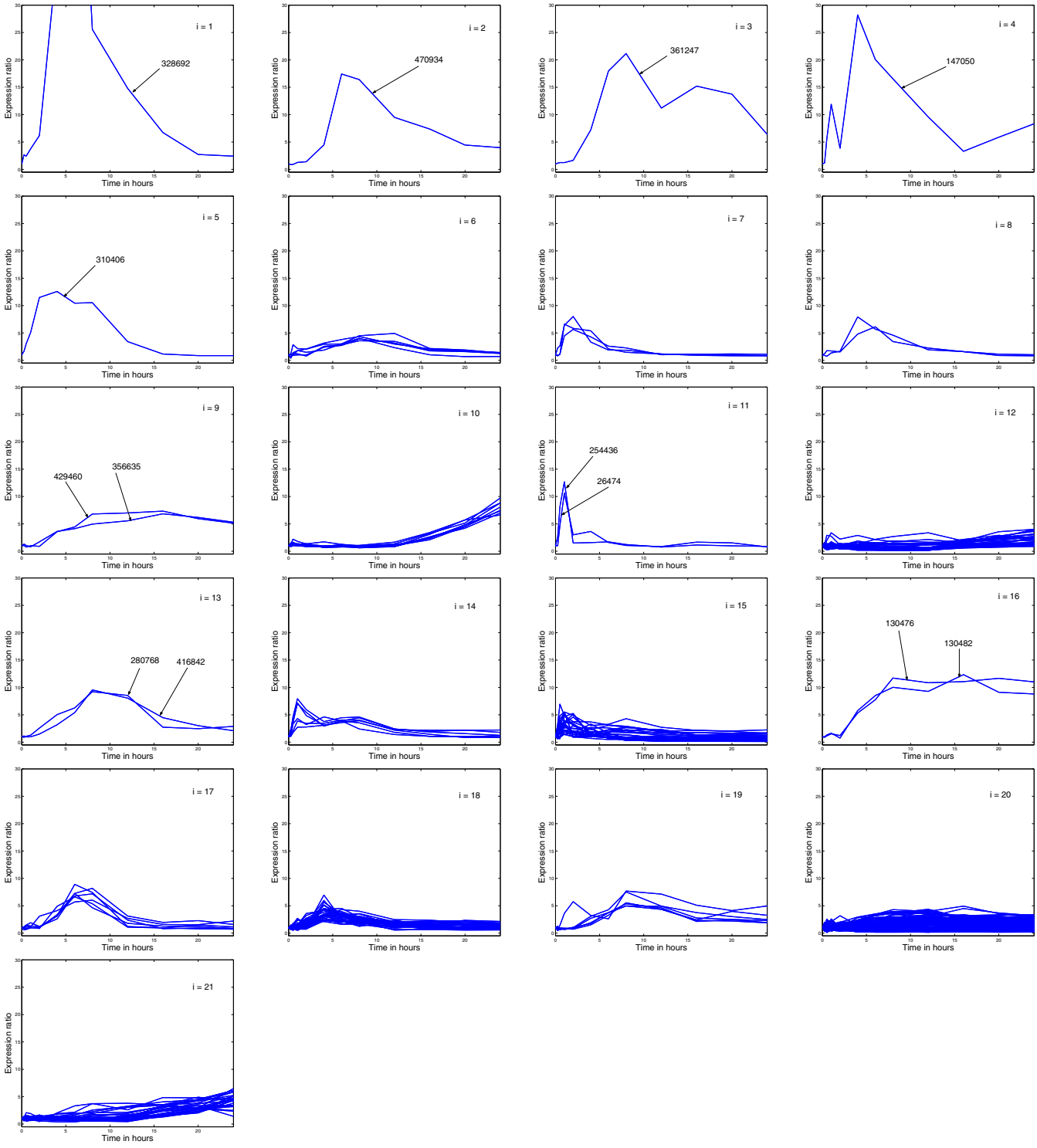


Figure 3: Different clusters obtained using PAAC on the 517 gene temporal expression profiles, where $k = 21$.

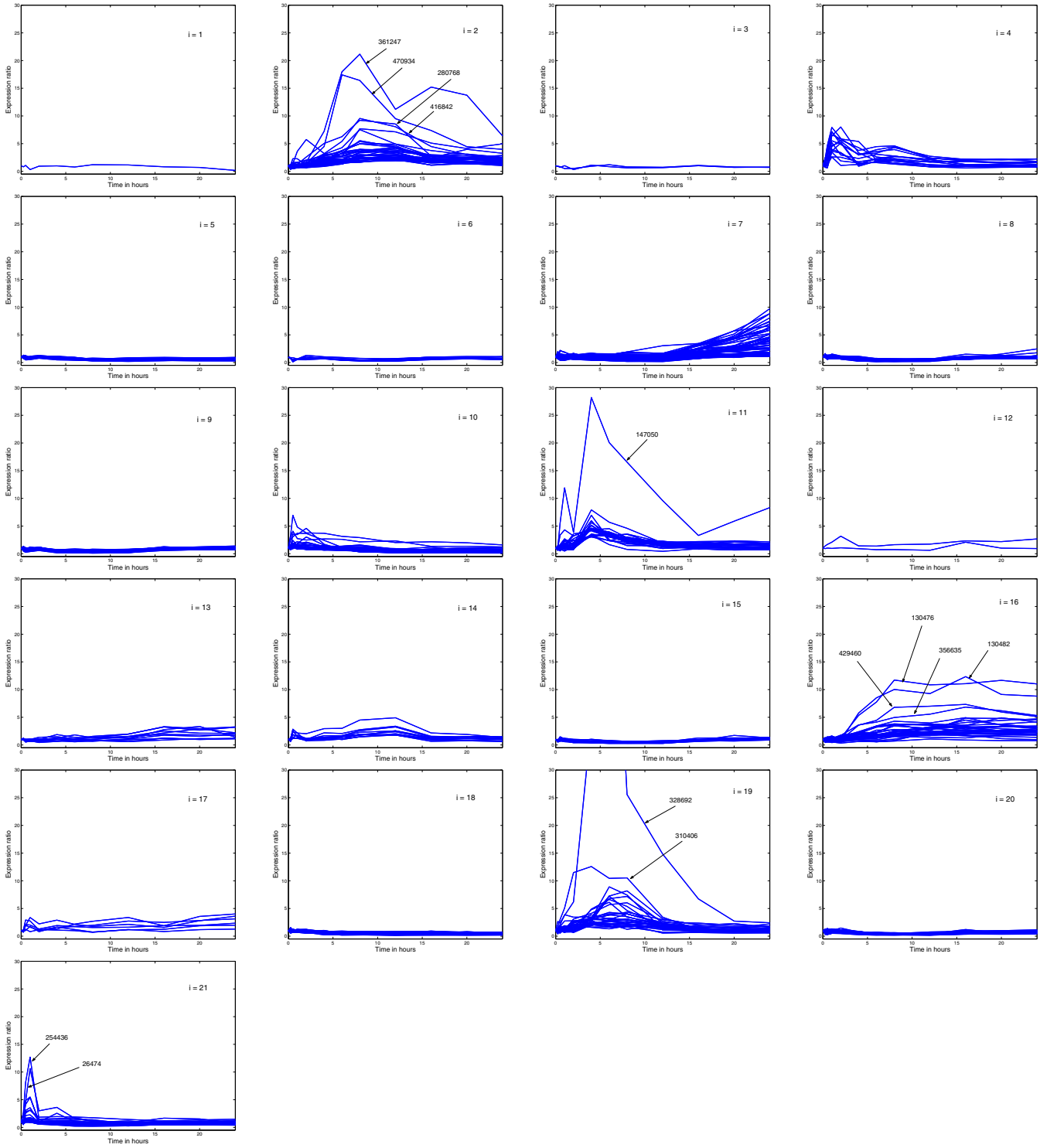


Figure 4: Different clusters obtained using the Pearson correlation distance on the 517 gene temporal expression profiles, where $k = 21$.

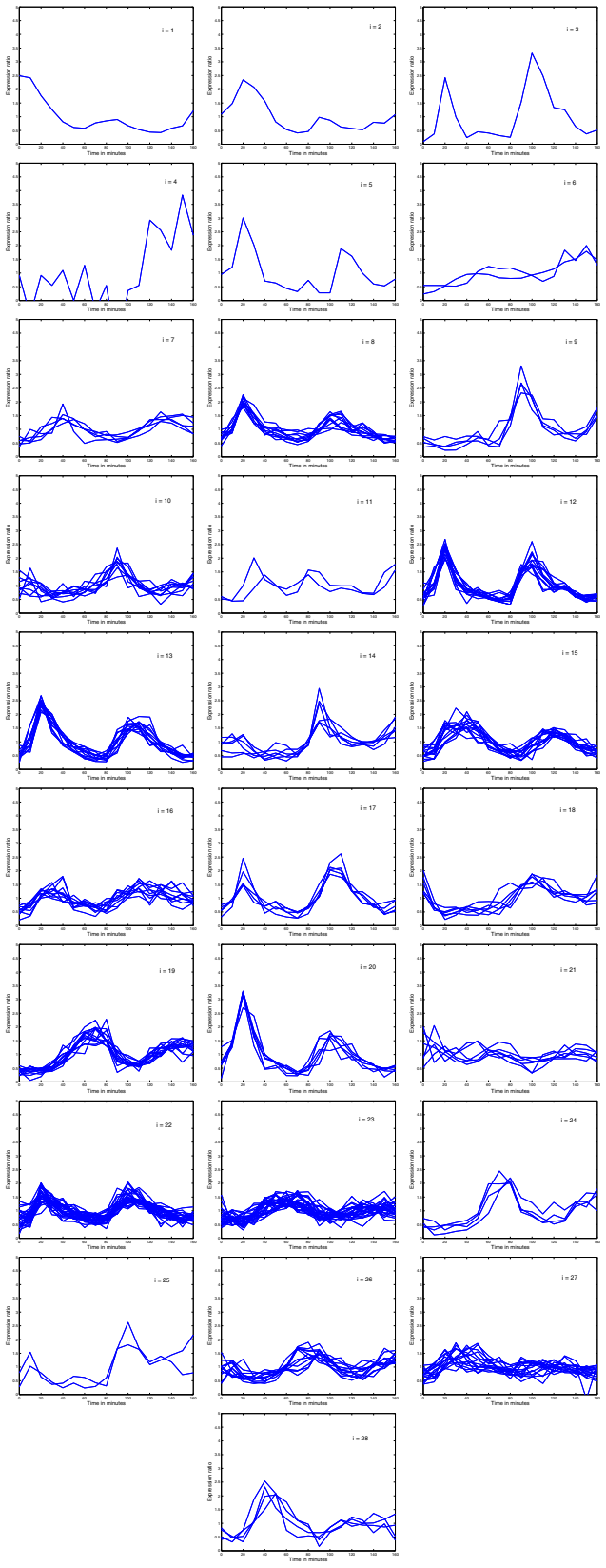


Figure 5: Different clusters obtained using PAAC on the 221 gene temporal expression profiles from Table 1 of [13], where $k = 28$.

[7] L. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: identification and analysis of coexpressed genes. *Genome Res.*, 9:1106–1115, 1999.

[8] 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.

[9] 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.

[10] 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.

[11] G. Sherlock. Analysis of large-scale gene expression data. *Curr. Opin. Immunol.*, 12:201–205, 2000.

[12] 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.

[13] 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.

PAAC	Gene	Phase	PAAC	Gene	Phase	PAAC	Gene	Phase
1	YBR067c/TIP1	Early G_1	15	YEL061c/CIN8	S	22	YLL002w/	Late G_1
2	YGL055W/OLE1	Early G_1	15	YGR140W/CBF2	S	22	YLR457C/NBP1	Late G_1
3	YDL227C/HO	Late G_1	15	YHR172W/	S	22	YPL057C/SUR1	Late G_1
4	YAL001C/TFC3	S	15	YLR045c/STU2	S	22	YPL124W/NIP29	Late G_1
5	YJL115W/ASF1	S	15	YNL126W/	S	22	YBR275c/RIF1	S
6	YDL198C/SHM1	G_2	15	YPR141C/KAR3	S	22	YCR065w/HCM1	S
6	YDR146c/SWI5	M	15	YJR006W/HUS2	S	22	YDL197C/ASF2	S
7	YKL049C/CSE4	G_2	15	YER001w/MNN1	S	22	YKL127W/PGM1	S
7	YKL048C/ELM1	G_2	15	YER003c/PMI40	S	22	YDL095W/PMT1	S
7	YER069w/"ARG5,6"	G_2	16	YNL225C/	Late G_1	23	YLR210W/CLB4	S
7	YJR112W/NNF1	G_2	16	YDR224c/HTB1	Late G_1	23	YDR150w/NUM1	M
7	YPR167C/MET16	M	16	YDR225w/HTA1	Late G_1	23	YMR198W/CIK1	S
8	YFL008W/SMC1	Late G_1	16	YGL200C/EMP24	Late G_1	23	YBL052c/	S
8	YMR078C/CHL12	Late G_1	16	YNL272C/SEC2	Late G_1	23	YIL126W/STH1	S
8	YPL209C/IPL1	Late G_1	16	YDR488c/PAC11	S	23	YCR035c/	S
8	YGR152C/RSR1	Late G_1	16	YBL002w/HTB2	S	23	YIL050W/	G_2
8	YIL159W/	Late G_1	16	YBL003c/HTA2	S	23	YBL097w/	G_2
8	YNL233W/	Late G_1	16	YKL067W/YNK1	S	23	YJL099W/CHS6	G_2
8	YKL045W/PRI2	Late G_1	16	YER118c/SSU81	S	23	YJR076C/CDC11	G_2
8	YNL312W/RFA2	Late G_1	17	YJL074C/SMC3	Late G_1	23	YCR084c/TUP1	G_2
8	YML061C/PIF1	Late G_1	17	YGR041W/	Late G_1	23	YGL255W/ZRT1	G_2
8	YLR233C/EST1	Late G_1	17	YNL082W/PMS1	Late G_1	23	YJL137c/GLG2	G_2
8	YOR026W/BUB3	S	17	YOL090W/MSH2	Late G_1	23	YCR073c/	G_2
9	YDL179w/	Early G_1	17	YHR153c/SPO16	Late G_1	23	YDR389w/SAC7	G_2
9	YLR079w/SIC1	Early G_1	18	YCR005c/CIT2	Early G_1	23	YKL068W/NUP100	G_2
9	YJL157C/FAR1	Early G_1	18	YCL040w/GLK1	Early G_1	23	YGR092W/DBF2	M
9	YKL185W/ASH1	Early G_1	18	YLR258W/GSY2	Early G_1	23	YOR058C/ASE1	M
10	YJL194W/CDC6	Early G_1	18	YNL173C/	Late G_1	23	YPL242C/	M
10	YLR274W/CDC46	Early G_1	18	YOR317W/FAA1	Late G_1	23	YCL037c/SRO9	M
10	YPR019W/CDC54	Early G_1	18	YNL073W/MSK1	S	23	YKL130C/	M
10	YHR005c/GPA1	Early G_1	19	YIL106W/MOB1	G_2	23	YNL053W/MSG5	M
10	YGR183C/QCR9	Early G_1	19	YCL014w/BUD3	G_2	23	YIL162W/SUC2	M
10	YLR273C/PIG1	Early G_1	19	YGR108W/CLB1	M	23	YDL048c/STP4	M
10	YLL040c/	Early G_1	19	YPR119W/	M	23	YHR152w/SPO12	M
10	YHR038W/	Late G_1	19	YBR138c/	M	23	YKL129C/MYO3	M
10	YAL040C/CLN3	M	19	YHR023w/MYO1	M	24	YPL058C/PDR12	Early G_1
11	YDR277c/MTH1	S	19	YOL069W/NUF2	M	24	YBR038w/CHS2	G_2
11	YML091C/RPM2	S	19	YJR092W/BUD4	M	24	YGL116W/CDC20	M
12	YDL127w/PCL2	Late G_1	19	YLR353W/BUD8	M	24	YGR143W/SKN1	M
12	YPR120C/	Late G_1	19	YMR001C/CDC5	M	25	YLR286C/CTS1	Late G_1
12	YDL003W/RHC21	Late G_1	19	YGL021W/ALK1	M	25	YGL089C/MF(alpha)2	Late G_1
12	YAR007C/RFA1	Late G_1	19	YLR131c/ACE2	M	26	YBR200w/BEM1	Early G_1
12	YBL035c/POL12	Late G_1	19	YOR025W/HST3	M	26	YBL023c/MCM2	Early G_1
12	YBR088c/POL30	Late G_1	20	YGR109C/CLB6	Late G_1	26	YBR202w/CDC47	Early G_1
12	YDL164C/CDC9	Late G_1	20	YNL289W/PCL1	Late G_1	26	YEL032w/MCM3	Early G_1
12	YML102W/	Late G_1	20	YLR313C/	Late G_1	26	YLR395C/COX8	Early G_1
12	YPR175W/DPB2	Late G_1	20	YPR018W/RLF2	Late G_1	26	YMR256c/COX7	Early G_1
12	YDR097C/	Late G_1	20	YPL153C/SPK1	Late G_1	26	R281W/YOR1	Early G_1
12	YLR032w/RAD5	Late G_1	20	YBR070c/	Late G_1	26	YOR316C/COT1	Late G_1
12	YML027W/YOX1	Late G_1	21	YJR159W/SOR1	G_2	26	YCR042c/TSM1	M
12	YMR179W/SPT21	Late G_1	21	YBR104w/YMC2	G_2	26	YOR229W/	M
13	YJL187C/SWE1	Late G_1	21	YLR014c/PPR1	G_2	26	YDL138W/RGT2	M
13	YPL256C/CLN2	Late G_1	21	YOR274W/MOD5	G_2	26	YIL167W/	M
13	YMR076C/PDS5	Late G_1	21	YDR464w/SPP41	G_2	27	YNR016C/ACC1	Early G_1
13	YER070w/RNR1	Late G_1	21	YLL046c/RNP1	G_2	27	YBR160w/CDC28	Late G_1
13	YLR103c/CDC45	Late G_1	22	YER111c/SWI4	Early G_1	27	YBR252w/DUT1	Late G_1
13	YNL102W/CDC17	Late G_1	22	YOR373W/NUD1	Early G_1	27	YBR278w/DPB3	Late G_1
13	YOR074C/CDC21	Late G_1	22	YKL092C/BUD2	Early G_1	27	YDR297w/SUR2	Late G_1
13	YKL113C/RAD27	Late G_1	22	YMR199W/CLN1	Late G_1	27	YDL155W/CLB3	S
13	YLR383W/	Late G_1	22	YKL042W/	Late G_1	27	YFR037C/	S
13	YML060W/OGG1	Late G_1	22	YLR212C/TUB4	Late G_1	27	YPL016W/SWI1	S
13	YIL140W/SRO4	S	22	YPL241C/CIN2	Late G_1	27	YDL093W/PMT5	S
13	YAR008W/	S	22	YDR507c/GIN4	Late G_1	27	YKR001C/SPO15	S
14	YDL181W/INH1	Early G_1	22	YGL027C/CWH41	Late G_1	27	YER016w/BIM1	S
14	YML110C/	Early G_1	22	YJL173C/RFA3	Late G_1	27	YER017c/AFG3	S
14	YIL009W/FAA3	Early G_1	22	YNL262W/POL2	Late G_1	27	YPR111W/DBF20	G_2
14	YBR083w/TEC1	Early G_1	22	YDL101C/DUN1	Late G_1	27	YOR188W/MSB1	G_2
14	YPL187W/MF(alpha)1	Late G_1	22	YLR234W/TOP3	Late G_1	27	YJL092W/HPR5	G_2
14	YJR148W/TWT2	Late G_1	22	YML021C/UNG1	Late G_1	27	YKL032C/IXR1	G_2
15	YPL127C/HHO1	Late G_1	22	YLR382C/NAM2	Late G_1	28	YMR190C/SGS1	S
15	YLL021w/SPA2	Late G_1	22	YJL196C/	Late G_1	28	YIR017C/MET28	S
15	YBL063w/KIP1	S	22	YBR073w/RDH54	Late G_1	28	YHR086w/NAM8	S
15	YDR113c/PDS1	S	22	YKL101W/HSL1	Late G_1	28	YJR137C/	S
15	YDR356w/NUF1	S	22	YKL165C/	Late G_1			

Table 1: Genes from Table 1 of [13], clustered using PAAC, where $k = 28$.