Anal. Chem. **2004,** *76,* 404−411

Parametric Time Warping

Paul H. C. Eilers*

Department of Medical Statistics, Leiden University Medical Centre, P.O. Box 9604, 2300 RC Leiden, The Netherlands

A parametric model is proposed for the warping function when aligning chromatograms. A very fast and stable algorithm results that consumes little memory and avoids the artifacts of dynamic time warping. The parameters of the warping function are useful for quality control. They also are easily interpolated, allowing alignment of batches of chromatograms based on warping functions for a limited number of calibration samples.

In some analytical instruments one may encounter (time) warping problems: signals that are stretched or shrunken along the horizontal scale, relative to a reference signal. The prototypical example is chromatography. Because of aging of the column, chromatograms of the same chemical sample taken at different moments will not be exactly the same: corresponding peaks will not appear at the same elution times, because of changes in the speed at which a component moves through the column. This is not a large problem if peaks are identified by trained persons using visual inspection. But it is if the data are going to be used directly as input for multivariate statistical analysis like principal components analysis or curve resolution.

Time warping tries to solve these problems by transforming time (or, implicitly, elution speed) so that the peaks in two chromatograms get aligned precisely. Forshed et al.[1] presented an up-to-date overview of historical developments in warping, as applied to chemometrics and process control. The majority of the algorithms is based on dynamic programming, and they are known as dynamic time warping (DTW). Refinements based on correlations have been proposed.[2] The results are mostly of dubious quality: sudden jumps can occur, as well as plateaus. Also DTW consumes a lot of time and memory: for series with a length of thousands of samples, computation time has to be measured in minutes rather than seconds. The algorithm of Forshed et al.[1] is different: it uses a genetic algorithm to divide data into segments and apply a separate shift to each segment for alignment. It appears to be reasonably fast and not to consume large amounts of memory. Another improvement, for LC−MS data, is presented by Bylund et al.[3]

There have also been efforts to develop local shifting algorithms. Johnson et al.[4] searched for matching peaks in two chromatograms, while Fraga et al.[5] exploited the observed rank of two-dimensional data.

Surprisingly, it seems that there have been no efforts to model the warping function itself explicitly. In this paper, a fast and simple warping algorithm is presented, based on a parametric model for the warping function. Its performance is shown on a large set of experimental chromatograms. A novel property is that this approach allows easy interpolation of warping functions. This is useful to correct a whole batch of chromatograms on the basis of a limited series of warping function derived from calibration samples.

PARAMETRIC TIME WARPING

Suppose we are given samples of two series, $x_i = x(t_i)$ and $y_i = y(t_i)$, for $i = 1...m$. In most applications $t_i = ih$, with h a fixed sampling interval. For the algorithm to be presented, this is not necessary, but to simplify the presentation, we will assume this to hold. We seek a warping function $w(t)$ such that $x(w(t))$ is "near to" $y(t)$. A natural measure of nearness is the familiar integral of squares of differences:

$$S^* = \int_{t=h}^{m} [y(t) - x(w(t))]^2 \, \mathrm{d}t \tag{1}$$

This is not the only possible choice: in theory one might consider other measures like the integrated absolute differences. The quadratic norm here has the practical advantage of leading to linear regression equations.

As we work with sampled signals, we have to approximate this integral by a sum:

$$S = \sum_{i \in H} [y_i - \hat{x}(w(t_i))]^2 \tag{2}$$

where H indicates the set of indices i for which $\hat{x}(w(t_i))$ can be computed, i.e., $h \leq w(t_i) \leq mh$. We compute $\hat{x}(u)$, with $u = w(t)$, by linear interpolation:

$$\hat{x}(u) = x_j + (u - jh)/(x_{j+1} - x_j) \tag{3}$$

where $j = \lfloor u/h \rfloor$, the integer part of u/h.

* E-mail: p.eilers@lumc.nl.

[1] Forshed, J.; Schuppe-Koistinen, I.; Jocobsson, S. P. *Anal. Chim. Acta* **2003**, 487, 189−199.

[2] Vest Nielsen, N.-P.; Carstensen, J. M.; Smedsgaard, J. *J. Chromatogr., A* **1998**, *805*, 17−35.

[3] Bylund, D.; Danielsson, R.; Malmquist, G.; Markides, K. E. *J. Chromatogr., A* **2002**, *961*, 237−244.

[4] Johnson, K. J.; Wright, B. W.; Jarman, K. H.; Synovec, R. E. *J. Chromatogr., A* **2003**, *996*, 141−155.

[5] Fraga, C. G.; Prazen, B. J.; Synovec, R. E. *Anal. Chem.* **2000**, *72*, 4154−4162.

10.1021/ac034800e CCC: $27.50
Published on Web 12/09/2003

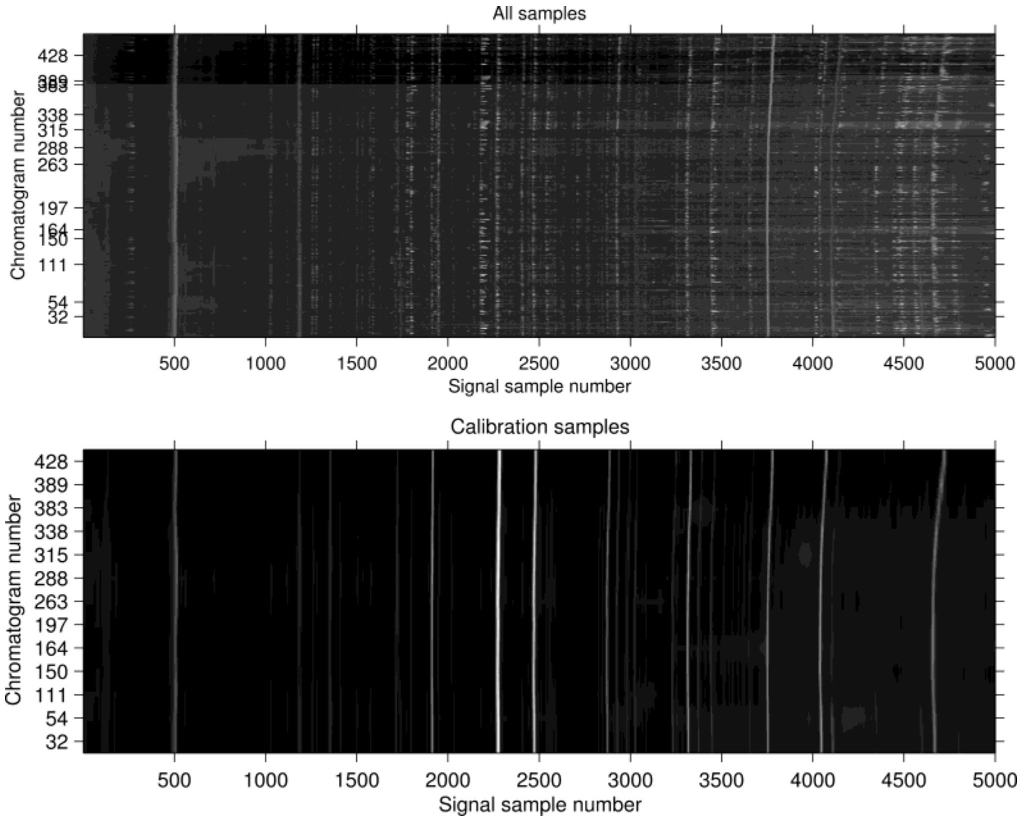

Figure 1. Upper panel: all chromatograms in a set of 460, displayed as an image. Brightness is proportional to the logarithm of the signal. The calibration samples are marked on the vertical axis. Lower panel: only the 13 chromatograms of the calibration samples.

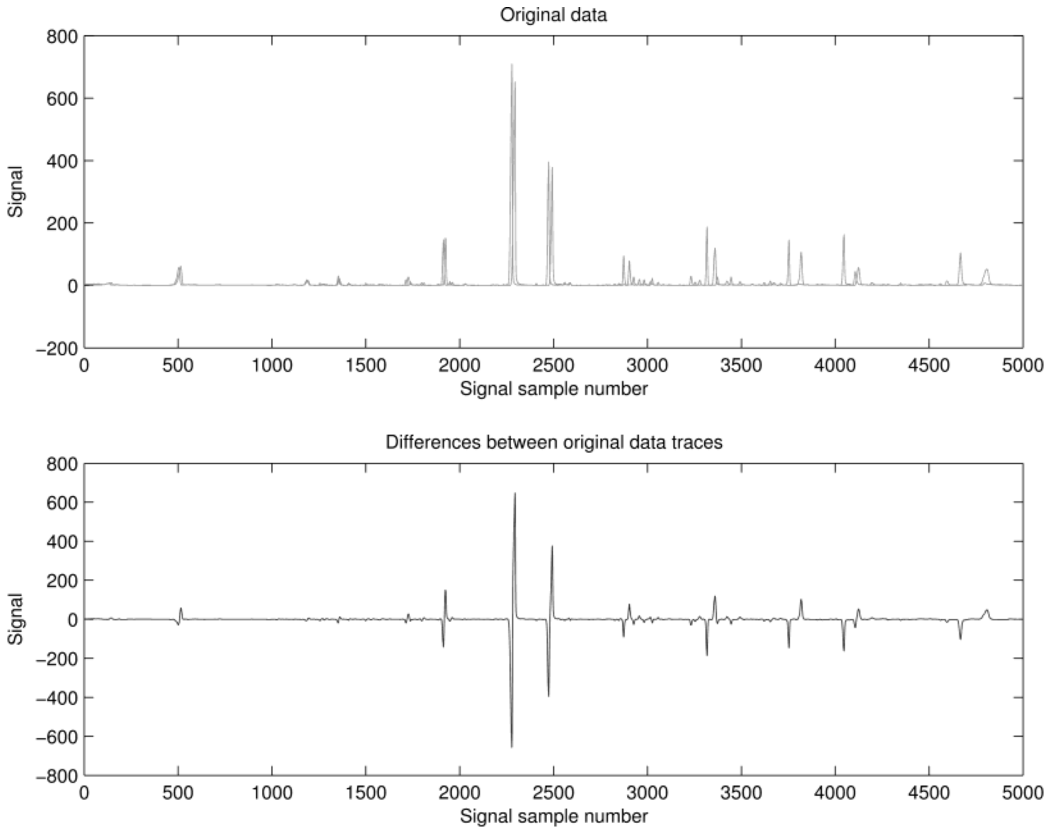

Figure 2. Upper panel: two calibration samples. Lower panel: difference of the two chromatograms.

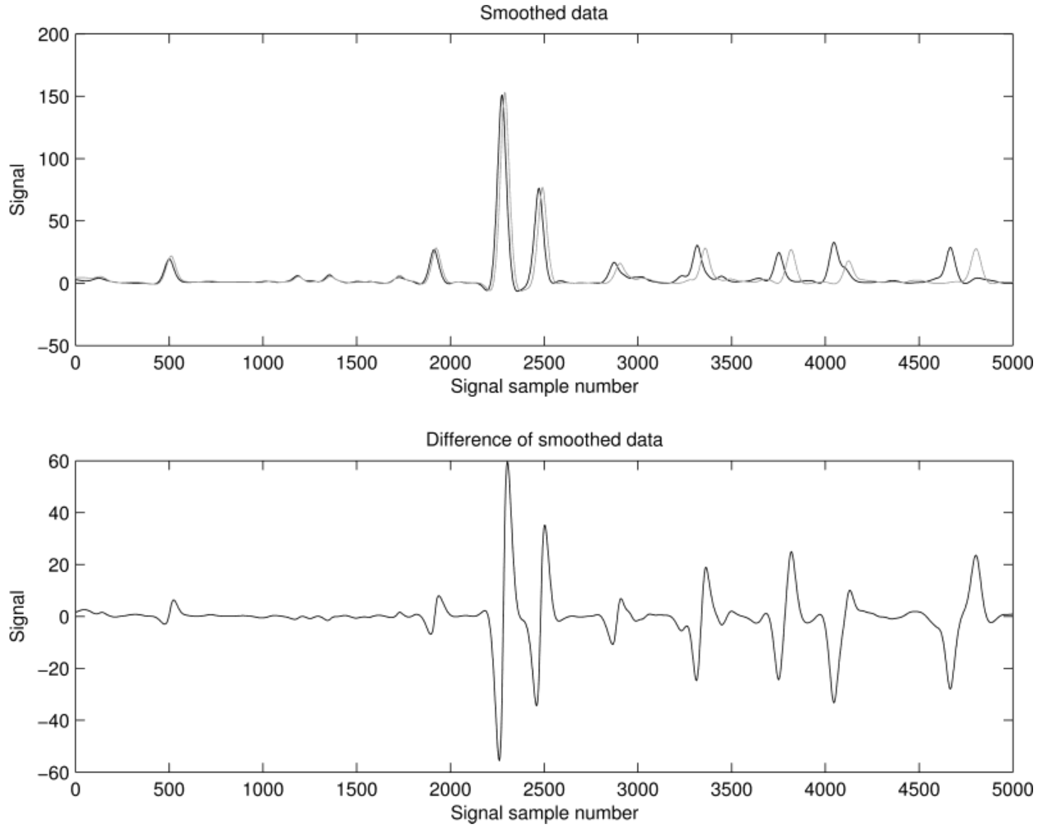

Figure 3. Illustration of broadening the peaks by heavy smoothing. Upper panel: the two calibration samples. Lower panel: difference of the two chromatograms.

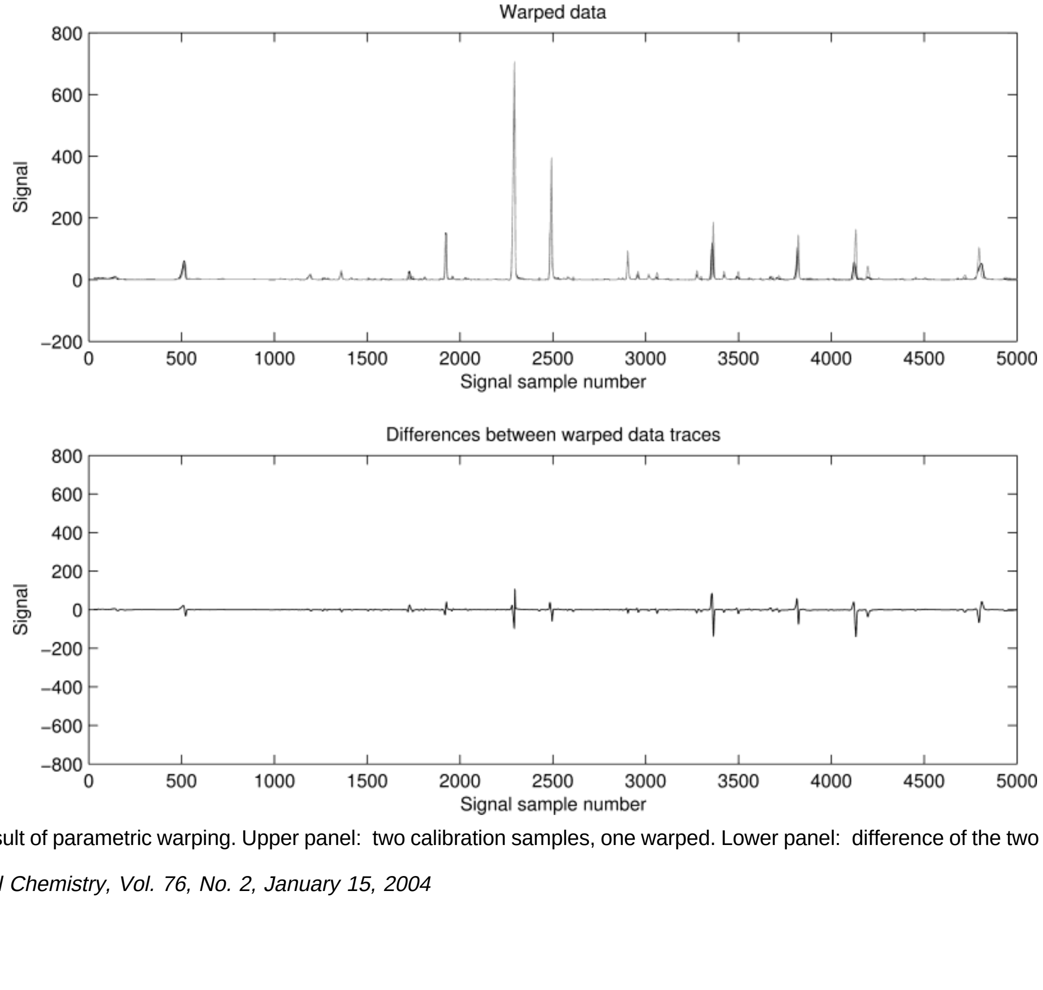

Figure 4. Result of parametric warping. Upper panel: two calibration samples, one warped. Lower panel: difference of the two chromatograms.

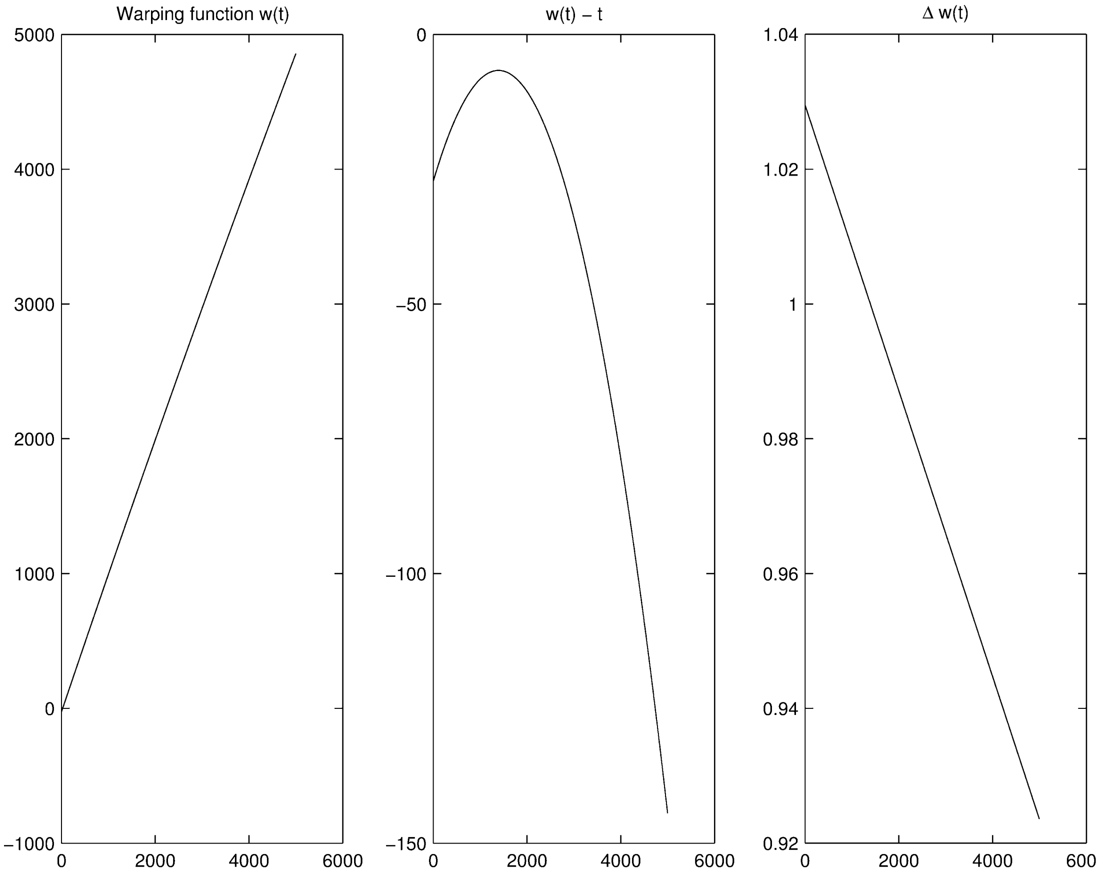

Figure 5. Several displays of the warping function. Left panel: the warping function itself. Middle panel: after subtraction of unwarped time, to emphasize the quadratic component. Right panel: differences of the warping function.

A simple model for the warping function is a polynomial of degree K in t:

$$w(t) = \sum_{k=0}^{K} a_k t^k \tag{4}$$

Experience has shown that $K = 2$ will be sufficient in many applications. If an approximation $\tilde{a}$ to the coefficient vector a is known, and $\tilde{w}(t) = \sum_{k=0}^{K} \tilde{a}_k t^k$, then the following Taylor expansion holds:

$$x(w(t)) = x(\tilde{w}(t) + \Delta w(t)) \approx x(\tilde{w}(t)) + x'(\tilde{w}(t))\Delta w(t) = x(\tilde{w}(t)) + \sum_{k=0}^{K} \Delta a_k x'(\tilde{w}(t))t^k \tag{5}$$

Combining (2) and (5), we find that we have to minimize the following approximation

$$S^* = \sum_{i \in H} [y_i - x(\tilde{w}(t_i)) - \sum_{k=0}^{K} x'(t_i) t_i^k]^2 \tag{6}$$

which is equivalent to regression of the residuals $y_i - x(\tilde{w}(t))$ on the set of basis functions $x'(t)t^k$ to find corrections Δa. We repeat this process until convergence.

Although it is perhaps not directly clear from the formulas, the two chromatograms can have widely different lengths (in time and in their number of samples). The Matlab implementation checks that the warping function is only computed for available samples of the template x (the set H in the summation in (2)).

Experience has shown that chromatograms with sharp peaks need a rather good approximation to the warping function to get successful convergence. This is a nuisance if we want an automatic algorithm. A simple solution is to artificially broaden the peaks by strong smoothing of the signals. The Whittaker smoother is used here,[6] but a running mean or any other smoother will work as well.

Two plots are useful to interpret a solution $w(t)$. One shows $w(t) - t$, as the difference between the "real" time of x and the transformed time of y. The other shows the differences $w(ih + h) - w(ih)$, which can be interpreted as the effective time steps between samples after warping. In the case of no warping, i.e., when $w(t) = t$, the difference will be h uniformly.

Baseline removal is important to get a sharply defined minimum of S. Especially if y and x have different baseline levels, there is a permanent contribution to the sum of differences. Experience has shown that asymmetric least-squares smoothing is very effective for chromatograms. It is described in the Appendix.

The chromatograms should also have approximately the same average value. This can be achieved simply by dividing them by

[6] Eilers, P. H. C. *Anal. Chem.*, in press.

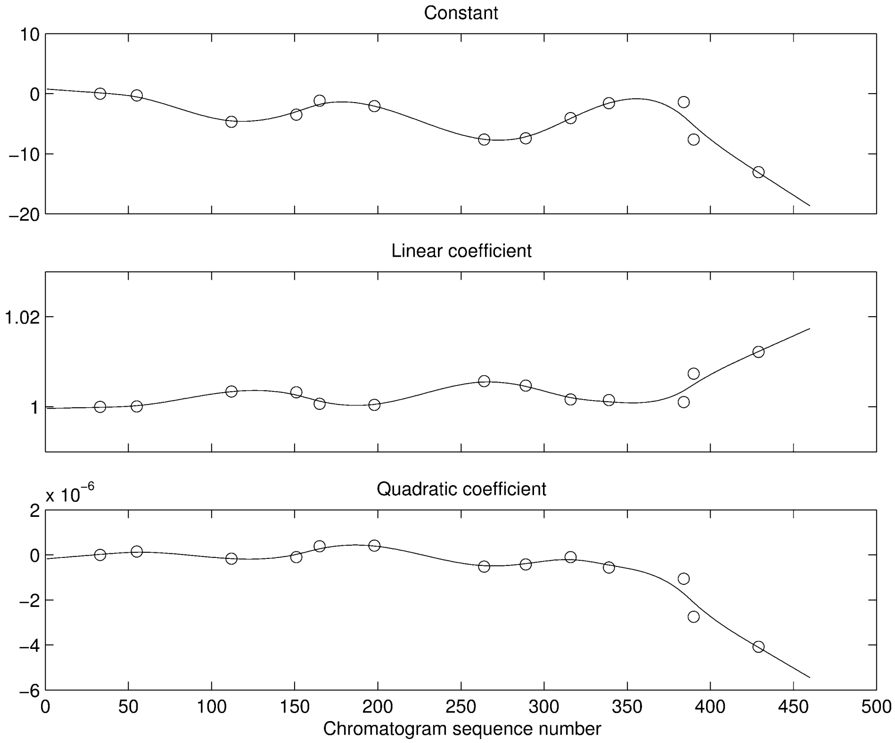

Figure 6. Coefficients of the warping functions, with smoothly interpolating curves. From top to bottom: for constant linear and quadratic component.

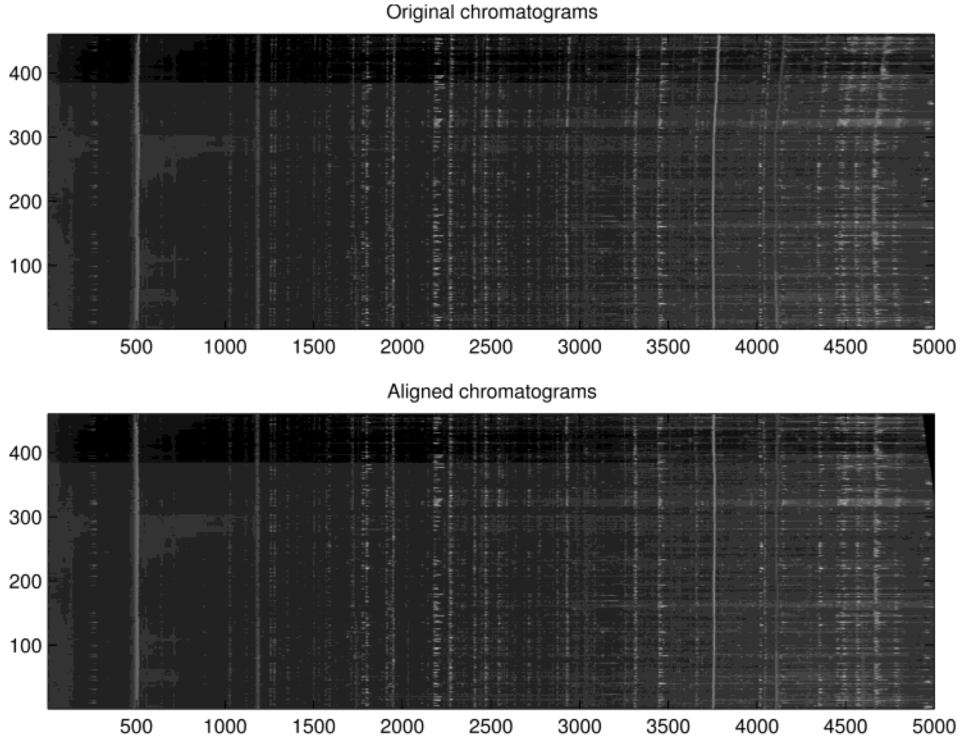

Figure 7. Gray scale representation of the results of aligning all chromatograms, based on smooth interpolation and extrapolation of the warping functions for 13 calibration chromatograms. Intensity is proportional to the logarithm of the signal. Top panel: before alignment. Bottom panel: after alignment.

their average values (after baseline removal).

A bundle of parametric warping functions can easily be interpolated. Assume that we have collected a series of chromatograms and that a number of them are calibration samples, collected in a matrix $Y = [y_{ij}]$ with n columns. If the first column of Y is the reference sample, and all other columns are warped to align them with the reference, we get n series of warping function coefficients, a_{kj}. The first of these is trivial, with $a_0 = a_2 = 0$ and $a_1 = 1$. Calibration sample j has "position" c_j in the series of all samples. Depending on the way the instrument was operated, c_j

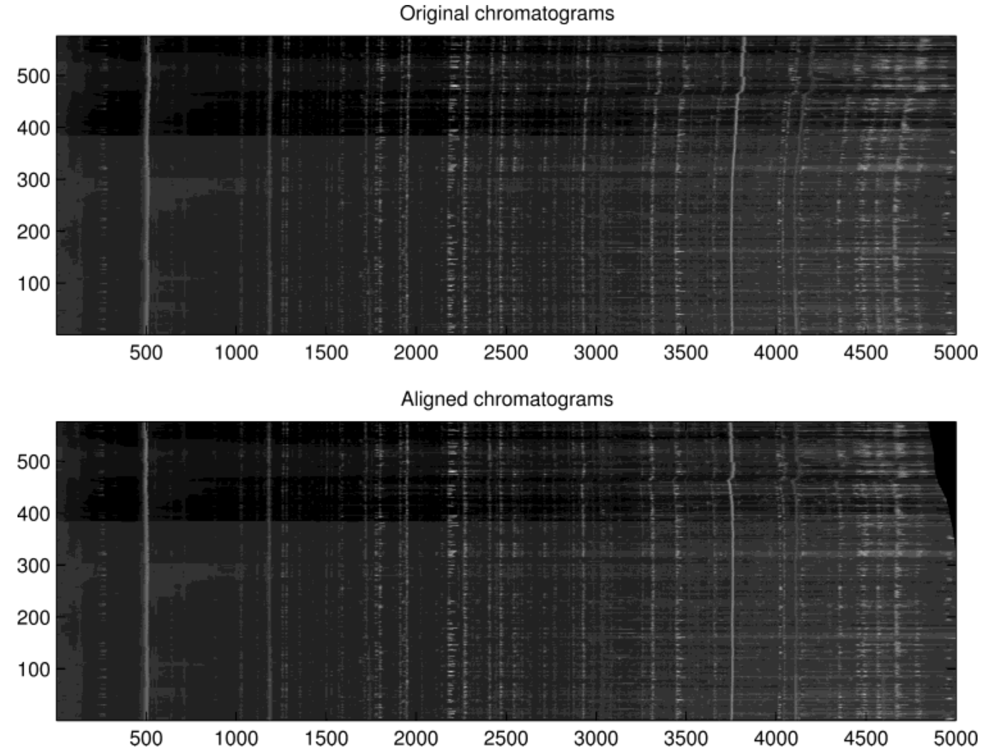

Figure 8. Gray scale representation of the results of aligning an extended set of chromatograms, based on smooth interpolation and extrapolation of the warping functions of 16 calibration chromatograms. Intensity is proportional to the logarithm of the signal. Top panel: before alignment. Bottom panel: after alignment.

might simply be the sequence number or reflect actual date and time of the analysis. Using (smooth) interpolation, we can compute estimated coefficients of the warping function at any desired position between the calibration samples. The Whittaker smoother[6] is suitable for this task. A plot of the coefficients is also useful for quality control.

One has to check if (where) smooth interpolation is applicable. Sometimes a sudden jump in the warping function might be needed. An example will be shown in the next section. A solution is to visually divide the data into episodes and to fit and interpolate warping function for each episode separately.

One can approach multiple alignment in two ways. If, as described above, the first calibration is used as the references, one has to invert the (interpolated) warping function to align the data. It also possible to work the other way around: take each new calibration sample as the "reference" and compute the warping function that aligns it with the first calibration sample. Interpolation of the bundle so obtained immediately gives the warping function for all intermediate data. One could argue that the first calibration sample is obtained under the most favorable circumstances and so is the logical choice for the reference. With strong aging, later calibration samples might also miss one or more peaks at the end. So the first approach seems the most logical one.

To invert of the warping function, a Newton−Raphson approach is suitable. Suppose that a series y and the corresponding warping $\hat{w}$ be given. Let $w_i = w(t_i) = \sum_{k=0}^{K} a_k\, t_i^k$. The goal is to find $\hat{t}$. The first derivative is $w'(t_i) = \sum_{k=0}^{K} k a_k\, t_i^k$. Let $\tilde{t} = (w - a_0)/a_1$ be a first approximation. An improved value will be obtained as $\tilde{t} + \Delta t$ with $\Delta t = (\hat{w} - w(\tilde{t}))/w'(\tilde{t})$. Generally only one correction is sufficient, but the computation can be repeated if necessary. Of course, $\hat{t}$ will generally not coincide with the sampling positions, so (linear) interpolation of y will be necessary.

AN APPLICATION

The application is based on data that were kindly provided by Claire Boucon and Sijmen de Jong of Unilever Research Vlaardingen. Of a series of 460 chromatograms, 13 were from calibration samples. Figure 1 gives an overview of the data and the calibration traces. Figure 2 shows two calibration traces before warping. To illustrate the heavy smoothing used for a reliable start, Figure 3 shows the smoothed data before warping. On a 1-GHz Pentium III computer, it take less than 1 s to estimate the warping function for two signals of 5000 samples.

The result of warping is shown in Figure 4. The warping function is shown in Figure 5. Notice that the warping function itself, as shown in the leftmost panel, is not very informative. The difference with unwarped time (middle panel) emphasizes the quadratic component and shows the reduced speed in the aged column. The differences in the rightmost panel show the ratios between time steps and so are a direct indicator of changes in elution speed.

Figure 6 shows the coefficients of the quadratic warping function plotted against their sequence number, together with the smooth interpolating (and extrapolating) function. The Whittaker smoother[4] was used for smooth interpolation. For each sequence number, three interpolated coefficients are available from which a warping function can be computed. After aligning all traces with these functions, the results look like Figure 7.

Actually these chromatograms are part of a larger set of 576, with 16 calibration samples. If we apply the interpolation procedure to all data, we get the result that is shown in Figure 8. Generally

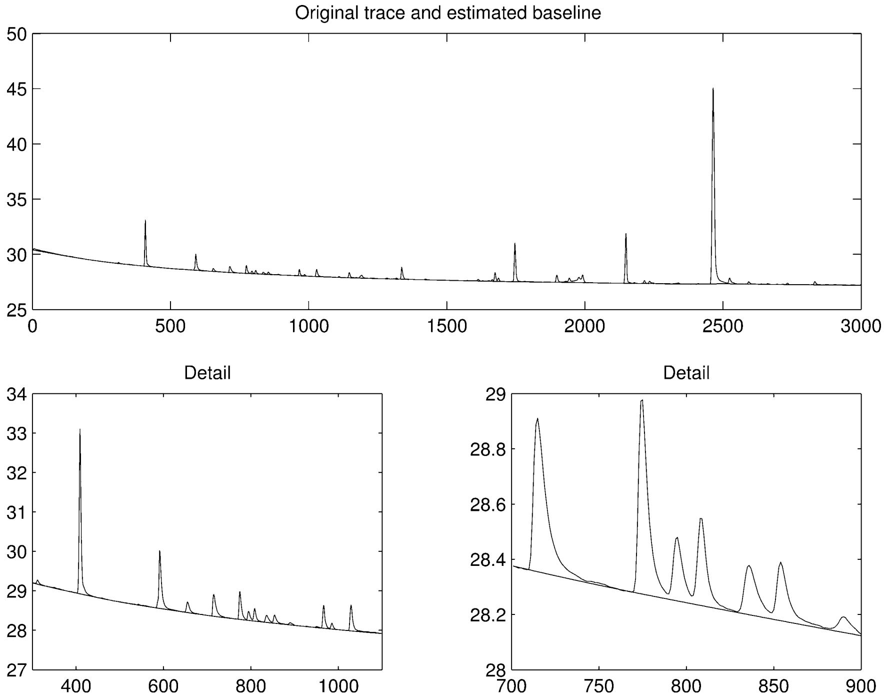

Figure 9. Baseline estimation with asymmetric least-squares smoothing ($p = 0.001$, $\lambda = 10^5$).

the algorithm does a good job, but there are some sudden changes near the end of the higher-numbered chromatograms that escape alignment with interpolation.

DISCUSSION

Parametric time warping is an extremely fast and effective method for aligning chromatographic traces. It can be implemented in one page of Matlab code, and even for a signal of 5000 samples the computations takes less than 1 s. No large matrices are used, so demands on memory are small. Because of the inherently smooth nature of the warping function, no artifacts like peak flattening do occur. Large stretches without peaks can do no harm either, because of the global nature of the warping function. Matlab functions, example scripts, and the calibration chromatograms are provided as Supporting Information.

A novel feature is the possibility to easily interpolate (the coefficients of) warping functions (obtained from calibration samples). The parameters of the warping functions, especially those of the quadratic function, might be useful for quality monitoring, because it provides a measure of nonuniform retardation. One can imagine a kind of control chart that is updated after each calibration sample, with limits that indicate when action will be needed. In this way, the amount of warping of an instrument could be discussed in quantitative way.

A simple parametric warping function will not always work. In an HPLC-DAD analysis of plant extracts, it was found that warping was nonuniform: speeding up and slowing down alternated over the course of the chromatograms. A semiparametric warping function, using P-splines[7] solved this problem. Details will be reported elsewhere.

Parametric time warping does not rely on any special features of the data series, such as matching peaks.[4] So it can also be suitable for signals with relatively few details, such as a single peak or even a (nearly) monotone growth or decay. It is also possible to eliminate parts of traces completely, using prior zero/one weights in (2). This can be a convenient way to eliminate disturbing artifacts.

The Supporting Information to this article contains some Matlab functions and example data so that the reader can easily experiment with the algorithm. Hans Boelens and Henk-Jan Ramakers (University of Amsterdam) developed a graphical user interface to algorithms for dynamic and parametric time warping; it can be downloaded from http://www.tipb.nl.

APPENDIX: ASYMMETRIC LEAST-SQUARES BASELINE ESTIMATION

A strong baseline, especially when it changes over time, can seriously jeopardize the performance of the algorithm. In principle it might occur that the warping function tries to match the baselines instead of the interesting part of the chromatogram. A simple but effective and completely automatic algorithm for baseline estimation was developed in cooperation with Hans Boelens (University of Amsterdam). It uses asymmetric least squares (ALS). ALS was used for hypothesis testing by Newey and Powell[8] and for exploratory data analysis by Eilers.[9]

The Whittaker smoother for discrete (time) series,[6] minimizes the function

$$Q = \sum_i v_i (y_i - f_i)^2 + \lambda \sum_i (\Delta^2 f_i)^2 \tag{7}$$

where y gives the data, f a smooth trend, and v prior weights. The elements of v are 1 in all places where y is observed or allowed

[7] Eilers, P. H. C.; Marx, B. D. *Stat. Sci.* **1996**, *11*, 89−121.

[8] Newey, W. K.; Powell, J. L. *Econometrica* **1987**, *55*, 819−847.

[9] Eilers, P. H. C. *Kwant. Methoden* **1988**, *11*, 45−62.

to influence f. In all other places, the elements of v are 0; there smoothly interpolated values will be computed for f. The (positive) parameter λ sets the weight of the second term, the roughness penalty; the larger λ, the smoother f will be.

Consider the following choice of asymmetric weights: $v_i = p$ if $y_i > f_i$, and $v_i = 1 - p$ if $y_i \leq f_i$, with $0 < p < 1$. So positive deviations from the trend will get weights different from negative residuals. There is an chicken-and-egg problem: given the weights v, the computation of f is straightforward, and given f, the computation of v is trivial. But how to find them both? Experience shows that starting from $v \equiv 1$ and iterating between the two computations quickly and reliably leads to a solution in about 10 iterations. One can show[9] that this is a gradient algorithm. Because the function Q is convex in f (personal communication with Patrick Groenen, Erasmus University, Rotterdam), convergence is theoretically assured.

With p near zero and rather large λ, f will tend to follow the valleys of y. This is exactly what we like to see in a baseline estimate. Figure 9 shows an example with a rather dramatic baseline. Experience has shown that this algorithm, using visual inspection to choose the parameters p and λ is effective and fast. Optimal choices of λ and p from the data are presently investigated and the results will be reported elsewhere.

SUPPORTING INFORMATION AVAILABLE

Additional information as noted in the text. This material is available free of charge via the Internet at http://pubs.acs.org.

Received for review July 15, 2003. Accepted October 11, 2003.

AC034800E