FAST MULTISEGMENT ALIGNMENTS FOR TEMPORAL EXPRESSION PROFILES

Adam. A. Smith* and Mark Craven
Departments of Computer Sciences and Biostatistics & Medical Informatics, University of Wisconsin, Madison, Wisconsin 53706, USA
Email: aasmith@cs.wisc.edu, craven@biostat.wisc.edu

We present two heuristics for speeding up a time series alignment algorithm that is related to dynamic time warping (DTW). In previous work, we developed our *multisegment* alignment algorithm to answer similarity queries for toxicogenomic time-series data. Our multisegment algorithm returns more accurate alignments than DTW at the cost of time complexity; the multisegment algorithm is $O(n^5)$ whereas DTW is $O(n^2)$. The first heuristic we present speeds up our algorithm by a constant factor by restricting alignments to a cone shape in alignment space. The second heuristic restricts the alignments considered to those near one returned by a DTW-like method. This heuristic adjusts the time complexity to $O(n^3)$. Importantly, neither heuristic results in a loss in accuracy.

1. INTRODUCTION

Characterizing and comparing temporal gene-expression responses is an important computational task for answering a variety of questions in biological studies. We have previously presented an approach for answering similarity queries about gene-expression time series that is motivated by the task of characterizing the potential toxicity of various chemicals[1]. This approach is designed to handle the plethora of problems that arise in comparing gene expression time series, including sparsity, high-dimensionality, noise in the measurements, and the local distortions that can occur in similar time series. Our experimental evaluation showed that our approach produces more accurate alignments and classifications of gene-expression time series than a handful of alternative approaches, and is robust to relative distortions in time between similar chemical treatments. However, this accuracy comes at the cost of efficiency: the algorithm's time complexity is $O(n^5)$. In this paper, we present two heuristic methods for speeding up our alignment algorithm. We show that these heuristics result in significant speedups without sacrificing the accuracy of the resulting alignments.

The task that we consider is motivated by the need for faster, more cost-efficient protocols for characterizing the potential toxicity of industrial chemicals. The effects of toxic chemicals may often be predicted by how they influence global gene expression over time[2]. By using microarrays, it is possible to measure the expression of thousands of genes simultaneously. It is likely that transcriptional profiles will soon become a standard component of toxicology assessment and government regulation of drugs and other chemicals.

The source we use for toxicology-related gene expression data is the EDGE (Environment, Drugs and Gene Expression) database[2]. EDGE contains expression profiles from mouse liver tissue following exposure to a variety of chemicals and physiological changes, which we refer to as *treatments*. Some of the treatments in EDGE have been assayed as time series. Figure 1-A provides a simplified illustration of the type of data with which we are concerned. The small database in this figure contains time series data for four different treatments, each of which includes measurements for three genes. The true, underlying expression response is not known, but instead the database contains sampled observations which may be noisy. We use the term *observation* to refer to the expression measurements made at a single time point in a treatment.

Figure 1-B then shows the computational task. Given an expression profile as a query, we want to identify the treatment in the database that has the expression profile most similar to the query. In the general case, the query and/or some of the database treatments are time series. In this case, we want to also determine the temporal correspondence between the query and putatively similar treatments in the database. In the toxicology domain, we are

*Corresponding author.

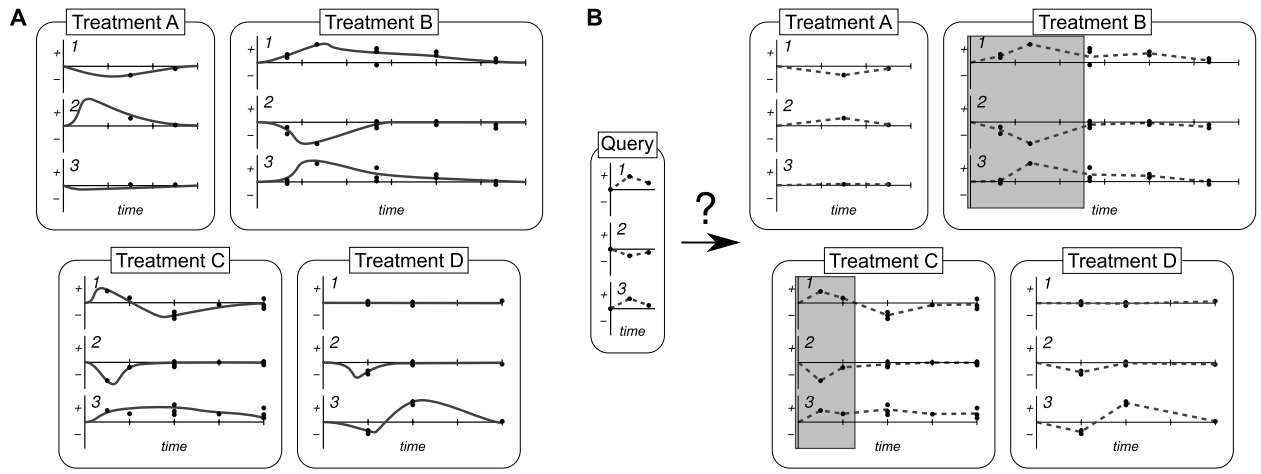

Fig. 1. An example of the similarity-query task for four different treatments with three genes. In Panel A the curves show the actual hidden expression profile for each treatment, even though we must rely on the noisy sampled observations (the points). In Panel B we have reconstructed the profiles at unobserved times, and used them to perform a similarity query. The highlighted areas represent possible good matches.

interested in answering this type of query in order to characterize poorly understood chemicals.

We have developed an approach that is designed to handle several key challenges that this task presents.

- The time series available from toxicological studies are typically sparse, containing measurements from fewer than ten time points.
- Because the time series have been sampled at non-uniform time intervals which vary between treatments, the time points present in a given query may not correspond to measured points in database series.
- Queries may vary in their number of observations or their extent. Some queries may consist of only a single observation, whereas others may contain multiple time points. Some may span only a few hours whereas others include measurements taken over days.
- A given query and its best match in the database may differ in the amplitude, temporal offset, or temporal extent of their responses. For example, the expression profile represented by a query treatment may be similar to a database treatment except that the gene expression responses are attenuated, or occur later, or take place more slowly. Alternatively, the query may be similar to a truncated version of the database series, or vice versa.

Our approach to this task involves first using an interpolation algorithm to reconstruct unobserved expression values from sparse time series, and then using an alignment algorithm to find the highest scoring alignment of the query series to each treatment series in the database. The approach returns the treatment from the database that is most similar to the query, and the calculated alignment between the two series.

Several different methods have been applied to the task of aligning gene-expression time series. Aach and Church[3] were the first to apply the method of *dynamic time warping*[4, 5] to gene expression profiles, and other groups have followed[6, 7]. Dynamic time warping, originally developed for speech recognition problems, is an approach for aligning pairs of time series that uses dynamic programming to find an optimal alignment with respect to a given scoring function. Although DTW has a time complexity of $O(n^2)$, Ratanamahatana and Keogh[8] have shown that using bounding heuristics can effectively make DTW run in $O(n)$. However their method is designed for *global* alignments, which align all of one series to the entirety of the other. Our method, in contrast, does not force the two series to be globally aligned. Instead, it permits a type of *local* alignment in which the end of one series is unaligned. We refer to this case as *shorting* the alignment. This aspect of the approach is motivated by the consideration that

one of the series may show more of the temporal response than the other (e.g., one series may not have been measured for as long, or may have responded more quickly).

Bar-Joseph et al.[9] used a warping method that finds a linear mapping between the two time series being aligned. Although it allows local alignments like our method, the linear model does not adequately represent complex alignments. Our method considers alignments that represent a middle ground, in terms of expressiveness, between dynamic time warping and linear warping approaches. Our method is based on a "multisegment" model that warps different regions of the series by different amounts.

Another time-series alignment approach that is somewhat similar to our multisegment method is *correlation optimized warping* (COW)[10]. This method compares time series by dividing them into several roughly equal segments and summing the Pearson's correlations of corresponding segments. Unlike our approach, the COW method assumes that the series are to be globally aligned, without any shorting. Further, the use of correlation can be limiting as COW is unable to distinguish between two series that are proportional to one another.

In previous work, we evaluated our multisegment alignment method in the context of aligning and classifying gene-expression profiles from the EDGE database. This empirical evaluation showed that our multisegment method returned more accurate alignments and classifications than dynamic time warping, two linear alignment methods, and the COW algorithm. The disadvantage of the multisegment method is that its computational complexity is $O(n^5)$ where n is the number of time points in the series being aligned (assuming the two series have the same length). Although the number of *observed* time points is typically small in the series we consider, n is considerably larger because each series is represented by interpolated "pseudo-observations" in addition to the observed time points.

2. TIME SERIES ALIGNMENT METHODS

In this section we discuss two previously developed methods for aligning two series. Figure 2 illustrates the type of alignment problem we consider. The figure shows the types of alignments calculated by dynamic time warping and by our multisegment method[1]. (For simplicity, the figure shows each treatment as consisting of only a single gene.) These alignment paths exist in *alignment space*, where each dimension represents the time of one of the aligned series. A point (x, y) on a path corresponds to a mapping between Series 1 at time x and Series 2 at y. The diagonal represents a special path, in which no warping of time takes place. Here, both alignments short, so that the whole of Series 2 is aligned with only a portion of Series 1.

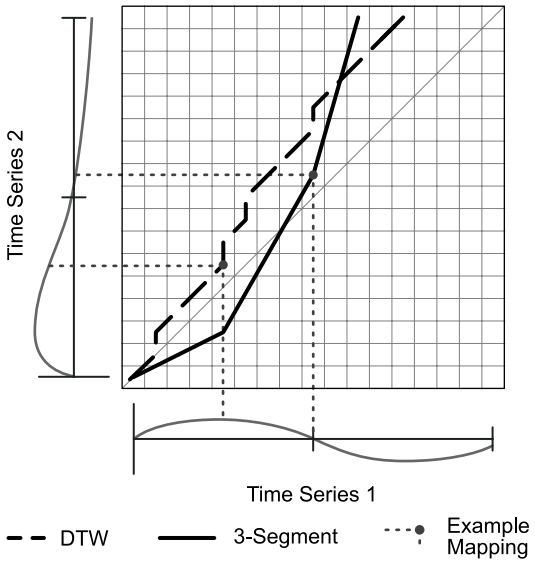

Fig. 2. Aligning two time series in *alignment space* with dynamic time warping and our three-segment model. We refer to this graph as an *alignment matrix*. A point (x, y) corresponds to a mapping between Series 1 at time x and Series 2 at y. The paths thus show the overall alignments chosen by both methods. Notice that the alignments *short* before Series 1 has ended, as there is no evidence that Series 2's expression has begun to increase again at the end.

2.1. Dynamic Time Warping

Dynamic time warping[4, 5] is often used for time series alignment problems. Briefly, this method computes an *alignment matrix* Γ from two series as shown in Figure 2. In our context, the series are a query series q and a database series d. Each element $\gamma(x, y)$ holds the best score aligning q, up to time x, against d, up to time y. The matrix elements

are calculated recursively as:

$$\gamma(x,y) = D_E(q_x, d_y) + \min \begin{cases} \gamma(x-1,y) \\ \gamma(x,y-1) \\ \gamma(x-1,y-1) \end{cases} \quad (1)$$

where $D_E(q_x, d_y)$ is the Euclidean distance between points q_x and d_y in the two series. The base element $\gamma(0,0)$ is just the Euclidean distance at time 0.

Traditional DTW then returns $\gamma(q.r, d.r)$, where $q.r$ and $d.r$ are the rightmost (last) times of the two series, along with the path that resulted in this score. However we are interested in possibly shorting the alignment, finding a local alignment rather than a global one. In this case, allowed alignments are those that explain the entire extent of at least one of the two given time series. We scan the elements of Γ that represent alignments that include the entirety of the query series, the entirety of the database series, or both, and return the best one:

$$\text{bestscore} = \min_{a \le q.r,\ b \le d.r} \begin{cases} \frac{\gamma(a,d.r)}{\sqrt{|a|^2+|d.r|^2}} \\ \frac{\gamma(q.r,b)}{\sqrt{|q.r|^2+|b|^2}} \end{cases}. \quad (2)$$

The variables a and b represent positions in the two time series. Given series of length m and n, the alignment matrix has mn entries to be calculated. Each of these calculations takes constant time. Thus, if $m \approx n$, the time complexity is $O(n^2)$.

2.2. Multisegment Time Series Alignment

The three-segment path in Figure 2 shows the type of alignment that our multisegment model calculates. In each segment the amplitude and stretching relationships between the two series are somewhat different. We use the term *stretching* to refer to distortions in the rate of some response, and the term *amplitude* to refer to distortions in the magnitude of the response. The total number of segments is specified in advance.

To determine the similarity between a query time series q and a particular database series d, we can calculate how likely it is that q is a somewhat distorted exemplar of the same process that resulted in d. In particular, we can think of a generative process that uses d to generate similar expression profiles. We can then ask how probable q looks under this generative process.

Given this generative process idea, we calculate the probability of a particular alignment of query q given a database series d as follows:

$$P(q|d,s,a) = P_m(m) \prod_{i=1}^{m} P_s(s_i) P_a(a_i) P_e(q_i|d_i, s_i, a_i), \quad (3)$$

where m is the number of segments in the alignment, q_i and d_i refer to the expression measurements for the ith query and database segments respectively, and s_i is the stretching value and a_i is the amplitude value for the ith segment. The location of each segment pair is assumed to be given here. P_m represents a probability distribution over the number of segments in an alignment, up to some maximum number M of allowed segments. P_s represents a probability distribution over possible stretching values for a pair of segments, P_a represents a probability distribution over possible amplitude values, and P_e represents a probability distribution over expression observations in the query series, given the database series and the stretching and amplitude parameters.

We omit the details of this generative model here, but we note that it results in a scoring function that can be used to assess the likelihood of any given alignment. Moreover, we can find optimal scoring alignments under this model using dynamic programming. The core of the dynamic program involves filling in a three-dimensional matrix Γ in which each element $\gamma(i,x,y)$ represents the best score found with i segments that align the query subseries from time 0 to x with the database subseries from time 0 to y. Here, x and y represent time points in the two series.

We define $\gamma(i,x,y)$ with the following recurrence relation:

$$\gamma(i,x,y) = \max_{a<x,\ b<y} \begin{cases} \log P_m(i) + \gamma(i-1,a,b) \\ \quad + \mathsf{score}(a,x,b,y) \\ \quad \textbf{if } x = q.r \textbf{ or } y = d.r \\ \gamma(i-1,a,b) \\ \quad + \mathsf{score}(a,x,b,y) \\ \quad \textbf{otherwise} \end{cases}. \quad (4)$$

Here, $\log P_m(i)$ is a scoring term that considers only the number of segments used in the alignment, and score is the remainder of the scoring function, which takes into account the stretching and amplitude distortions involved in aligning a pair of segments, in

addition to how well the observations from the two segments match given these distortions. Again, the indices a and b represent positions in the two time series and $q.r$ and $d.r$ refer to the rightmost (last) time coordinates in the query series and the database series, respectively. The base case is similar. We then find the highest scoring element of Γ that corresponds to a legal shorting:

$$\text{bestscore} = \max_{i,\ a \le q.r,\ b \le d.r} \begin{cases} \gamma(i, a, d.r) \\ \gamma(i, q.r, b) \end{cases}, \qquad (5)$$

along with its generating path.

Assuming that the two series are of equal length n, Γ is of size $O(n^2)$. For each entry of Γ we must scan through $O(n^2)$ previous segments, and perform $O(n)$ calculations to score a pair of segments. This results in a final time complexity of $O(n^5)$.

2.3. Data

The data we use in our experiments comes from the EDGE toxicology database[2], and can be downloaded from *http://edge.oncology.wisc.edu/*. Our data set consists of 216 unique observations of microarray data, each of which represents the expression values for 1600 different genes. 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 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 six hours up to 96 hours. 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.

The alignment methods we use all assume that the data is sampled at regular intervals. Because that is not the case here, all our experiments use an interpolation preprocessing step to generate regularly sampled pseudo-observations. Database and query series are interpolated using third order splines in most cases. We use second order splines when there are only two data points.

2.4. Experiments

In order to evaluate our multisegment alignment method on the EDGE data, we assembled query series by randomly sampling a random number of observations of the same treatment but at different times. We then tested the query against a database built from all the remaining observations. In some cases, the expression responses induced by similar treatments may evolve at different rates. To simulate this situation, we temporally distort some query series. For example, one of the distortions doubles all times in the first 48 hours (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 hours. The other distortions were similar.

We note that this task is only a surrogate for the actual task with which we are concerned: classifying uncharacterized chemicals and aligning them with the most similar treatment in the database. It is a useful surrogate, however, because it is a task in which we know the most similar treatment and the correct alignment of the query to this treatment.

We preprocessed each query and the eleven database treatments using splines to reconstruct pseudo-observations at every four hours. In this experiment, our method returned the database treatment with the highest scoring alignment, as defined by Equation 5. We then measured 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 *classification accuracy* and the latter as *alignment accuracy*.

In the experiments we report here, our multisegment method is limited to three segments in its alignments. Our previous experiments [1] indicate that the accuracy of the multisegment method remains stable when it is allowed to use more segments.

We considered several other alignment methods as baselines. Dynamic time warping is described in Section 2.1. Linear parametric warping finds the coefficient which, when multiplied by the query times, results in the smallest Euclidean distance between the series. It is similar to the method used by Bar-Joseph et al.[9]. Finally *correlation optimized warping*

(COW)[10] is another segment-based method that divides both series into the same number of segments and then sums the cross correlations of corresponding segments.

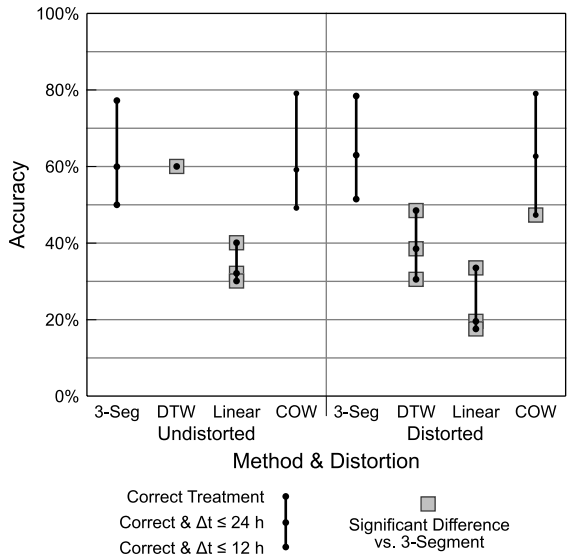

Fig. 3. Classification and alignment accuracies of several methods, including our three-segment model. Each line represents a different method. In each, the top point represents classification accuracy, while the bottom two points add the additional correctness criteria of the average alignment error being less than 24 hours and 12 hours, respectively. Highlighted points are those that are significantly different from the multisegment method ($p \leq 0.05$), by McNemar's χ^2 test.

The results of these experiments are shown in Figure 3. The left half represents those cases in which we did not distort the queries temporally, while in the right half we show the cases in which we did. Each line represents a different method. For each method the top point represents classification accuracy, the middle point represents alignment accuracy by adding the criterion that the average time error in the mapping is less than or equal to 24 hours, and the bottom point shows alignment accuracy where this tolerance is decreased to 12 hours. Highlighted points are those that are significantly different from the multisegment method ($p \leq 0.05$) according to McNemar's χ^2 test. Only COW exhibits accuracies competitive with those of our multisegment method. These estimates of COW's alignment accuracies are optimistic, however, because we have run COW with many settings for its parameters and report only the best results here.

Thus our multisegment method dominates the others in terms of both classification and alignment accuracy, but this comes at the cost of efficiency. Its time complexity of $O(n^5)$ is much greater than the other algorithms. With spline interpolation providing a pseudo-observation every four hours, a typical value for n is on the order of 25. The three-segment method takes about three minutes to do a single alignment. By contrast, the $O(n^2)$ DTW does the calculation in a fraction of a second. We would like our multisegment algorithm to be able to scale to handle queries for large databases of expression time series.

3. THE CONE HEURISTIC

We now describe the first of two new heuristics which address the efficiency limitation of our multisegment algorithm.

The alignment methods we use work by filling in an alignment matrix Γ. One well studied heuristic in similar time-series alignment problems is to restrict the cells of the matrix that are calculated. Several ways of doing this are illustrated in Figure 4. Each panel shows the alignment space when warping one series against another, and the shaded elements indicate the area to which the search is restricted. The so-called *Sakoe-Chiba Band*[4] and *Itakura Parallelogram*[11] are the most commonly used heuristics. The former restricts the search to a constant distance from the diagonal, while the latter allows progressively more warping closer to the middle. However both of these methods implicitly assume that the alignment being sought is a global one, in which there is no shorting of either input series. Here we consider a novel approach which confines the search to a cone starting at the origin and centered on the diagonal, as illustrated in Panel C of the figure.

Formally, we define the cone by a slope $c > 1$. We modify Equation 4 so that:

$$\gamma(i, x, y) = \textit{undefined} \textbf{ if } \frac{x}{y} > c \textbf{ or } \frac{y}{x} > c. \quad (6)$$

With this heuristic, we do not compute undefined values and we do not consider segments anchored in them.

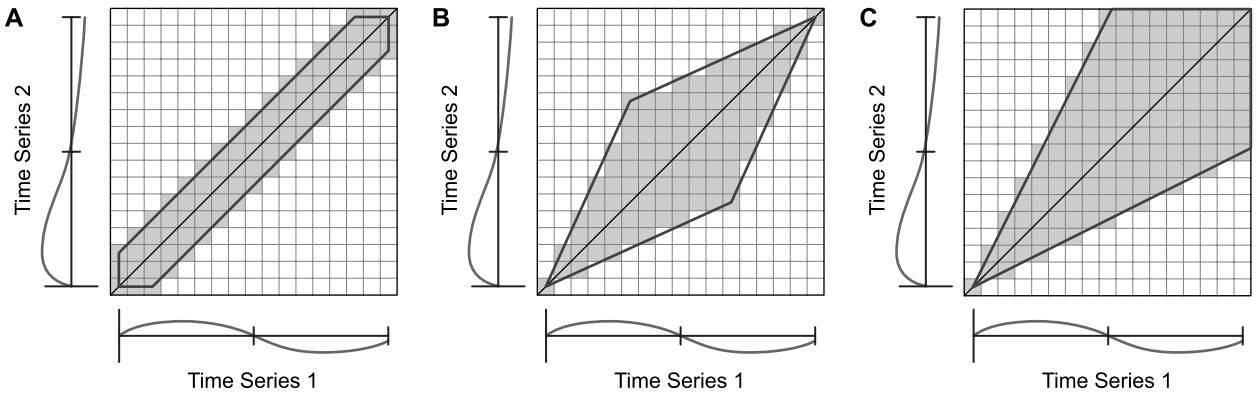

Fig. 4. Restricting the search in alignment space by shape. Each cell represents one element of the alignment matrix, and the shaded areas are those that are actually calculated. Both the Sakoe-Chiba Band (A) and the Itakura Parallelogram (B) are intended for globally aligning whole series to each other. By contrast, our cone (C) is designed for local alignments in which one series might be shorted.

3.1. Theoretical Analysis

Here we do a theoretical analysis of the expected speed-up in restricting the search space to a cone. The primary effect is to reduce the number of segments calculated by a constant factor, so we do not expect to see an improvement in its time complexity of $O(n^5)$. Instead we expect to see a constant speed-up proportional to the relative size of the cone to the alignment matrix. For a square matrix (i.e. where both series are of the same length), the relative size of the cone depends only on the slope of its bounding rays. With a slope of c, this ratio is $\frac{c-1}{c}$. We expect the execution time of the multisegment method with the cone heuristic to take roughly this fraction of the exact calculation's time.

We expect deviation from this value in two cases. First, nonsquare matrices will exhibit smaller ratios, as the cone covers proportionately less of their areas. Second, our calculation assumes that an alignment matrix can be split to an arbitrarily fine granularity. Because the matrix really consists of discrete elements, the ratio of elements covered by a cone will be more than expected for small matrices. For example, in a 5×5 matrix a cone with $c = 2$ will cover 13 cells, for a ratio of $\frac{13}{25}$ rather than $\frac{1}{2}$.

3.2. Experiments

Here we evaluate restricting the search in alignment space to a cone, in order to assess (i) its speedup relative to our original multisegment method, and (ii) its ability to find high-scoring alignments and produce accurate time-series classifications.

For the experiments in this section, we again use the data described in Section 2.3 and the same set of queries we used in Section 2.4.

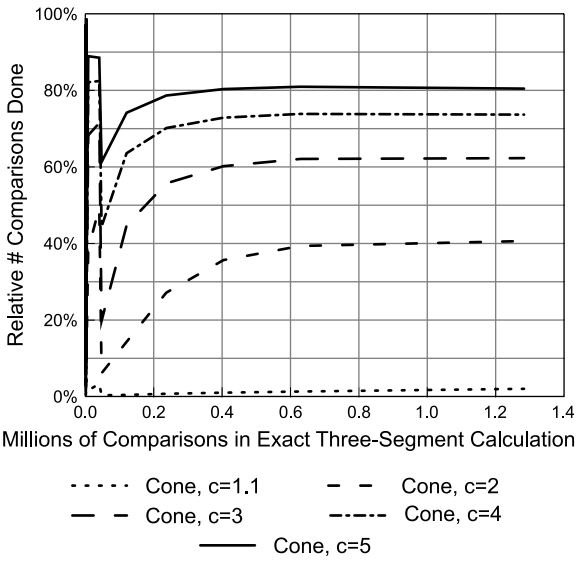

Fig. 5. Relative speed of the cone heuristic method. Time is measured by the number of comparisons of a point in each series.

First, we determine how much faster the cone method makes our calculations. We measure time in terms of *point comparisons*, or comparisons of a single time point in one series with a single time point in another. This is a good surrogate for calculation time

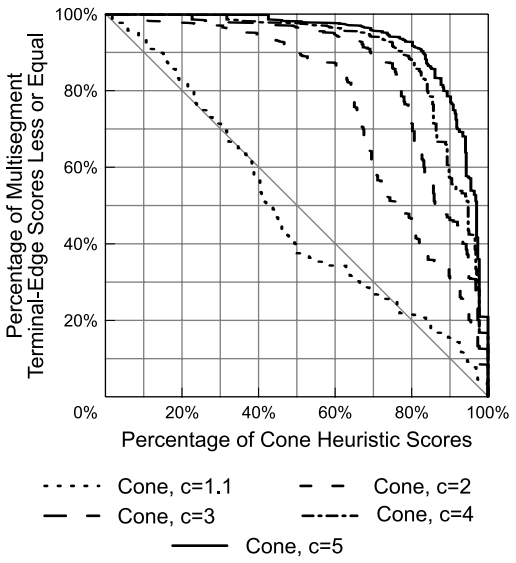

Fig. 6. Comparison of the cone heuristic method scores to the scores on the terminal edges of the alignment matrix of the exact multisegment calculation. These are the best alignments found for each legal shorting of the alignment.

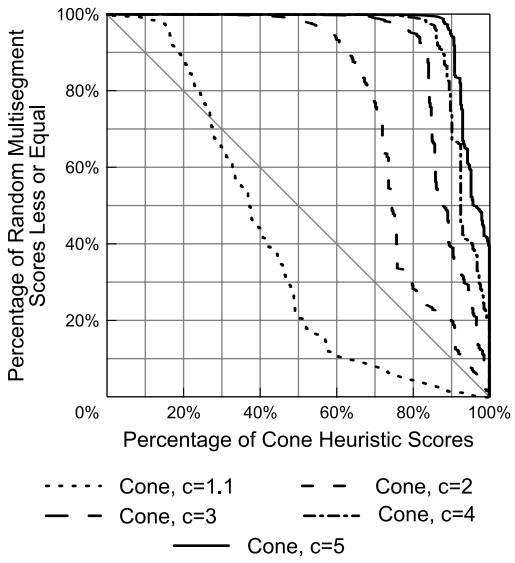

Fig. 7. Comparison of the cone heuristic method scores to the scores of 1000 randomly sampled multisegment alignments.

needed. Dynamic time warping performs $O(n^2)$ of these calculations, while our method performs $O(n)$ for each of $O(n^4)$ pairs of segments compared, for a total of $O(n^5)$. Figure 5 graphs the number of point comparisons done by the exact multisegment method versus the fraction done by the heuristic cone method, for the undistorted experiment we ran in Section 2.4. Although we obtain good speed-up, it is by no more than a constant factor. When the series are of roughly equal length, the time taken is in good agreement with our earlier calculated value of $\frac{c-1}{c}$. We predicted earlier that nonsquare matrices, which have less area covered by the cones, would run faster. These account for the dips visible in the figure. Additionally as predicted, smaller matrices tend to have larger values on this graph, because of the way cones divide the cells discretely.

Next we assess the alignments found using the cone heuristic, when we align each query to the database observations derived from the same treatment. We compare the score of each such alignment to a standard set of other alignments of the same query and database series. The first standard set we use consists of possible alignments found when doing the exact calculation. Recall that by Equation 5, the multisegment method chooses the best score from among all the possible shortings. We use all these scores—the best found for each shorting—as the standard set. Because the scores are drawn from the terminal edges of the alignment matrix, we refer to this set as the terminal-edge scores. We illustrate the comparison in Figure 6. Here we graph the percentage of heuristic-based scores that are better or equal to a percentage of the terminal-edge scores. For example, 80% of the cone-based scores are better or equal to at least roughly 70% of the edge scores when $c = 3$. If each query's score were drawn from the same distribution as its standard set, we would expect the curves to roughly coincide with the graphed diagonal. Thus with $c = 1.1$, the alignment found will likely not be better than picking one of the edge alignments at random. However with larger cones ($c \geq 2$), there is a good chance that the heuristic will lead to an alignment that scores well.

Figure 7 shows a similar comparison, but this time we compare against the scores from 1000 randomly sampled three-segment alignments as our standard set. These alignments are determined by randomly picking three contiguous segments in alignment space from origin to a terminal edge, and then picking the best amplitude coefficient for each segment by least squares. As before, when $c = 1.1$

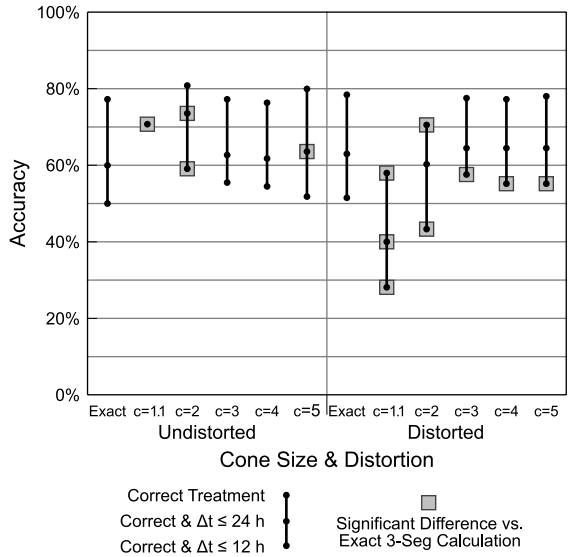

Fig. 8. Classification and alignment accuracies for the cone heuristic method with varying values of the slope parameter. Also shown for reference are the accuracy values for the exact three-segment method. Highlighted points are significantly different ($p \leq 0.05$) from the exact method by McNemar's χ^2 test.

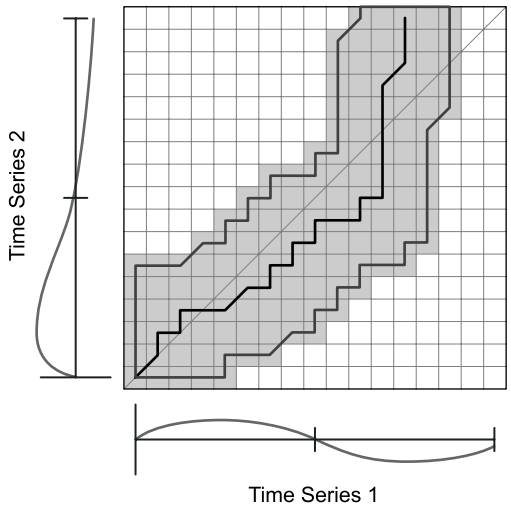

Fig. 9. Alignment space diagram of the hybrid-DTW heuristic for the multisegment method. We first find an alignment path using our hybrid-DTW method, and then we restrict the multisegment search to elements within a spread s of this path. Here s is two.

the alignment does not appear to be much better than random. With wider cones, however, the resulting alignments have better scores than most random alignments. Thus we conclude that, given a large enough cone, the segments that are not calculated are often not part of the optimal alignment. The alignment that is found by the cone heuristic will likely be comparable to the best alignment found by the exact method.

Finally we perform the classification/alignment task as in Section 2.4, except using the cone heuristic with the multisegment method. The results are shown in Figure 8. Except for the most narrow cones considered ($c = 1.1$), there is not a loss in accuracy due to finding suboptimal alignments. There may be some benefit to accuracy in using a moderate cone, with $c = 2$ or $c = 3$. If so, this is because such cones preclude the multisegment method from using extreme alignments, such as mapping the beginning of one series to the end of the other or using too great a slope in alignment space.

4. THE HYBRID-DTW HEURISTIC

Now we consider an alternative method for speeding up our multisegment alignment method. Here we restrict the search space by doing a first pass with a DTW-like method, and then considering only multisegment alignments that are close to the DTW path in alignment space. This method is illustrated in Figure 9.

We refer to the first pass method as *hybrid-DTW*, because it combines the dynamic programming of dynamic time warping with a scoring function similar to that used in our multisegment algorithm. The scoring function we use for it is:

$$D(q_x, d_y) = \begin{array}{l} D_E^2(q_x, \alpha d_y) + D_E^2(\frac{1}{\alpha} q_x, d_y) \\ -D_E^2(q_x, \mu_{DB,y}) - D_E^2(\mu_{DB,x}, d_y) \end{array} \quad (7)$$

where D_E is the Euclidean distance, α is a value chosen by least squares to minimize the first two terms, and $\mu_{DB,x}$ is the average value in the database for time x. Unlike classic DTW, any element $\gamma(x, y)$ can either add to or subtract from the final score. In DTW, the final score is a normalized sum, so it has a strong bias to reduce the number of elements on its alignment path. Our hybrid-DTW is able to avoid this bias.

Given the alignment path returned by the hybrid-DTW calculation, the second step of our approach restricts the search space of the exact multisegment calculation. We define the *spread* s to be the maximum distance from the hybrid-DTW path that we will search. We modify Equation 4 so that:

$$\gamma(i,x,y) = \begin{array}{l} undefined \\ \textbf{if } |x - x_h| > s \textbf{ or } |y - y_h| > s \end{array} \quad (8)$$

for all points (x_h, y_h) in the hybrid path. Thus we only consider segments that both begin and end within s of the hybrid path.

4.1. Theoretical Analysis

Like classic DTW, the time complexity of hybrid-DTW is $O(mn)$, where m and n are the lengths of the series being aligned. The maximum length of the path it returns is $m + n$. This gives us a maximum bound on the number of segments calculated for the multisegment method of:

$$\sum_{i=1}^{m+n} (i-1)(2s+1) = \frac{1}{2}((m+n)^2 - (m+n))(2s+1), \quad (9)$$

where s is the spread. Assuming that $s \ll m \approx n$, the number of segments considered is $O(n^2)$. We multiply this value by the $O(n)$ time required to calculate the score of a segment, and obtain a total time complexity of $O(n^3)$.

4.2. Experiments

As with the cone heuristic, we assess the hybrid-DTW heuristic by considering (i) its speedup relative to the original multisegment method, and (ii) the quality of the alignments it finds. Recall that we have interpolated pseudo-observations at four-hour intervals. Thus we evaluate these criteria with s ranging from zero elements (zero hours) up to four elements (16 hours).

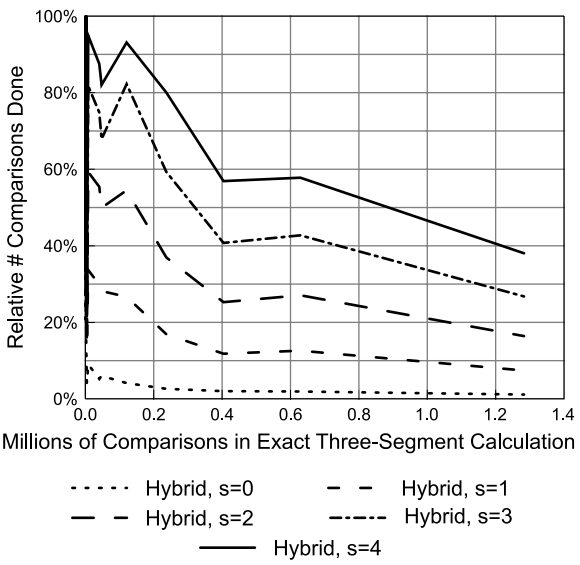

Fig. 10. Relative speed of the multisegment method with and without the hybrid-DTW heuristic. Again, time is measured by the number of comparisons of a point in each series.

Speedup is shown in Figure 10, which is again measured in terms of point comparisons. With $s = 0$ or $s = 1$ we obtain speedups of an order of magnitude. We again see the dips corresponding to non-square matrices. The best speed-ups occur for the largest matrix sizes. In contrast to the cone heuristic, the ratio of comparisons done still appears to be decreasing for the largest values. This is what we would expect with a better time complexity.

Next we consider the resulting alignment scores, and as before we compare them to the scores of both terminal-edge alignments and random alignments. The results of these score comparisons are shown in Figures 11 and 12. The hybrid-DTW heuristic method does well here, even when s is zero. Most of the scores found using this heuristic are better than or equal to the edge and random scores for the same query.

Finally, Figure 13 shows the classification and alignment accuracies for the exact multisegment calculation and the calculation using the hybrid-DTW first pass. There is no significant difference in accuracy when using the heuristic versus doing the exact multisegment calculation. For completeness, the figure also shows the accuracies when using the hybrid-DTW method by itself (i.e. not as a first pass for the multisegment method). Although it is more ro-

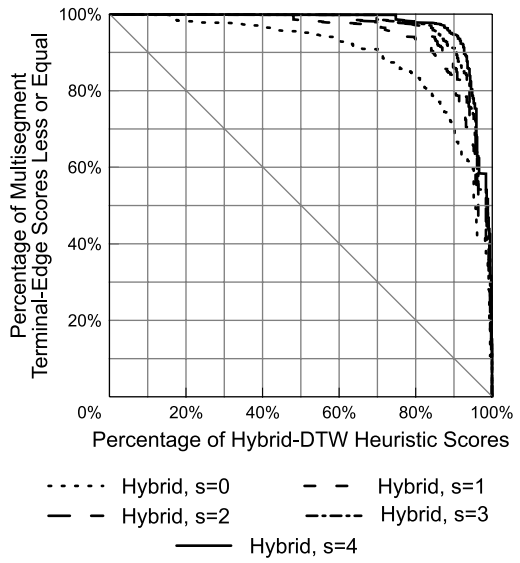

Fig. 11. Comparison of the hybrid-DTW heuristic scores to the scores on the terminal edges of the alignment matrix of the exact multisegment method.

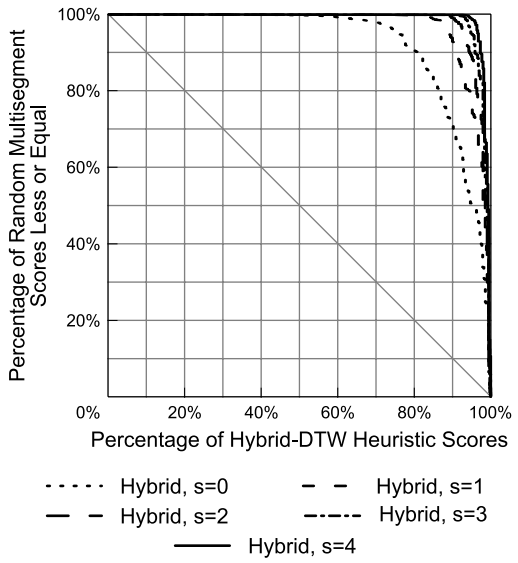

Fig. 12. Comparison of the hybrid-DTW heuristic scores to the scores of 1000 randomly sampled multisegment alignments.

bust to distortion than ordinary DTW and seems to align well, its accuracy—especially classification accuracy—is still significantly worse than that of the multisegment method.

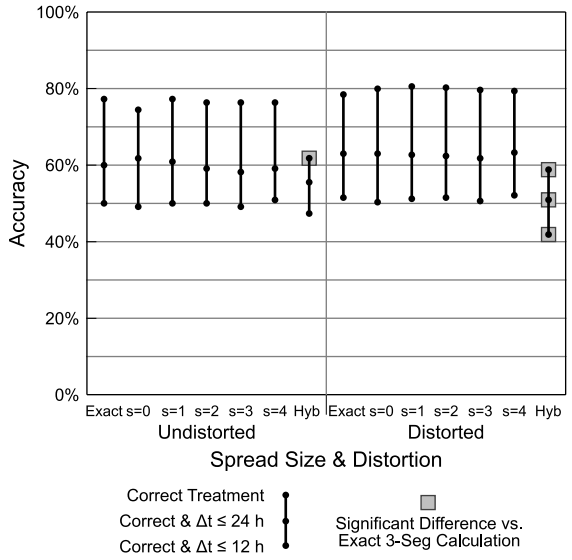

Fig. 13. Classification and alignment accuracies for the hybrid-DTW heuristic method with varying values of the spread parameter. Also shown for reference are the accuracy values for the exact three-segment method and the hybrid-DTW method on its own. As before, highlighted points are significantly different ($p \leq 0.05$) from the exact method by McNemar's χ^2 test.

Taken together, these results imply that the hybrid-DTW's alignment paths are a good approximation to those found by the multisegment method. Using spread values of zero or one has the potential to speed up the calculation greatly while not significantly harming the accuracy provided.

5. DISCUSSION

In a previous investigation[1] we showed that our multisegment alignment method is more accurate than other methods, both in terms of classification and alignment accuracy. In this study, we have presented two heuristics that can be used to speed up its calculation.

- By restricting the search space in the warping matrix to a cone, we may speed up the calculation by a constant factor. The cone may also serve as a useful bias, preventing alignments that are shorted too much when there is reason to believe that they should be excluded. The cone shape is analogous to both the Sakoe-Chiba Band and the Itakura Parallelogram, but it allows non-global alignments.
- By restricting the search space to alignments that are near those found by a modified DTW method,

we can improve the time complexity from $O(n^5)$ to $O(n^3)$.

Of the two heuristics, the hybrid-DTW based method offers clearly superior results. However the cone based heuristic is not without merit, as it seems to have somewhat of a regularization effect, biasing the alignments found toward more accurate ones. In future work we will explore using the two methods in conjunction.

Acknowledgments

This work was supported by NIH/NIEHS grant R01 ES012752, and NIH/NLM grant R01 LM07050. We would also like to thank Christopher Bradfield and Aaron Vollrath of the EDGE project.

References

1. Smith AA, Vollrath A, Bradfield C, Craven M. Similarity queries for temporal toxicogenomic expression profiles. *PLoS Computational Biology* 2008; In press.
2. Hayes K, Vollrath A, Zastrow G, McMillan B, Craven M, Jovanovich S, Walisser J, Rank D, Penn S, Reddy J, Thomas R, Bradfield C. EDGE: A centralized resource for the comparison, analysis and distribution of toxicogenomic information. *Molecular Pharmacology* 2005; **67**: 1360–1368.
3. Aach J, Church G. Aligning gene expression time series with time warping algorithms. *Bioinformatics* 2001; **17**: 495–508.
4. Sakoe H, Chiba S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE ASSP Magazine* 1978; **26**: 43–49.
5. Sankoff D, Kruskal J. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley 1983.
6. Criel J, Tsiporkova E. Gene time expression warper: A tool for alignment, template matching and visualization of gene expression time series. *Bioinformatics* 2006; **22**: 251–252.
7. Liu X, Müller HG. Modes and clustering for time-warped gene expression profile data. *Bioinformatics* 2003; **19**: 1937–1944.
8. Ratanamahatana C, Keogh EJ. Three myths about dynamic time warping data mining. In: *Proceedings of SIAM International Conference on Data Mining*. SIAM, 506–510.
9. Bar-Joseph Z, Gerber G, Gifford D, Jaakkola T, Simon I. Continuous representations of time-series expression data. *Journal of Computational Biology* 2003; **10**: 341–356.
10. Nielsen NV, Carstensen JM, Smedsgaard J. Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping. *Journal of Chromatography A* 1998: 17–35.
11. Itakura F. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 1975; **23**: 67–72.