SIMULTANEOUSLY SEGMENTING MULTIPLE GENE EXPRESSION TIME COURSES BY ANALYZING CLUSTER DYNAMICS

SATISH TADEPALLI[1], NAREN RAMAKRISHNAN [1], LAYNE T. WATSON[1], BHUBANESHWAR MISHRA[2], AND RICHARD F. HELM[3]

[1]*Department of Computer Science, Virginia Tech, Blacksburg, VA 24061*
[2]*Courant Institute of Mathematical Sciences, New York University, NY 10012*
[3]*Department of Biochemistry, Virginia Tech, Blacksburg, VA 24061*

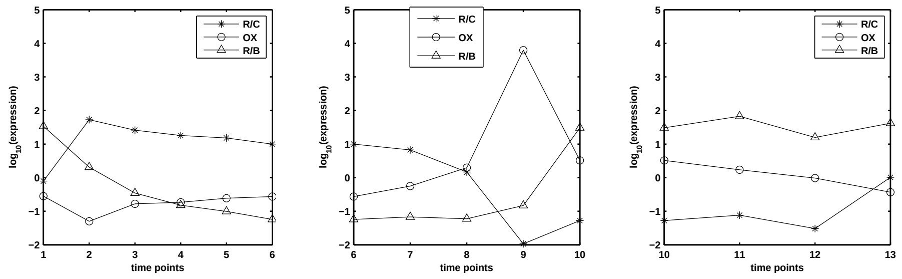
We present a new approach to segmenting multiple time series by analyzing the dynamics of cluster formation and re-arrangement around putative segment boundaries. This approach finds application in distilling large numbers of gene expression profiles into temporal relationships underlying biological processes. By directly minimizing information-theoretic measures of segmentation quality derived from Kullback-Leibler (KL) divergences, our formulation reveals clusters of genes along with a segmentation such that clusters show concerted behavior within segments but exhibit significant re-grouping across segmentation boundaries. The results of the segmentation algorithm can be summarized as Gantt charts revealing temporal dependencies in the ordering of key biological processes. Applications to the yeast metabolic cycle and the yeast cell cycle are described.

1. Introduction

Time course analysis has become an important tool for the study of developmental, disease progression, and cyclical biological processes, e.g., the cell cycle [8], metabolic cycle [9], and even entire life cycles. The growing affordability of microarray screens has fostered the generation of many time series datasets. Recent research efforts have considered using static measurements to “fill in the gaps” in the time series data [7], quantifying timing differences in gene expression [11], and reconstructing regulatory relationships [6].

One of the attractions of time series analysis is its promise to reveal temporal relationships underlying biological processes: which process occurs before what, what are the “checkpoints” that must be satisfied (and when), and whether there can be alternative pathways of time series progression. Although such analysis can be conducted by tracking individual genes whose function is known, we desire to automatically mine, in an unsupervised manner, temporal relationships involving *groups* of genes, which are not *a priori* defined. In particular, we desire to identify both segments of the time course where groups show concerted behavior and boundaries between segments where there is significant “re-grouping” of genes. We cast this problem as a form of time series segmentation where the segmentation criterion is driven by measures over cluster dynamics.

It is important to contrast our goals with prior work. Typical works on time series segmentation [3] are focused on segmenting a single time series whereas we are focused on simultaneously segmenting multiple time series. Typical works on segmentation view it as a problem of clustering time points with the constraint that data samples in a cluster must belong to successive time points, whereas we have the twin goals of clustering time points as well as clustering the genes. Typical works on segmentation are focused on homogeneity assumptions within a segment whereas we explicitly model each segment as a heterogeneous mix of multiple clusters which can themselves be redefined across segments. Our work is hence directly targeted to mining datasets involving thousands of genes where there are complex inter-relationships and re-organizations underlying the dataset.



	R/C	Ox	R/B	others
W1-C1	920	102	72	100
W1-C2	80	427	377	309
W1-C3	102	374	403	336

	R/C	Ox	R/B	others
W2-C1	421	60	370	328
W2-C2	200	795	150	106
W2-C3	481	48	332	311

	R/C	Ox	R/B	others
W3-C1	509	337	34	285
W3-C2	422	458	20	280
W3-C3	191	108	778	180

	W2-C1	W2-C2	W2-C3
W1-C1	421	401	372
W1-C2	380	448	365
W1-C3	378	402	435

	W3-C1	W3-C2	W3-C3
W2-C1	380	355	428
W2-C2	415	435	410
W2-C3	368	390	421

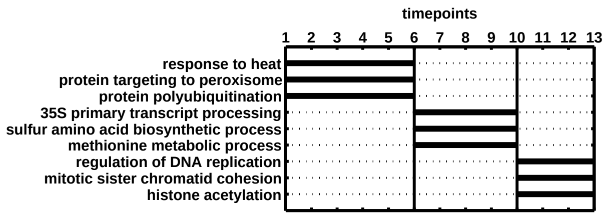


Figure 1. Preview of results: the yeast metabolic cycle dataset (top row) involves the staged coordination of a reductive, charging phase (time points [1–6]), followed by oxidative metabolism ([6–10]), followed by reductive metabolism ([10–13]). Contingency tables capture the concerted grouping of genes within segments (second row) as well as the re-groupings between segments (third row). Observe that the contingency tables in the second row involve significant enrichments whereas the tables in the third row approximate a uniform distribution. Gantt chart views (bottom row) depict the temporal coordination of biological processes underlying the dataset. Only some of the enriched functions are displayed, for lack of space.

As an example, consider the yeast metabolic cycle (YMC), using the dataset of Tu *et al.* [9]. The YMC is a carefully coordinated mechanism between a reductive charging (**R/C**) phase involving degradation and ubiquination activities, followed by oxidative metabolism (**Ox**), where oxygen is used up to generate adenosine triphosphates (ATPs), culminating in reductive metabolism (**R/B**) characterized by decrease in oxygen uptake and emphasis on DNA replication and cell division. Different genes are central to each of these phases. Tu *et al.* analyzed this time course by tracking 'sentinel' genes showing periodic behavior and arrived at a segmentation into nine intervals, corresponding to three complete instantiations of the YMC. One of these cycles, spanning 3 segments and 13 time points, is depicted in Fig. 1 (top). We analyzed 3602 gene expression profiles over a 15 hour period using our segmentation algorithm and it identified the same segmentation. Further, the clusters in each of our segments are enriched in the corresponding groups of genes, and demonstrate significant re-grouping across segments, as shown in Fig. 1 (middle two rows). These time-bounded enrichments can be summarized using a Gantt chart, as shown in Fig. 1 (bottom), which identifies biological processes as they become active or inactive in different segments. We reiterate that the time point boundaries, the groups of genes important in each segment, and the functions enriched in them, are inferred automatically. No explicit modeling of periodicity or other prior biological knowledge has been imparted to the segmentation algorithm.

2. Problem Formulation

We are given multiple vectors of measurements $\mathcal{G} = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_\nu\}$, where each $\mathbf{g}_i$ is a time series over $\mathcal{T} = \{t_1, t_2, \ldots, t_l\}$. The components of $\mathbf{g}_i$ can refer to absolute/relative gene expression values or some other processed version of the same, such as mean-centered and log-transformed values, or even coefficients in a principal component decomposition of $\mathcal{G}$. The problem of segmentation is to express $\mathcal{T}$ as a sequence of segments or windows: $\{w_{t_1}^{t_a}, w_{t_a}^{t_b}, \ldots, w_{t_k}^{t_T}\}$ where each window $w_{t_s}^{t_e}$, $t_s \le t_e$, is a set of consecutive time points beginning at (and inclusive of) time point t_s and ending at (and inclusive of) time point t_e. Note that adjacent windows have one time point overlap since we view partitions of the time course in terms of intervals rather than individual time points.

We first describe a way to evaluate a given segmentation before presenting an algorithm for identifying segmentations. We begin by studying the case of just two adjacent windows: $w_{t_a}^{t_b}$ and $w_{t_b}^{t_c}$. Given two clusterings of genes, one for each of the windows, our evaluation criterion requires that these two sets of clusters are highly dissimilar, i.e., genes clustered together in some cluster of $w_{t_a}^{t_b}$ move out of their clusters and are clustered together with different genes in $w_{t_b}^{t_c}$. For instance, given a dataset with 18 genes and 3 clusters in either window, the evaluation criterion prefers contingency table (a) below over tables (b) and (c). Here the rows refer to clusters of $w_{t_a}^{t_b}$ and the columns refer to clusters of $w_{t_b}^{t_c}$. We achieve this by enforcing that the (projected) row-wise and column-wise distributions from the contingency table resemble a uniform distribution.

$$\boxed{\begin{matrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{matrix}} \qquad \boxed{\begin{matrix} 6 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 6 \end{matrix}} \qquad \boxed{\begin{matrix} 0 & 6 & 0 \\ 6 & 0 & 0 \\ 0 & 0 & 6 \end{matrix}}$$

(a) (b) (c)

Formally, given two windows $w_{t_a}^{t_b}$ and $w_{t_b}^{t_c}$, which have been clustered into r and c clusters (respectively), we define the $r \times c$ contingency table over the clusterings. Entry n_{ij} in the $(i, j)^{th}$ cell of the table represents the overlap between the genes clustered together in cluster i of $w_{t_a}^{t_b}$ and in cluster j of $w_{t_b}^{t_c}$. The sizes of the clusters in $w_{t_a}^{t_b}$ are given by the column-wise sums across each row: $n_{i\cdot} = \sum_j n_{ij}$, while the sizes of clusters in $w_{t_b}^{t_c}$ are given by row-wise sums across each column: $n_{\cdot j} = \sum_i n_{ij}$. Using these, we define $(r + c)$ probability distributions, one for each row and one for each column. The distribution corresponding to row i, $\mathbf{R}_i$, takes values from the column indices, i.e., $1 \ldots c$, with value j ($1 \leq j \leq c$) occurring with probability $\frac{n_{ij}}{n_{i\cdot}}$. Similarly, the column distribution for column j, $\mathbf{C}_j$, takes values from the row indices, i.e., $1 \ldots r$, with value i ($1 \leq i \leq r$) occurring with probability $\frac{n_{ij}}{n_{\cdot j}}$. We capture the deviation of these row-wise and column-wise distributions w.r.t. the uniform distribution as[a]:

$$\mathcal{F} = \frac{1}{r}\sum_{i=1}^{r} D_{KL}(\mathbf{R}_i||U(\frac{1}{c})) + \frac{1}{c}\sum_{j=1}^{c} D_{KL}(\mathbf{C}_j||U(\frac{1}{r})) \tag{1}$$

where $D_{KL}(\cdot||\cdot)$ is the Kullback-Leibler (KL) divergence between two probability distributions:

$$D_{KL}(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

with the limits $0 \log \frac{0}{q(x)} \rightarrow 0$, $p(x) \log \frac{p(x)}{0} \rightarrow \infty$ implied, and $U(\cdot)$ denotes the uniform distribution whose argument is the probability of any outcome. The optimization problem is then to minimize $\mathcal{F}$.

Observe that the combinations of the r row-wise KL-divergences and c column-wise KL-divergences are averaged to form $\mathcal{F}$. This is to mitigate the effect of lopsided contingency tables ($r \gg c$ or $c \gg r$) wherein it is possible to optimize $\mathcal{F}$ by focusing on the "longer" dimension without really ensuring that the other dimension's projections are close to uniform. Finally, note that Eq. 1 can be readily extended to the case where we have more than two segments[b].

Minimizing $\mathcal{F}$ will yield row-wise and column-wise distribution estimates that are close to the respective uniform distributions and, hence, result in independent clusterings across the neighboring windows. However, there are special cases when $\mathcal{F}$ could have a minimum, but the resulting clustering doesn't quite meet our intuition of independent clusterings and hence, to avoid these cases, we regularize $\mathcal{F}$ with additional terms as described later.

3. Clustering across windows

We now turn our attention to the clustering algorithm that must balance two conflicting criteria: the clusters across neighboring windows must be independent but the clusters must exhibit concerted behavior within a window. In typical clustering algorithms, each cluster has a prototype and the data vectors are assigned to the nearest cluster based on some distance measure from these prototypes. The prototypes are iteratively improved to find the best possible clusters.

Again, we develop our notation for two adjacent windows and the extension to greater numbers of windows is straightforward. Given a gene vector $\mathbf{g}_k$, let its projection onto the 'left' window $w_{t_a}^{t_b}$ be referred to as $\mathbf{x}_k$, and its projection onto the 'right' window $w_{t_b}^{t_c}$ be referred to as $\mathbf{y}_k$.

[a]An alternate formulation is to cast the uniform distribution requirement over all the contingency table entries rather than over row and column marginals separately as done here. However, our approach is more intuitive since the converse problem of finding highly dependent clusters then reduces to simply maximizing Eq. 1. This converse problem is known as associative clustering and has been previous studied [2] using measures such as the Bayes factor.

[b]The objective function defined in Eq. 1 has connections to the principle of minimum discrimination information (MDI) introduced by Kullback for the analysis of contingency tables [4]. The MDI principle states that if q is the assumed or true distribution, the estimated distribution p must be chosen such that $D_{KL}(p||q)$ is minimized. In our case, q is the uniform distribution desired and p is the distribution estimated from observed data.

Recall that sets of such projections are clustered separately such that the clusters are maximally dissimilar. Let r and c be the number of clusters for $\mathbf{x}$ and $\mathbf{y}$ vectors, which results in a $r \times c$ contingency table. Let $\mathbf{m}_i^{(x)}$ be the prototype vector for the ith cluster of the $\mathbf{x}$ vectors. The assignment of a data vector to the clusters is given by the probability distribution $\mathbf{V}(\mathbf{x}_k) = \{V_i(\mathbf{x}_k)\}$, where $\sum_{i=1}^{r} V_i(\mathbf{x}_k) = 1$. The probabilities $V_i(\mathbf{x}_k)$ are the cluster membership indicator variables, i.e., the probability that data sample k is assigned to cluster i. Similar cluster prototypes $\mathbf{m}_j^{(y)}$, distributions $\mathbf{V}(\mathbf{y}_k)$, and cluster indicator variables $V_j(\mathbf{y}_k)$ are defined for $\mathbf{y}$ vectors as well. Then contingency table counts can be calculated as $n_{ij} = \sum_k V_i(\mathbf{x}_k) V_j(\mathbf{y}_k)$. In *hard clustering* algorithms, like the traditional k-means, each data sample is assigned to the nearest cluster with a probability of 1. However, calculating n_{ij} using hard memberships renders the function $\mathcal{F}$ in Eq 1 non-differentiable at certain points, as a result of which we cannot leverage classical numerical optimization algorithms to minimize $\mathcal{F}$. To avoid this problem, cluster indicator variables are typically parameterized as a smooth function that is continuously differentiable and which assigns a non-zero cluster membership probability for each data sample, i.e $V_i(\mathbf{x}_k), V_i(\mathbf{y}_k) \in (0, 1)$. We present a novel smoothing approximation that tracks the cluster indicator variables with high accuracy. First, we define

$$\gamma_{(i,i')}(\mathbf{x}_k) = \frac{||\mathbf{x}_k - \mathbf{m}_{i'}^{(x)}||^2 - ||\mathbf{x}_k - \mathbf{m}_i^{(x)}||^2}{D}, 1 \le i, i' \le r$$

where D is the point-set diameter

$$D = \max_{k,k'} ||\mathbf{x}_k - \mathbf{x}_{k'}||^2, 1 \le k, k' \le \nu$$

A well known approximation to $\min_{i'} \gamma_{(i,i')}(\mathbf{x}_k)$ is the Kreisselmeier-Steinhauser (KS) envelope function [10] given by

$$KS_i(\mathbf{x}_k) = \frac{-1}{\rho} \ln \Big[\sum_{i'=1}^{r} \exp(-\rho\, \gamma_{(i,i')}(\mathbf{x}_k)) \Big]$$

where $\rho \gg 0$. The KS function is a smooth function which is differentiable to any degree. Using this the cluster memberships are redefined as:

$$V_i(\mathbf{x}_k) = Z(\mathbf{x})^{-1} \exp \Big[\rho\, KS_i(\mathbf{x}_k) \Big]$$

where $Z(\mathbf{x})$ is a normalizing function such that $\sum_i^r V_i(\mathbf{x}_k) = 1$. The cluster memberships for the “right” window, $V_j(\mathbf{y}_k)$, are also smoothed similarly. If the data sample belongs to a cluster, the cluster membership probability is slightly less than 1 and for all the other clusters it is slightly greater than 0.

Minimizing the function $\mathcal{F}$ in Eq 1 should ideally yield clusters that are independent across windows and local within each window. However, using smooth cluster prototypes gives rise to an alternative minimum solution where each data sample is assigned with uniform probability to every cluster. Recall the contingency table example from Sec 2; each of the 18 samples can be assigned to the 3 row clusters (and 3 column clusters) with probability $[1/3, 1/3, 1/3]$ and the estimate of count matrix from these soft counts would still be uniform in each cell

($\sum_k V_i(\mathbf{x}_k)V_j(\mathbf{y}_k) = 2$). To avoid degenerate solutions such as these, we require maximum deviation of individual data vector probabilities ($V_i(\mathbf{x}_k)$ and $V_j(\mathbf{y}_k)$) from the uniform distribution over the number of clusters. This leads to the regularized objective function:

$$\begin{aligned}\mathcal{F} =&\frac{\lambda}{r}\sum_{i=1}^{r} D_{KL}(\mathbf{R}_i||U(\frac{1}{c})) + \frac{\lambda}{c}\sum_{j=1}^{c} D_{KL}(\mathbf{C}_j||U(\frac{1}{r}))\\ &-\frac{1}{\nu}\sum_{k=1}^{\nu} D_{KL}(\mathbf{V}(\mathbf{x}_k)||U(\frac{1}{r})) - \frac{1}{\nu}\sum_{k=1}^{\nu} D_{KL}(\mathbf{V}(\mathbf{y}_k)||U(\frac{1}{c}))\end{aligned} \tag{2}$$

where λ is the weight, set to a value greater than 1, to give more emphasis to minimizing the row and column wise distributions. This also enforces equal cluster sizes. Substituting the KL-divergences in Eq. 2:

$$\begin{aligned}\mathcal{F} =&\frac{\lambda}{r}\sum_{i=1}^{r}\sum_{j=1}^{c}\frac{\sum_{k=1}^{\nu}V_i(\mathbf{x}_k)V_j(\mathbf{y}_k)}{\sum_{k=1}^{\nu}V_i(\mathbf{x}_k)}\log_2\Big(\frac{\sum_{k=1}^{\nu}V_i(\mathbf{x}_k)V_j(\mathbf{y}_k)}{\sum_{k=1}^{\nu}V_i(\mathbf{x}_k)}\Big/\frac{1}{c}\Big)\\ &+\frac{\lambda}{c}\sum_{j=1}^{c}\sum_{i=1}^{r}\frac{\sum_{k=1}^{\nu}V_i(\mathbf{x}_k)V_j(\mathbf{y}_k)}{\sum_{k=1}^{\nu}V_i(\mathbf{y}_k)}\log_2\Big(\frac{\sum_{k=1}^{\nu}V_i(\mathbf{x}_k)V_j(\mathbf{y}_k)}{\sum_{k=1}^{\nu}V_i(\mathbf{y}_k)}\Big/\frac{1}{r}\Big)\\ &-\frac{1}{\nu}\sum_{k=1}^{\nu}\sum_{i=1}^{r}V_i(\mathbf{x}_k)\log_2\Big(\sum_k V_i(\mathbf{x}_k)\Big/\frac{1}{r}\Big) - \frac{1}{\nu}\sum_{k=1}^{\nu}\sum_{j=1}^{c}V_j(\mathbf{y}_k)\log_2\Big(\sum_k V_j(\mathbf{y}_k)\Big/\frac{1}{c}\Big)\end{aligned}$$

The derivative of this function w.r.t the prototypes $\mathbf{m}_i^{(x)}$ is given by:

$$\begin{aligned}\nabla_{\mathbf{m}_i^{(x)}}\mathcal{F} = \frac{1}{\ln(2)}\sum_{i'=1}^{r}\sum_{k=1}^{\nu}\Bigg(&\frac{\lambda}{r}\Bigg\{\sum_{j=1}^{c}\Big[1+\ln\Big(\frac{\sum_{k'=1}^{\nu}V_{i'}(\mathbf{x}_{k'})V_j(\mathbf{y}_{k'})}{\sum_{k'=1}^{\nu}V_{i'}(\mathbf{x}_{k'})}\Big/\frac{1}{c}\Big)\Big]\\ &\Big[\frac{V_j(\mathbf{y}_k)}{\sum_{k'=1}^{\nu}V_{i'}(\mathbf{x}_{k'})} - \frac{\sum_{k'=1}^{\nu}V_{i'}(\mathbf{x}_{k'})V_j(\mathbf{y}_{k'})}{\sum_{k'=1}^{\nu}(V_{i'}(\mathbf{x}_{k'}))^2}\Big]\Bigg\}\\ &-\frac{\lambda}{c}\Bigg\{\sum_{j=1}^{c}\Big[1+\ln\Big(\frac{\sum_{k'=1}^{\nu}V_{i'}(\mathbf{x}_{k'})V_j(\mathbf{y}_{k'})}{\sum_{k'=1}^{\nu}V_j(\mathbf{y}_{k'})}\Big/\frac{1}{r}\Big)\Big]\Big[\frac{V_j(\mathbf{y}_k)}{\sum_{k'=1}^{\nu}V_j(\mathbf{y}_{k'})}\Big]\Bigg\}\\ &-\frac{1}{\nu}\Big[1+\ln(V_i(\mathbf{x}_k)\Big/\frac{1}{r})\Big]\Bigg)\times\nabla_{\mathbf{m}_i^{(x)}}V_{i'}(\mathbf{x}_k)\end{aligned}$$

where

$$\begin{aligned}\nabla_{\mathbf{m}_i^{(x)}}V_{i'}(\mathbf{x}_k) = \frac{2\rho(\mathbf{x}_k-\mathbf{m}_i^{(x)})}{D}\Bigg(&\delta_{i',i}V_i(\mathbf{x}_k) - \exp(-\rho\gamma_{(i',i)}(\mathbf{x}_k))(V_{i'}(\mathbf{x}_k))^2 Z(\mathbf{x})\\ &+\sum_{i''=1}^{r}\Big[(V_{i''}(\mathbf{x}_k))^2\exp(-\rho\gamma_{(i'',i)}(\mathbf{x}_k))V_{i'}(\mathbf{x}_k)Z(x)\Big] - V_i(\mathbf{x}_k)V_{i'}(\mathbf{x}_k)\Bigg)\end{aligned}$$

Here $\delta_{i',i}$ is the Kronecker's delta. The index variables are as follows: i, i', and i'' over the clusters in $\mathbf{x}$ vectors, j over the clusters in the $\mathbf{y}$ vectors, and k and k' over the data vectors. The derivatives w.r.t. prototypes $\mathbf{m}_j^{(y)}$ are calculated analogously.

4. Segmentation Algorithm

Let $\mathcal{T} = \{t_1, t_2, \ldots, t_l\}$ be the given time series data set, and l_{min} and l_{max} be the minimum and maximum window lengths respectively. For each time point $t_a \in T$, we define the set of windows starting from t_a as $S_{t_a} = \{w_{t_a}^{t_b} | l_{min} \leq t_b - t_a + 1 \leq l_{max}\}$. Given a window $w_{t_a}^{t_b}$, the choices for the next window are given by S_{t_b}, the set of windows starting form t_b. We experimented with both dynamic programming and greedy algorithms but the results were not qualitatively different and we present the greedy algorithm for ease of exposition. We cluster the data in $w_{t_a}^{t_b}$ simultaneously with each of the windows $w_{t_b}^{t_c} \in S_{t_b}$ using the objective function in Eq 2 and choose the window $w_{t_b}^{t_c}$ that has the minimum value for the objective function. Optimization of the objective function for clustering is performed using the conjugate gradient procedure in the LANCELOT FORTRAN package for optimization. The initial cluster prototypes are set using individual k-means clusters in each window. The conjugate gradient procedure iteratively improves these initial prototypes till a local minimum of the objective function is attained. Since local optimization procedures are sensitive to initialization, we perform 100 random restarts of the procedure (each time with different k-means prototypes found in individual windows) and choose the best (minimum) of the optimized solutions as the score for the next window choice. The window corresponding to the minimum of these best choices is selected as the next window. This process is continued till we reach the end of the time course.

5. Experiments

Datasets: Our experimental datasets constitute gene expression measurements spanning the yeast metabolic cycle (YMC) and the yeast cell cycle (YCC). As stated earlier, the YMC dataset [9] consists of 36 time points collected over 3 continuous cycles. The original dataset consists of 6555 unique genes from the *S. cerevisiae* genome. We first eliminated those genes that do not have an annotation in any GO biological process category (revision 4.205 of GO released on 14 March 2007), resulting in a universal set of 3602 genes. The gene expression values were log transformed (base 10) and normalized such that the mean expression of each gene across all time points is zero. To segment this dataset we experimented with the number of clusters in each segment ranging from 3 to 15, a minimum window length l_{min} of 4 and maximum window length l_{max} of 7, and $\lambda = 1.4$. The λ value was adjusted to give approximately equal sized clusters with good intra-cluster similarities. The YCC was taken from the well known experiment of Spellman et al. [8]. Due to lack of space, we describe our analysis on only the $\alpha-$factor time course from [8], which has 6076 genes with 18 time points over approximately 2 cycles. We considered the genes with no missing values and mean centered each gene's expression across all time points to zero. From this data, we removed the genes that do not have an annotation in GO biological process category which resulted in a final set of 2196 genes. To segment this dataset, again we ranged from 3 to 15 clusters in each window, a minimum window length l_{min} of 3 and maximum window length l_{max} of 5, and $\lambda = 1.25$ (adjusted as before).

Evaluation metrics: We evaluate our clusterings and segmentations in five ways: cluster stability, cluster reproducibility, functional enrichment, segmentation quality, and segmentation sensitivity. We assess **cluster stability** using a bootstrap procedure to determine significance of genes brought together. Recall that each window except the first and last windows has two sets of clusters, one set independent with respect to the previous window and the other independent with respect to the next window. We are interested in the genes that are significantly clustered together in these two sets of clusters, as they represent the genes that are specific to the window under consideration. We calculate a contingency table between these two clusterings for each window (excluding the first and the last window). Each cell in the contingency table represents the number of genes that are together across the two independent sets of clusters. We randomly sample 1000 sets of clusters and compute the contingency tables w.r.t. them. The distribution of counts in each cell of the table was tabulated and found to be approximately normal (Shapiro-Wilk normality test with $p = 0.05$). We now evaluate each cell of the actual contingency table with respect to the corresponding random distribution and retain only those cells that have more genes than that observed at random with $p < 0.05$ (Bonferroni corrected with the number of cross clusters to account for multiple hypothesis testing). To ensure **reproducibility** of clusters, we retain only those genes in each significant cell of the contingency table that are together in more than 150 of the 200 clusterings (conducted with different initializations). For the first and last windows, which have only 100 randomly initialized clusterings, we retain those genes that are clustered together in more than 75 of the 100 clusterings. After the above two steps, we perform **functional enrichment** using the GO biological process ontology (since we are tracking biological processes) over the selected clusters of genes. A hypergeometric $p-$value is calculated for each GO biological process term, and an appropriate cutoff is chosen using a false discovery rate (FDR) $q-$ level of 0.01.

The **segmentation quality** is calculated as a partition distance [5] between the “true” segmentation (from the literature of the YMC and YCC) to the segmentations computed by our algorithm. Since this measure requires partitions with no overlap between blocks, we view each segment/window as a set of unit-sized intervals, rather than time points. Thus, the window w_1^4 has carinality 3 and two elements in common with z_2^4. Given two segmentations S_1 and S_2, the partition distance is given by:

$$P_D = -\sum_{w_{t_a}^{t_b} \in S_1} \sum_{z_{t_a}^{t_b} \in S_2} |w_{t_a}^{t_b} \cap z_{t_a}^{t_b}| \, \log_2 \frac{|w_{t_a}^{t_b} \cap z_{t_a}^{t_b}|}{|w_{t_a}^{t_b}|} - \sum_{z_{t_a}^{t_b} \in S_2} \sum_{w_{t_a}^{t_b} \in S_1} |w_{t_a}^{t_b} \cap z_{t_a}^{t_b}| \, \log_2 \frac{|w_{t_a}^{t_b} \cap z_{t_a}^{t_b}|}{|z_{t_a}^{t_b}|}$$

The **segementation sensitivity** to variations in the number of clusters is calculated as the ratio of average KL-divergences between the segments to the maximum possible KL divergence. Suppose we have $|S|$ windows in a given segmentation $S = \{w_{t_1}^{t_a}, w_{t_a}^{t_b}, \ldots, w_{t_j}^{t_k}, w_{t_k}^{t_l}\}$ with c clusters in each window. Let $\mathcal{F}_{max}$ be the objective function value for the maximally similar clustering (the $c \times c$ diagonal contingency table as in the example in section 2). Then the measure we compute is

$$KL_{avg} = \frac{1}{|S|-1} \left[\frac{\mathcal{F}_{\{w_{t_1}^{t_a}, w_{t_a}^{t_b}\}}}{\mathcal{F}_{max}} + \frac{\mathcal{F}_{\{w_{t_b}^{t_c}, w_{t_c}^{t_d}\}}}{\mathcal{F}_{max}} + \ldots + \frac{\mathcal{F}_{\{w_{t_j}^{t_k}, w_{t_k}^{t_l}\}}}{\mathcal{F}_{max}} \right]$$

where $\mathcal{F}_{\{w_{t_a}^{t_b}, w_{t_b}^{t_c}\}}$ is the objective function value obtained during clustering the pair of adjacent windows $\{w_{t_a}^{t_b}, w_{t_b}^{t_c}\}$. Lower values of this ratio indicate that the segmentation captures maximal independence between adjacent segments while higher values indicate the clusters obtained are more similar in adjacent segments.

Results

YMC: The segmentation generated for the minimum number (3) of clusters is: [1-6], [6-10], [10-13], [13-16], [16-21], [21-26], [26-31], [31-36], which correspond to alternating R/B, R/C, and Ox phases. The GO categories enriched ($p < 1e-7$) in one cycle of this dataset has already been depicted in Fig. 1. This segmentation is stable upto a setting of 8 clusters after which it begins to deviate from the "true" segmentation (discussed further below).

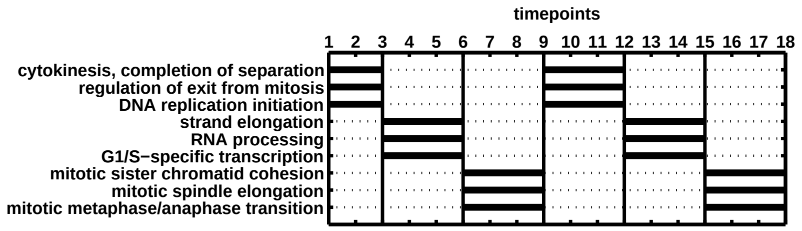


Figure 2. Gantt chart resulting from segmentation of Spellman et al. dataset. To preserve space, only some of the enriched GO biological process terms are shown.

YCC: The segmentation (Fig. 2) generated for YCC—[1-3], [3-6], [6-9], [9-12], [12-15], [15-18]—is also periodic with the stages approximately corresponding to alternating M/G1, $\{$G1,S$\}$, $\{$G2,M$\}$ phases. Note that each phase is of very short length in this experiment as compared to YMC: the phases M/G1, G1, S each last for approximately 2 time points, while the G2 phase lasts only for one time point. Due to our minimum window length of 3 (set so that we recover significant clusterings and re-groupings), we cannot resolve these short-lived phases. A possible approach is to use continuous-representations such as spline fits to gain greater resolution of data sampling. Nevertheless, the key events occurring in these segments are retrieved with high specificity ($p < 1e-7$) as shown in Fig. 2.

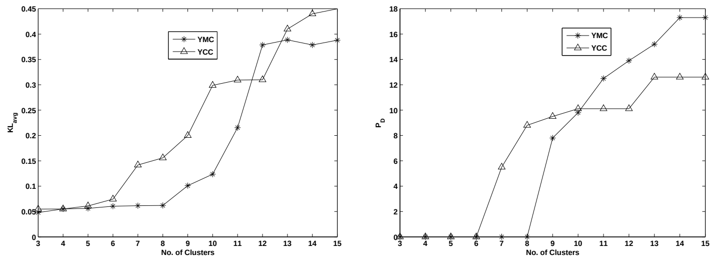


Figure 3. Tracking (left) segmentation sensitivity and (right) segmentation quality with number of clusters.

The effect of the number of clusters on segmentation characteristics is studied in Fig. 3. In Fig. 3 (left), we see that as the number of clusters increases, it is increasingly difficult to obtain independent clusterings and hence, for higher values of the number of clusters, the segmentation problem actually resembles associative clustering (observe that this curve tends toward a KL_{avg} value of 0.5). Fig. 3 (right) tracks the segmentation quality, and shows that the correct segmentation is recovered for many settings in the lower range for number of clusters but, as the number of clusters increases, while there are fewer choices for the segmentation algorithm, all of which considerably deviate from the true segmentation. Nevertheless, comparing the two plots, we see that KL_{avg} tracks the segmentation quality P_D well and hence can be a useful surrogate for determining the “right” number of clusters.

Biological significance: One of the applications of our methods is to decode temporal relationships between biological processes. Since cell division processes are enriched in both YCC and YMC, we superimposed those segments of the two Gantt charts (from Fig. 1 and Fig. 2) and noticed that the oxidative metabolism phase of YMC typically precedes the transition from G1 to S in the YCC. Such a connection has been investigated in the literature by Futcher [1] but through the use of a custom experiment observing metabolism during the course of the cell cycle. As the budding yeast grows in size, it is hypothesized that the accummulation of carbohydrates is one of the ways in which it gets past the size checkpoint controlling entry into S. This finding demonstrates the potential for knowledge discovery by mining Gantt charts using our methods.

6. Discussion

We have presented a novel approach to simultaneously segment multiple time course data using clusters and movement of data points across clusters as a driving criterion for optimization. It is important to emphasize that our objective criteria are naturally posed over contingency tables, rather than over other indirect measures of cluster movement. Our approach recovers the periodicity of the underlying biology even though the algorithm is not steered toward modeling it. The Gantt charts resulting from our analysis can serve the basis for reconstructing temporal dependencies underlying biological processes. In future work, we aim to develop richer models of cluster re-organization, e.g., dynamic revisions in the number of clusters and split-and-merge behaviors of clusters, leading to inference of complete temporal logic models.

References

1. B. Futcher. Metabolic Cycle, Cell Cycle, and the Finishing Kick to Start. Genome Biology, Vol. 7:107, 2006.
2. S. Kaski, J. Nikkilä, J. Sinkkonen, L. Lahti, J.E.A. Knuuttila, and C. Roos. Associative clustering for exploring dependencies between functional genomics data sets. IEEE/ACM TCBB, 2(3):203–216, 2005.
3. E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting time series: A survey and novel approach. In Data Mining in Time Series Databases. World Scientific Publishing Company, 2003.
4. S. Kullback and D.V. Gokhale. The Information in Contingency Tables. Marcel Dekker Inc., 1978.
5. R.L. De Mántaras. A Distance-Based Attribute Selection Measure for Decision Tree Induction. Machine Learning, 6(1):81–92, 1991.
6. Y. Shi, T. Mitchell, and Z. Bar-Joseph. Inferring Pairwise Regulatory Relationships from Multiple Time Series Datasets. Bioinformatics, 23(6):755–763, 2007.
7. I. Simon, Z. Siegfried, J. Ernst, and Z. Bar-Joseph. Combined Static and Dynamic Analysis for Determining the Quality of Time-series Expression Profiles. Nature Biotechnology, 23:1503 – 1508, 2005.
8. P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein, and B. Futcher. Comprehensive Identification of Cell Cycle-Regulated Genes of the Yeast Saccharomyces Cerevisiae by Microarray Hybridization. Molecular Biology of the Cell, 9(12):3273–3297, 1998.
9. B.P. Tu, A. Kudlicki, M. Rowicka, and S.L. McKnight. Logic of the Yeast Metabolic Cycle: Temporal Compartmentalization of Cellular Processes. Science, 310(5751):1152–1158, 2005.
10. G.A. Wrenn. An Indirect Method for Numerical Optimization using the Kreisselmeier-Steinhauser Function. NASA Contractor Report 4220, March 1989.
11. T. Yoneya and H. Mamitsuka. A Hidden Markov Model-based Approach for identifying Timing Differences in Gene Expression under Different Experimental Factors. Bioinformatics, 2007.