BIOINFORMATICS

Vol. 25 ISMB 2009, pages i119–i127
doi:10.1093/bioinformatics/btp206

Clustered alignments of gene-expression time series data

Adam A. Smith[1,2,*], Aaron Vollrath[3], Christopher A. Bradfield[3] and Mark Craven[1,2,*]

[1]Department of Biostatistics & Medical Informatics, [2]Department of Computer Sciences and [3]Department of Oncology, University of Wisconsin, Madison, USA

ABSTRACT

Motivation: Characterizing and comparing temporal gene-expression responses is an important computational task for answering a variety of questions in biological studies. Algorithms for aligning time series represent a valuable approach for such analyses. However, previous approaches to aligning gene-expression time series have assumed that all genes should share the same alignment. Our work is motivated by the need for methods that identify sets of genes that differ in similar ways between two time series, even when their expression profiles are quite different.

Results: We present a novel algorithm that calculates *clustered alignments*; the method finds clusters of genes such that the genes within a cluster share a common alignment, but each cluster is aligned independently of the others. We also present an efficient new segment-based alignment algorithm for time series called SCOW (shorting correlation-optimized warping). We evaluate our methods by assessing the accuracy of alignments computed with sparse time series from a toxicogenomics dataset. The results of our evaluation indicate that our clustered alignment approach and SCOW provide more accurate alignments than previous approaches. Additionally, we apply our clustered alignment approach to characterize the effects of a conditional Mop3 knockout in mouse liver.

Availability: Source code is available at http://www.biostat.wisc.edu/~aasmith/catcode.

Contact: aasmith@cs.wisc.edu

1 INTRODUCTION

Characterizing and comparing temporal gene-expression responses is an important computational task for answering a variety of questions in biological studies. In previous work (Smith and Craven, 2008; Smith *et al.*, 2008), we have introduced methods for answering similarity queries about gene-expression profiles after exposure to some chemical or treatment. These methods have been motivated by the task of quickly and accurately characterizing the potential toxicity of chemicals. A fundamental step in comparing two time series is with temporally align the series using a method such as *dynamic time warping* (Sakoe and Chiba, 1978; Sankoff and Kruskal, 1983). Previous approaches to aligning gene-expression time series have assumed that all genes should be aligned in lockstep with one another. In other words, these methods assume that the transformation that specifies how one series relates to another is the same for all genes. Here, we present a novel approach that finds clusters of genes such that the genes within a cluster share a common alignment, but each cluster is aligned independently of the others. Our method is similar to *k*-means clustering (Duda *et al.*, 2000) in that it alternates between assigning genes to clusters and recomputing the alignment for each cluster using the genes assigned to it. We also present a novel *multi-segment* alignment algorithm that computes more accurate alignments for sparse gene-expression time series than previous methods.

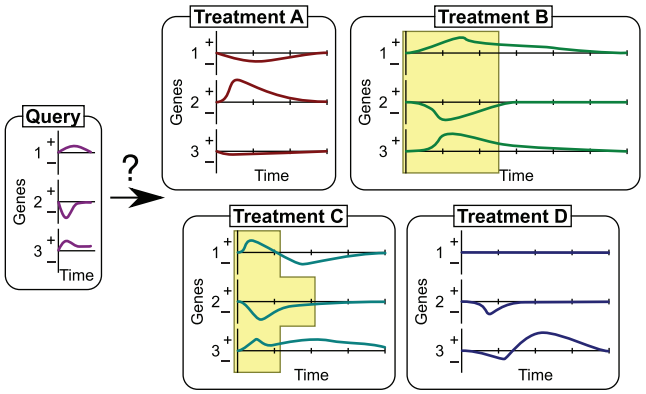

Fig. 1. The time-series similarity task. Given a gene-expression time series as a query, we want to find the time series in the database which are most similar to the query. Shaded areas represent strong matches to the given query. Notice that for both Treatments B and C, the best alignment to the query does not account for the entire extent of the treatments. Also notice that with Treatment B, all genes can be aligned together, whereas with Treatment C the second gene should be aligned separately.

One application for time-series alignment that we consider is the task of answering similarity queries as illustrated in Figure 1. Given an expression profile as a query, we want to identify treatments in a database that have expression profiles most similar to the query. When the query and/or some of the database treatments are time series, we assess similarity by determining the temporal correspondence between the query and treatments in the database. In our toxicogenomics application, we might be trying to determine if an uncharacterized chemical induces an expression response similar to any known toxicants. The figure shows a simple case in which our database consists of expression profiles from four different treatments, and each expression profile characterizes only three genes.

Figure 1 illustrates two important issues that arise in this task. Sometimes (as with Treatment B) all genes should be aligned (i.e. warped) together to find the best correspondence. But, it may also happen that some genes need to be warped separately from the others, as with Treatment C. A second issue is that often the best alignment does not account for the complete extent of both time series. Therefore, we want to allow a type of local alignment in which the end of one series is unaligned. We refer to this case as *shorting* the alignment. The two main contributions of this work are algorithms that are designed to address both of these issues when computing time-series alignments.

*To whom correspondence should be addressed.

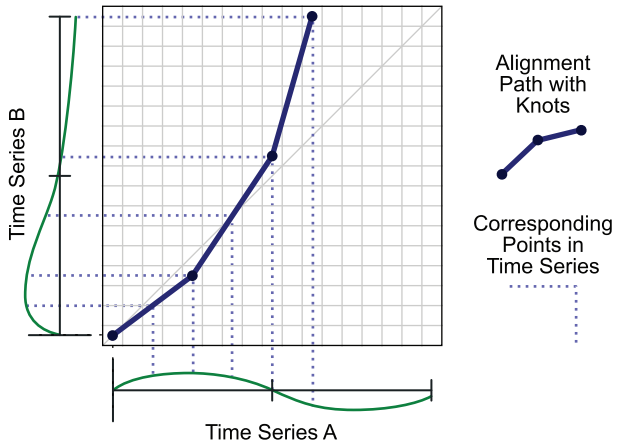

Fig. 2. Alignment space example. The multi-segment alignment path characterizes correspondences between two series, as shown by the dotted lines. Knots are the points of discontinuity in the path.

Figure 2 shows the alignment of two time series in *alignment space*, using a *multi-segment* alignment method. The alignment path determines which points in the two series are mapped to one another. For a given point in the path, the coordinate in the first time series directly below it and the coordinate in the second time series directly to its left correspond to one another. A multi-segment alignment can take into account that the nature of the relationship between the two series may vary in different segments. For example, it may be the case that the later part of the expression response occurs more slowly in one treatment than in a similar treatment. We refer to the points of discontinuity that define the segment boundaries as *knots*.

The alignment in Figure 2 also illustrates the concept of shorting. Here, Time Series A seems to have advanced more quickly than Time Series B, which has not started to increase at the end. An alignment path that represents shorting ends in the top row or the right column of the alignment space diagram, but not in the top-right cell. Note that we do not allow an alignment to short both series; all of one or the other must be mapped to some point in its mate.

In previous work (Smith *et al.*, 2008), we described a novel multi-segment alignment method and empirically demonstrated that it classifies and aligns our toxicogenomics data better than several competing methods, including dynamic time warping, several parametric methods (such as linear alignment) and another multi-segment method called correlation-optimized warping, or COW (Nielsen *et al.*, 1998). Parametric methods, which constrain the warping path to a simple functional form, often are not expressive enough to capture the most appropriate warping. In contrast, dynamic time warping can often be too expressive, finding high-scoring alignments of unrelated series. A multi-segment method provides a balance between these two methods.

The accuracy advantage of our previous multi-segment method over COW was slight. COW is a global alignment method that cannot short. On closer inspection, we found that our method discovered more accurate alignments in cases that required shorting, whereas COW dominated those trials that did not. Here, we present a modified version of COW that allows shorted alignments. We call the method SCOW, for shorting COW. Our algorithm for computing clustered alignments uses SCOW as its base alignment method.

Aach and Church (2001) were the first to apply the method of *dynamic time warping* (Sakoe and Chiba, 1978) to gene-expression profiles, and other groups have followed with this warping method (Criel and Tsiporkova, 2006; Liu and Müller, 2003) and others (Bar-Joseph *et al.*, 2003). Importantly, they have all done their warping on all genes together, whereas we compute clustered alignments. Also, our approach differs in that it compute multi-segment alignments and considers local alignments via shorting.

Other studies have investigated clustering gene-expression time series (Bar-Joseph *et al.*, 2003; Eisen *et al.*, 1998; Leng and Müller, 2006; Liu and Müller, 2003). The important differences between these approaches and ours are 2-fold: the goals of the clustering process and the notion of similarity used. Whereas these previous methods have focused on identifying clusters of genes that have similar expression profiles, our approach, in contrast, is focused on identifying clusters in which the genes have similar warpings. The genes in one of our clusters may have very different expression profiles, but they are similar in how they should be warped across the two time series being compared.

Listgarten *et al.* (2005) have developed a method for multiple alignment of time series data that has some similarities to our approach. Their method, however, computes a single alignment of multiple time series, whereas our method computes a clustered alignment of a pair of time series.

We are not the first group to develop algorithms for computing shorted alignments. Keogh (2003) devised a two-step shorting method that first finds the appropriate end points of an alignment before calculating a global alignment up to these points. Our approach to shorting is different in that the shorting decision is not decoupled from the computation of the alignment; the dynamic programming method considers shorted as well as non-shorted alignments.

2 METHODS

In this section, we detail two novel techniques that we have developed. The first is SCOW, which is a method for computing multi-segment alignments of two time series and assessing their similarity. The second is an algorithm which computes *clustered alignments* in which the genes within a cluster share a common alignment, but each cluster is aligned independently of the others.

2.1 SCOW

We start by describing COW (Nielsen *et al.*, 1998), which is a dynamic programming algorithm designed to find an optimal alignment between two series with multiple channels of information (such as genes). We then describe SCOW, which is our extension to COW.

COW was developed to align chromatography time-series data. Briefly, it aligns and scores two given time series based on their similarity. Here, we refer to the two series as q (for query series) and d (for database series). For each possible alignment, the series are partitioned into m segments, in which the i-th segments of the two series correspond to each other. The score of a given alignment is the sum of correlations between corresponding segments.

As shown in Figure 3A, COW searches for good segment boundaries in only a limited area of alignment space. The segments are assumed to be of constant length in q, and variable in d. The vector K contains the coordinates of the *knots* (segment endpoints) in q. These are usually evenly spaced. COW works by filling a zero-indexed matrix Γ, which is of dimensions $m+1$ by $|d|+1$. The element $\gamma_{k,x}$ contains the score of the best alignment of d from zero to x and q from zero to K_k (the k-th element of K) using k segments.

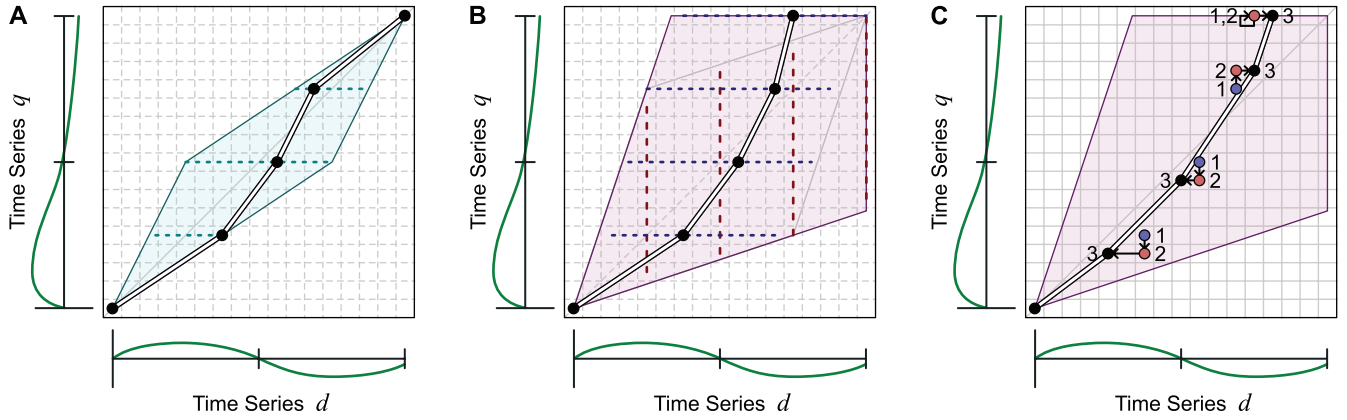

Fig. 3. COW and SCOW in alignment space. Both perform searches to find the best set of knots, or points of discontinuity, for a multi-segment alignment. (**A**) COW, which assumes no shorting and searches for good knots only in a single dimension, along the dotted lines. (**B**) The first step of SCOW, which searches independently in both dimensions. Subsequent steps are numbered in (**C**), as SCOW alternates horizontal and vertical movement of each knot until it converges.

It is filled using the following recurrence relations:

$$\gamma_{0,x} = \begin{cases} 0 & \text{if } x = 0 \\ -\infty & \text{otherwise} \end{cases} \tag{1}$$

$$\gamma_{k,x} = \max_{y \in \text{pred}(x,k)} \left[\gamma_{k-1,y} + \text{cor}\left(d(y,x), q(K_{k-1}, K_k)\right) \right] \tag{2}$$

where cor is the Pearson correlation, $q(a,b)$ represents a subseries of q from a to b and $d(a,b)$ is defined likewise. The predecessor function lists valid starting locations in d for segments ending at x:

$$\text{pred}(x,k) = \begin{matrix} x - \frac{|q|}{|d|}(K_k - K_{k-1}) - t, \ldots, \\ x - \frac{|q|}{|d|}(K_k - K_{k-1}) + t \end{matrix}, \tag{3}$$

with t being a user-defined 'slack parameter' that controls the size of the search space.

The best alignment, and its resulting score, is represented by the element of γ that corresponds to the end of the global alignment:

$$\text{BestScore}(\Gamma) = \gamma_{m,|d|}. \tag{4}$$

Note that COW can be used to align a one-channel time series, such as the expression profile of a single gene, or a multi-channel time series, such as the expression profile of a set of genes. The only difference between these two cases is in how the correlations are calculated.

A limitation of COW is that it forces the entirety of both series to be aligned to each other; it cannot short the alignment. Also, COW is apt to align segments which differ greatly in magnitude because it scores by correlation. Further, the computation in Equation (2) may sometimes return to an undefined value if the input segments do not have a defined correlation (as when both segments consist of all zeros).

Our SCOW is designed to rectify these problems. As shown in Figure 3B, SCOW searches for optimal knots in both dimensions. It first finds optimal knots with respect to q using evenly spaced knots in d, and with respect to d using evenly spaced knots in q. It uses the better alignment from these two passes as the starting point for an iterative process. From then on it alternates, which dimension's knot coordinates it holds constant, using the coordinates found by the previous pass as the constant knots in the next one. This iterative process is illustrated in Figure 3C, and Table 1 provides pseudocode describing the SCOW algorithm.

There are two different recurrence relations used in SCOW's dynamic programming formulation:

$$\gamma^q_{k,x} = \max_{y \in \text{pred}(x,k)} \left[\gamma^q_{k-1,y} + \text{score}\left(d(K^d_{k-1}, K^d_k), q(y,x)\right) \right], \tag{5}$$

Table 1. The pseudocode for SCOW

```
procedure SCOWAlign(series d, series q,
          set of genes G):

 // initial passes //

 K^q ← evenly spaced integers from 0 to |q|
 K^d ← evenly spaced integers from 0 to |d|
 calculate Γ^q, Γ^d using G
 if (BestScore(Γ^q) > BestScore(Γ^d)): α ← q
 else: α ← d
 K^α ← Traceback(Γ^α)

 // main loop //

 repeat:
    swap-dimension α
    calculate Γ^α using G
    calculate BestScore(Γ^α)
    K^α ← Traceback(Γ^α)
 until K^q, K^d converge
```

Knots are recalculated at least three times. The `Traceback` function extracts the best knots found from the previous pass to use in the next one.

$$\gamma^d_{k,x} = \max_{y \in \text{pred}(x,k)} \left[\gamma^d_{k-1,y} + \text{score}\left(d(y,x), q(K^q_{k-1}, K^q_k)\right) \right]. \tag{6}$$

The matrix Γ^q is calculated when the algorithm searches for knots with respect to q and holds them constant with respect to d, while Γ^d is calculated during the opposite case. The vectors K^q and K^d represent the coordinates of the knots in each dimension. The predecessor function is altered so as to not center around the line with slope $|q|/|d|$ but instead to enable a cone-shaped search space (as illustrated in Figure 3B) since we want to consider shorted alignments:

$$\text{pred}(x,k) = \begin{matrix} \max\left[x - \lambda(K_k - K_{k-1}), \frac{1}{\lambda} K_{k-1} \right], \ldots, \\ \min\left[x - \frac{1}{\lambda}(K_k - K_{k-1}), \lambda K_{k-1} \right] \end{matrix}, \tag{7}$$

where λ is the maximum slope allowed the aligning path in alignment space. In addition, SCOW does not assume a global alignment, but searches the last

row of the matrix for the best scoring alignment using m segments:

$$\text{BestScore}(\Gamma^{\alpha}) = \begin{cases} \gamma^{\alpha}_{m,|d|} & \text{if other dimension shorted} \\ \max_j \gamma^{\alpha}_{m,j} & \text{otherwise} \end{cases}. \quad (8)$$

This allows SCOW to short in the current dimension, if the other dimension is not already shorted. Thus the alignment found cannot short both q and d. The effect on the search can be seen in Figure 3C: the last knot cannot move down during the first step, because doing so would short both dimensions.

In addition to a different search procedure, SCOW also differs somewhat from COW in the function it uses to score alignments. The scoring function presented here is similar to one we used in previous work (Smith *et al.*, 2008). In particular, the scoring function includes terms that incur penalties for segments that involve stretching and significant differences in amplitude. We use the term *stretching* to refer to distortions in the rate of some expression response, and the term *amplitude* to refer to distortions in the magnitude of the response. Consider, for example, the alignment shown in Figure 2. The first segment in this alignment involves a noticeable amplitude difference (Time Series B has a higher amplitude than Time Series A), and the last segment involves significant stretching (this part of the response in Time Series B happens more slowly than the corresponding part of Time Series A).

We define the score of an alignment segment to be:

$$\text{score}(q_i, d_i) = \text{cor}(q_i, d_i) - \frac{\log^2 s_i}{2\sigma_s^2} - \frac{\log^2 a_i}{2\sigma_a^2} \quad (9)$$

Here, q_i and d_i denote the i-th segments of series q and d, respectively, s_i is the amount of stretching in the alignment of the i-th segments, a_i is the amplitude difference, and cor is the Pearson correlation. The stretching s_i is defined as the ratio of lengths between q_i and d_i, and a_i is the amplitude ratio between the two as determined by a weighted least squares fitting procedure.

The form of the stretching and amplitude terms comes from a generative, probabilistic model we developed in earlier work (Smith *et al.*, 2008). This previous approach uses probability distributions over possible stretching and amplitude values that have the following form:

$$p(v) = \frac{e^{\frac{-\sigma^2}{2}}}{\sigma\sqrt{2\pi}} \times e^{\frac{-\log^2 v}{2\sigma^2}}. \quad (10)$$

The key property of this distribution is that it is symmetrical around 1 such that $P(x) = P(1/x)$. Thus stretching, and amplitude values that deviate from 1 are penalized, and the penalty is the same regardless of which series, q or d, is considered to have the distortion.

For all of our experiments with COW and SCOW, we calculate correlations in the following way. We first use B-splines (Rogers and Adams, 1989) to interpolate between the observations in our time series (which are typically sparsely sampled). To calculate correlations between segments q_i and d_i, we resample their spline approximations to the same predetermined number of values for the two segments. We also alternately add and subtract a tiny value ϵ to values in q_i and d_i, so that correlation is always defined and two segments with constant values will have a correlation of one.

Like COW, SCOW operates with a time complexity of $O(n^3)$, where n is the length of the interpolated series to be aligned. Further, many of the calculations in successive passes of SCOW are the same, and may be cached. In contrast, the segment-based method from our previous work took $O(n^5)$ time to do an exhaustive search for the best segments to align the series. The speed-up is dramatic: what took the old method an hour to calculate takes SCOW only a few seconds.

2.2 Clustered alignments

Now we describe the algorithm we have developed for computing clustered alignments. The goal of this algorithm is to find sets of genes that would have very similar alignments if they were aligned independently. The alignment represented by each cluster may be quite different from the alignments that the other clusters represent. This approach is motivated by the fact that the relationship between two similar time series may differ depending on which subset of genes we consider.

Table 2. The pseudocode for our clustered alignment algorithm

```
procedure ClusterAlignments(series d, series q,
          # clusters k):

 // initialize cluster centroids //

centroid[1] ← null alignment
for all (genes g):
  possible[g] ← ScoreGene(q,d,g,Align(q,d,{g}))
  best[g] ← ScoreGene(q,d,g,centroid[1])
for (i ← 2 to k):
  worst ← argmin_g(best[g] − possible[g])
  centroid[i] ← Align(q,d,{worst})
  for all (genes g): best[g] ←
          max(best[g], ScoreGene(q,d,g,centroid[i]))
repeat:

  // assignment step //

  for all (centroids c): set[c] ← ∅
  for all (genes g):
      s ← argmax_c (ScoreGene(q,d,g,c))
      set[s] ← set[s] ∪ g

  // update step //

  for all (centroids c): c ← Align(d,q,set[c])
until sets converge
```

The algorithm we have devised is a variant of traditional k-means clustering (Duda *et al.*, 2000). In k-means, each cluster is represented by a centroid and the clustering process involves iteratively refining the locations of these centroids. For example, if we were clustering points in $\mathbb{R}^n$, each centroid would be represented by a point in $\mathbb{R}^n$. In our clustered alignment method, each 'centroid' is represented by an alignment (e.g. such as the one illustrated in Fig. 2). In our algorithm, as in standard k-means, the number of clusters is determined by a parameter k that is provided as an input.

We reiterate that, in contrast to previous methods which have focused on identifying clusters of genes that have similar expression profiles, our algorithm is focused on identifying clusters in which the genes have similar warpings. The genes in one of our clusters may have very different expression profiles.

Table 2 shows the pseudocode for our alignment clustering method. It takes as input two series, termed d and q, and the number of clusters k. It relies on the subroutines `Align`, which returns the best alignment between two series based on a given set of genes, and `ScoreGene`, which returns the score of two series when aligned using a given alignment and a specified gene. We use SCOW to perform these functions, using `SCOWAlign` for `Align` while using Equation (8) for `ScoreGene`. However, we could substitute any other alignment algorithm for this purpose.

The first step in the method is to assign the initial alignment centroids. We use a greedy method, similar to that used by Ernst *et al*. (2005) to select a representative set of gene alignments as the centroids. The first centroid is taken to be the null alignment, which represents no warping. For each gene, we record a best possible score (when the alignment is based solely on that gene), and the best score seen so far for that gene using one of the current centroids. Each additional centroid is initialized by finding the gene with the largest difference between its best score so far and its possible high score. The new centroid is the alignment calculated using this selected gene alone. After each new centroid is determined, the best scores for all the genes

are modified to take the new centroid into account. We proceed until all k centroids are defined.

Now we perform the assignment step and the update step in turn until convergence. For the assignment step, we score every gene with every cluster's centroid and assign the gene to the cluster with the highest score. For the update step, we set each centroid to the alignment calculated by aligning q and d using just the set of genes assigned to the cluster.

We continue iterating until the cluster assignments do not change. Because SCOW performs a heuristic search, however, it is possible that the process will not converge. In practice, this is seldom a problem. We can simply stop iterating after a large number of iterations, or when infinite loop conditions are detected by retaining a short history of cluster assignments. Alternatively, we can guarantee convergence by using an alignment algorithm that is exact.

3 RESULTS AND DISCUSSION

In this section, we describe a set of computational experiments that are designed to (i) evaluate the alignment accuracy of SCOW and our clustered alignment method, and (ii) assess how well the clustered alignment algorithm is able to uncover sets of genes that share similar alignments across two time series.

3.1 SCOW experiments

In our first set of experiments, we are interested in testing the ability of the SCOW method to find accurate alignments. We do this in the context of the task illustrated in Figure 1. Here, we are given an expression profile as a query, and we want to identify the treatment in the database that has the expression profile most similar to the query. We construct queries for which we know the correct matching database treatments and their correct alignments.

The data we use comes from the EDGE toxicology database (Hayes *et al.*, 2005), and can be downloaded from `http://edge.oncology.wisc.edu/`. Our dataset consists of 216 unique observations of microarray data, each of which represents the expression values for 1600 different genes.[1] Each of these expression values is calculated by taking the average expression level from four treated animals, divided by the average level measured in four control animals. The data are then converted to a logarithmic scale, so that an expression value of 0.0 corresponds to the average basal level observed in the control animals.

Each observation is associated with a treatment and a time point. The treatment refers to the chemical to which the animals were exposed and its dosage. The time point indicates the number of hours elapsed since exposure occurred. Times range from 6 h up to 96 h. The data used in our computational experiments span 11 different treatments, and for each treatment there are observations taken from at least three different time points. Additionally we can assume that for all treatments, there exists an implicit observation at time zero. This is the time at which the treatment was applied, so all expression values are assumed to be at the basal level.

We assemble queries by randomly sub-sampling time series in our dataset. We assemble 10 such queries from each treatment. We build each query by first selecting the number of observations to be in it, then choosing which time points will be represented, and finally picking an observation for each of these time points. The query sizes are chosen from a uniform distribution that ranges from one up to the number of observed times in the given treatment. The maximum size of a query is eight, although most consist of four or fewer observations. The time points are chosen uniformly as are the observations for each chosen time.

To test the ability of our approach to find accurate alignments in situations that require warping, we also assemble cases in which we distort the query time series temporally. We use three different distortions. The first one doubles all times in the first 48 h (i.e. it stretches the first part of the series), and then halves all times (plus an offset for the doubling) for the next 24 h. The second distortion halves for the first 36 h and then doubles for 60 h. The third one triples for the first 60 h and then thirds for another 20 h. It should be noted that not all the treatment observations extend this long in time. The short ones (e.g. those for which we only have measurements up to 24 or 48 h) will thus not be distorted as much as the long ones.

We then classify and align the query using all the other observations as the database. We preprocess both the query and the 11 database treatments using B-splines (Rogers and Adams, 1989) to reconstruct pseudo-observations at every 4 h (starting at time zero, when all expression values are at the basal level). We then align the query against all 11 treatments using our method. We return the database treatment with the highest scoring alignment, as defined by Equation (8). Because the alignment also maps each query time to a database treatment time, we can find the temporal error for any query time point. We then measure how accurately we are able to (i) identify the treatment from which each query series was extracted, and (ii) align the query points to their actual time points in the treatment. We refer to the former as *treatment accuracy* and the latter as *alignment accuracy*.

We consider several other alignment methods as baselines. The first is COW (Nielsen *et al.*, 1998), as described in Section 2. The second is a generative method we previously developed (Smith *et al.*, 2008), which we refer to as Generative Multisegment. Like SCOW, it finds alignments which consist of multiple segments each of which can have different warping parameters. However, the Generative Multisegment scoring function is based on a generative, probabilistic model, rather than correlation. Further it performs a complete search for the best segments to use, rather than using the heuristic search of SCOW.

The next baseline we consider is traditional Euclidean dynamic time warping (Sakoe and Chiba, 1978; Sankoff and Kruskal, 1983). Briefly, this method computes alignments by creating a matrix Γ with elements defined recursively as

$$\gamma_{i,j} = D(d_i, q_j) + \min\left[\text{pred}_{\text{DTW}}(\gamma_{i,j})\right] \tag{11}$$

where $D(d_i, q_j)$ is the Euclidean distance between points d_i and q_j in the two series and $\text{pred}_{\text{DTW}}(\gamma_{i,j})$ refers to the matrix elements adjacent to $\gamma_{i,j}$ with both indices less than or equal to i and j, respectively. The first element $\gamma_{0,0}$ is just the Euclidean distance at time 0, and each other element $\gamma_{i,j}$ is the score of warping d from times 0 to i and q from 0 to j. We then create a normalized score matrix $\overline{\Gamma}$ where

$$\overline{\gamma}_{i,j} = \gamma_{i,j} / \sqrt{|i|^2 + |j|^2}. \tag{12}$$

This makes it reasonable to compare warpings with different treatments, where one or the other dimension has been shorted.

[1] Technically, the expression measurements correspond to clones selected from liver-derived EST and full-length cDNAs. These clones represent products for 1600 unique genes.

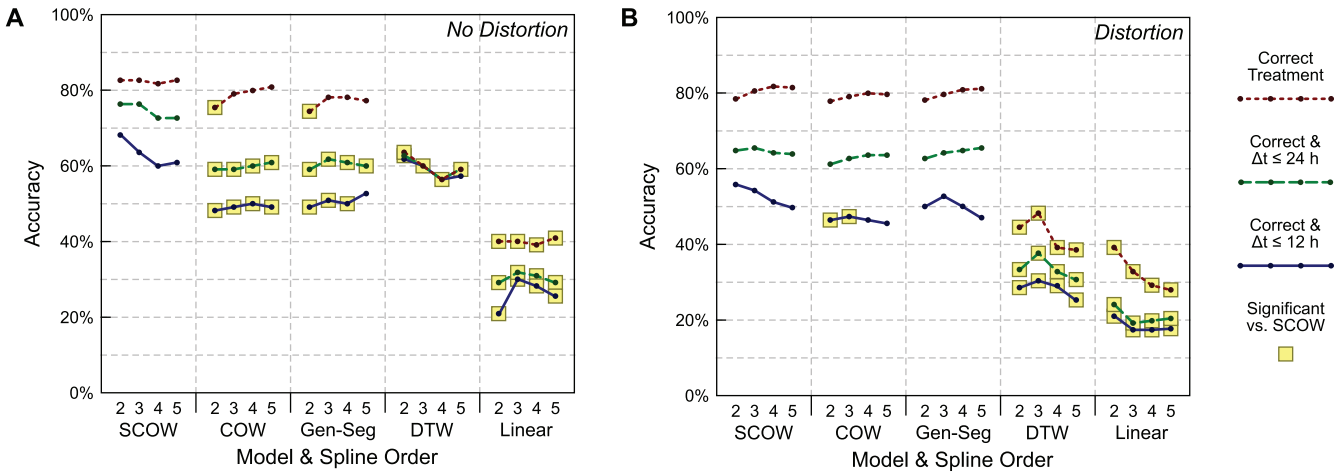

Fig. 4. Treatment and alignment accuracies when there is no temporal distortion (**A**), and when there is (**B**). The top lines represent treatment accuracy, while the bottom two lines add the criterion that the predicted times are within 24 h and 12 h, respectively, of the actual time, on average. For each alignment method, we show results when splines of various orders are used to interpolate the time series before alignments are calculated. Highlights represent cases in which there is a significant difference in accuracy from the corresponding SCOW case ($P \leq 0.05$ with McNemar's χ^2-test).

Finally, we consider linear parametric warping. This is similar to the method explored by Bar-Joseph *et al*. (2003), except that we make the assumption that the series are aligned at time zero. To find an alignment, we search possible slopes of the alignment line, and return the slope that results in the least average Euclidean distance between the query and the given database treatment.

For these experiments, SCOW, COW and Generative Multisegment use three segments in their alignments, and we set σ_s and $\sigma_a = 10$. Using more segments and setting σ_s and σ_a to other values yields substantially similar results.

The results of this experiment are shown in Figure 4. Figure 4A and B shows results for the queries without distortion and results for the distorted queries, respectively. For each method, the top line represents treatment accuracy with different orders of splines, the middle line represents alignment accuracy by adding the criterion that the average time error in the mapping is less than or equal to 24 h, and the bottom line shows alignment accuracy where this tolerance is decreased to 12 h. Highlighted boxes denote points that are significantly different from the corresponding SCOW point, as determined by McNemar's χ^2-test.

There are several interesting conclusions we can draw from these results. First, it is clear that the multi-segment alignments computed by SCOW, COW and Generative Multisegment are superior to the alignments determined by ordinary dynamic time warping and the linear alignment method. Second, SCOW finds more accurate alignments than the other two multi-segment algorithms, COW and Generative Multisegment. Based on these results, we conclude that SCOW is a state-of-the-art alignment method for gene-expression time series, and we therefore use it as the core alignment method for our clustered alignment approach.

3.2 Clustered alignment experiments

In our second set of experiments, we are interested in testing the ability of our clustered alignment algorithm to identify sets of genes that should share a common alignment. We first conduct an experiment designed to determine if our clustered alignment method

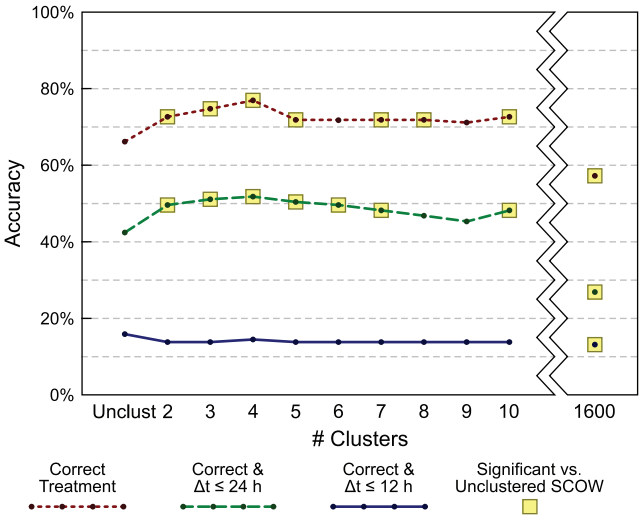

Fig. 5. Treatment and alignment accuracies, varying by the number of clusters when using SCOW. In the final case (1600), we warp every gene separately. Highlighted points are significantly different from the unclustered case, ($P \leq 0.05$ under McNemar's χ^2-test).

is able to find more accurate alignments when there are sets of genes that have different, known 'correct' alignments. This experiment is similar to the one in the previous section—we use the same data and substantially the same methodology. The difference is that we simultaneously apply five different temporal distortions to every query: each one is applied to $^1/_5$ of the genes. We then run our clustered alignment method, in conjunction with SCOW, on the data, allowing the number of clusters k to range from one (i.e. unclustered, ordinary SCOW) to 10. We also run the experiment with $k = 1600$, which warps every gene separately.

The results for queries containing three or more observations are shown in Figure 5. These results show the value of the clustered alignment approach with this dataset. The accuracy of the alignments

increases as k increases, until about $k=4$. After this point, there is a slight degradation in accuracy. For almost all values of k tested, however, the treatment and the 24 h alignment accuracies are greater with the clustered alignment method than with ordinary SCOW.

With queries containing fewer than three observations, the clustered alignment method actually results in somewhat less accurate alignments than the non-clustered method (i.e. ordinary SCOW). These results can be explained by a bias-variance trade-off (Geman *et al.*, 1992). The gene-expression data we use (like most expression time series) is sparse in time, and prone to noise (because of both technical limitations and biological variability among the animals). The sparsity and noise mean that it is difficult to compute accurate single-gene alignments. Aggregating genes into clusters has a regularization effect as this alignment error is averaged out (Bar-Joseph *et al.*, 2003). The more genes there are in a cluster, the greater the regularization effect. Thus we want to find the ideal trade-off between the high-bias approach of few clusters (or one cluster, in the limit), and the high-variance approach of many clusters. The variance component of the error is more significant in the case when the queries are short. We can conclude, however, that the clustered alignment approach demonstrates good predictive value for moderately sized queries and a range of values of k.

In our second experiment, we are interested in identifying sets of genes that are distorted in similar ways in a knockout experiment focusing on circadian rhythms. Mop3 is a transcription factor in hepatocytes (Bunger *et al.*, 2000, 2005) that is a positive regulator of circadian rhythm and activates the transcription of genes such as Per1 and Tim. There are two sets of mice in this experiment. The control group has a functional Mop3 gene, while the knockout group does not. This is a time-course study based on *Zt* which stands for *Zeitgeiber time*—the number of hours after exposure to light begins. Before Zt0, the mice are kept in darkness for a period. At Zt0 the lights turn on, and at Zt12 they turn off again. At intervals of 4 h from Zt0 to Zt20, three mice from each group are sacrificed, and microarrays are derived from pooled RNA samples from the livers of each set of mice. In all, 27 962 genes are measured. We interpolate the series with B-splines so that we can sample measurements every 2 h.

When aligning the control and knockout time series, we want to allow phase shifting. That is, we want to allow alignments of the two time series are not necessarily aligned at Zt0. In our previous experiment, it was reasonable to assume that the expression responses were all identical at time zero. We cannot make that assumption in this case, however. We modify SCOW to allow phase shifting by first concatenating the control time series with itself, to obtain 2 days worth of data. When computing alignments, we allow the control series to short at both ends by redefining the initialization [Equation (1)] of Γ^d:

$$\gamma_{0,x}=0. \tag{13}$$

However, we disallow the alignment from shorting the knockout series, at either end, by using Equation (4) to score Γ^q. Thus, all knockout series times must be mapped to some time in the control series, but the zero times need not correspond.

Figure 6 shows alignments for several genes in each cluster, as determined by our clustered alignment algorithm. Here, we set the number of clusters $k=5$. Each panel represents one of the clusters, and within each one we show the three genes with the highest relative scores for that cluster. The white alignment path in each plot represents the consensus alignment, when all genes are warped as a unit. The black alignment path represents the cluster's individual alignment. Note that we only show 1 day in the control dimension rather than 2 days. The alignments in panels C, D and E, all extend into 2 days. This is shown by a break in the black alignment path, as it wraps back to the left side and the beginning of the second day.

The clustered alignment allows us to uncover sets of genes that are disrupted in a similar manner by the knockout, even when their expression profiles are quite different. It is clear that the clustered alignments align the series better than the consensus alignment. Peaks and valleys in the expression data line up well for the black cluster alignment paths, whereas they often do not for the white consensus ones. For example, the genes in panel E have undergone a large phase shift. The consensus path often matches segments with quite different expression profiles, whereas the cluster path shifts the starting point by 12 h and achieves good agreement. In panel D, the genes appear to be acting more quickly in the knockout mice, while the consensus alignment would indicate they are acting more slowly. It should also be noted that often the genes within a cluster have very different expression profiles. Consider panel D, in which the profiles for the three genes are all quite different, but the mapping between control and knockout is similar. This effect illustrates the advantage of clustering alignments in contrast to clustering the expression profiles directly.

4 CONCLUSION

Alignment algorithms provide a valuable approach for gaining biological understanding from gene-expression time series. A variety of methods have been employed for such analyses, including dynamic time warping, linear alignment algorithms and multi-segment alignment methods.

We have presented new methods which advance the state of the art in two ways. Most importantly, we have developed an algorithm which is able to compute clustered alignments. This algorithm relaxes the assumption, common to previous work in expression time-series alignment, that all genes should be warped in the same way. Instead, our method identifies sets of genes that share a common alignment. It does this by simultaneously clustering genes and computing a shared alignment for the genes in each cluster. The second contribution introduced here is a new multi-segment alignment method, called SCOW, that features the ability to calculate 'shorted' alignments, a correlation-based scoring function, and an efficient dynamic programming algorithm for computing alignments.

The results of our empirical evaluation indicate that both the clustered alignment approach and SCOW improve the accuracy of alignments computed with sparse time series from a toxicogenomics dataset. Additionally, we applied our clustered alignment approach to a dataset involving a conditional Mop3 knockout in mouse liver. This analysis illustrates the power of the clustered alignment approach to find sets of genes that share similar temporal distortions.

Funding: National Institutes of Health/NIEHS (grant R01-ES012752); National Institutes of Health/NLM (grant R01-LM07050); National Institutes of Health/NCI (grants P30-CA014520 and T32-CA009135).

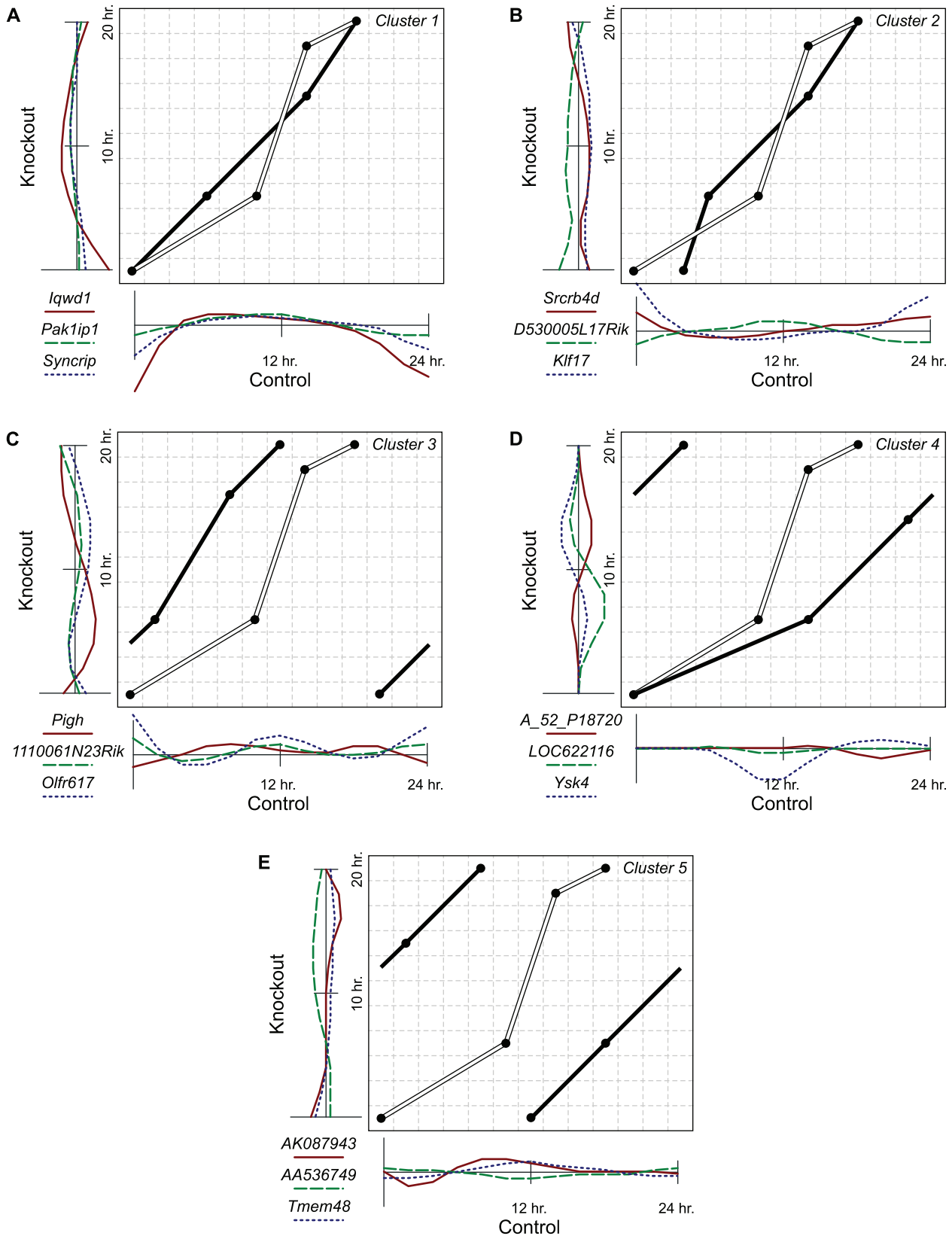

Fig. 6. Alignment clusters found by our method for the Mop3-knockout circadian data. Each panel shows the top three genes for a different cluster. The white alignment paths represent the consensus alignment for all the genes, while the black paths represent the cluster-specific alignments.

Conflict of Interest: none declared.

REFERENCES

Aach,J. and Church,G. (2001) Aligning gene expression time series with time warping algorithms. *Bioinformatics*, **17**, 495–508.

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

Bunger,M. *et al*. (2000) Mop3 is an essential component of the master circadian pacemaker in mammals. *Cell*, **103**, 1009–1017.

Bunger,M. *et al*. (2005) Progressive arthropathy in mice with a targeted disruption of the Mop3/Bmal-1 locus. *Genesis*, **41**, 122–132.

Criel,J. and Tsiporkova,E. (2006) Gene time expression warper: a tool for alignment, template matching and visualization of gene expression time series. *Bioinformatics*, **22**, 251–252.

Duda,R.O. *et al*. (2000) *Pattern Classification*. Wiley-Interscience Publication.

Eisen,M.B. *et al*. (1998) Cluster analysis and display of genome-wide expression patterns. *PNAS*, **95**, 14863–14868.

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

Geman,S. *et al*. (1992) Neural networks and the bias/variance dilemma. *Neural Comput.*, **4**, 1–58.

Hayes,K. *et al*. (2005) EDGE: a centralized resource for the comparison, analysis and distribution of toxicogenomic information. *Mol. Pharmacol.*, **67**, 1360–1368.

Keogh,E. (2003) Efficiently finding arbitrarily scaled patterns in massive time series databases. In *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*. Springer, pp. 253–265.

Leng,X. and Müller,H.-G. (2006) Classification using functional data analysis for temporal gene expression data. *Bioinformatics*, **22**, 68–76.

Listgarten,J. *et al*. (2005) Multiple alignment of continuous time series. In Saul,L. *et al.* (eds) *Advances in Neural Information Processing Systems 17*. MIT Press, pp. 817–824.

Liu,X. and Müller,H.-G. (2003) Modes and clustering for time-warped gene expression profile data. *Bioinformatics*, **19**, 1937–1944.

Nielsen,N.V. *et al*. (1998)Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping. *J. Chromatogr. A*, **805**, 17–35.

Rogers,D. and Adams,J. (1989) *Mathematical Elements for Computer Graphics*. McGraw-Hill.

Sakoe,H. and Chiba,S. (1978) Dynamic programming algorithm optimization for spoken word recognition. *IEEE ASSP Mag.*, **26**, 43–49.

Sankoff,D. and Kruskal,J. (1983) *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley.

Smith,A.A. and Craven,M. (2008) Fast multisegment alignments for temporal expression profiles. In *Proceedings of the 7th International Conference on Computational Systems Bioinformatics.* Vol. 7. Imperial College Press, pp. 315–326.

Smith,A.A. *et al*. (2008) Similarity queries for temporal toxicogenomic expression profiles. *PLoS Comput. Biol.*, **4**, e1000116.