Variable Penalty Dynamic Time Warping Code for Aligning Mass Spectrometry Chromatograms in R

David Clifford
CSIRO

Glenn Stone
University of Western Sydney

Abstract

Alignment of mass spectrometry (MS) chromatograms is sometimes required prior to sample comparison and data analysis. Without alignment, direct comparison of chromatograms would lead to inaccurate results. We demonstrate a new method for computing a high quality alignment of full length MS chromatograms using variable penalty dynamic time warping. This method aligns signals using local linear shifts without excessive warping that can alter the shape (and area) of chromatogram peaks. The software is available as the R package **VPdtw** on the Comprehensive R Archive Network and we highlight how one can use this package here.

Keywords: DTW, alignment, GC-MS, LC-MS, dilation, chromatogram.

1. Introduction

Dynamic time warping (DTW) is a fast and efficient means for aligning two signals. DTW was developed for speech recognition (Vintsyuk 1968; Sakoe and Chiba 1971) and has been successfully applied to chromatograms (Wang and Isenhour 1987). In its simplest form, dynamic time warping uses a dynamic programming algorithm to minimise the distance between two signals. This can be done symmetrically when both signals are warped to achieve alignment, or asymmetrically when one signal (the query) is warped to align it with another signal (the reference signal).

Asymmetric DTW aligns a query signal q to a reference signal r as follows. Suppose we have already aligned intensity $q(t_i)$ of the query to intensity $r(t_j)$ of the reference. Asymmetric DTW chooses one of the following three moves:

- align $q(t_{i+1})$ to $r(t_{j+1})$ (called a unit slope diagonal move);
- stretch the query signal by repeating $q(t_i)$ and aligning it with $r(t_{j+1})$ (expansion);

- contract the query by dropping $q(t_{i+1})$ and aligning $q(t_{i+2})$ with $r(t_{j+1})$ (contraction).

The goal of alignment is to minimise the cost of the warping, usually defined as a distance between the warped query and the reference.

$$\text{Cost}(w) = \sum_{i=1}^n d(r(t_i), q(w(t_i)))$$

More explicitly, we minimise the L_1 distance where $d(r(t_i), q(w(t_i))) = |r(t_i) - q(w(t_i))|$ over all possible warping paths w . There are many variations of DTW in terms of the number and types of moves that are permissible and cost functions or distance metric used. There are also memory efficient variations of DTW that limit the number of possible alignments by reducing the limit on the size of shifts that are possible overall. One such method is the Sakoe-Chiba DTW (Sakoe and Chiba 1971). The **dtw** package (Giorgino 2009; Tormene *et al.* 2008) on the Comprehensive R Archive Network (CRAN) implements many variations of DTW within R (R Development Core Team 2012) including the Sakoe-Chiba DTW.

The process of expanding and contracting the time axis of a query signal can produce a very high quality alignment to a reference signal. However, excessive numbers of expansions and/or contractions can often change the shape and area of chromatogram peaks in such a way that the resulting aligned peaks are distorted and appear artificial. Penalised dynamic time warping uses a penalty to constrain the use of expansions and/or contractions.

Implicit in this discussion is the fact that we are aligning two signals with similar lengths. In this respect, the unpenalised move at any point is the diagonal move with unit slope. If one is interested in matching two signals of very different lengths then such a warping will require many expansions / contractions of one signal to match the other and the predominant move will need to be an expansion / contraction as appropriate. In short, such signals will not be successfully matched using the technology discussed here.

Penalized DTW has been used in various applications, including detection of low-rate TCP attacks in Internet traffic (Sun *et al.* 2006) and alignment of somatosensory evoked potentials in neurology (Roberts *et al.* 1987). In both of these applications a constant penalty is added to the optimization function when an expansion or contraction move is taken by DTW. The optimization cost function then becomes

$$\text{Cost}(w) = \sum_{i=1}^n |r(t_i) - q(w(t_i))| + \lambda \sum_{i=1}^n I(i\text{-th move is an expansion or contraction})$$

where I is the indicator function. More advanced optimization cost functions using signal derivatives are considered in other applications as a way of improving the alignment results (Keogh and Pazzani 2001). Adaptively weighted DTW has been used to align short melody roots from Hungarian folk songs (Juhász 2007).

In Clifford *et al.* (2009) we introduced a variable penalty into the DTW process which we added to the distance metric when the signal is expanded or contracted. We refer to this method as variable penalty dynamic time warping (VPdtw). In Clifford *et al.* (2009), an L_2 norm was used and penalties were not doubled when contractions were used. In an improved version of this approach the optimization cost function becomes

$$\text{Cost}(w) = \sum_{i=1}^n |r(t_i) - q(w(t_i))| + \sum_{i=1}^n \lambda_i I(i\text{-th move is expansion}) \quad (1)$$

$$+ \sum_{i=1}^n 2\lambda_i \mathbf{I}(i\text{-th move is contraction}) \quad (2)$$

This form of VPdtw (Equation 2) is updated from the version described in Clifford *et al.* (2009) where the L_2 norm was used and penalties were not doubled when contractions were used. The switch to an L_1 norm makes this method less sensitive to the large dynamic ranges one finds with the intensity values of mass spectrometry data. Doubling the cost of contractions is a somewhat arbitrary choice that is found to improve performance in practice.

Adding a variable penalty λ_i to the optimization function is similar to rescaling the optimization function by adaptive weights when the latter is transformed using logs, but our approach adds the variable penalty only when expansion or contraction moves are taken. Also, this variable penalty DTW is shown here with an application to chromatograms. These signals are much longer and have a larger dynamic range compared with the melody roots used by Juhász (2007).

VPdtw was developed for aligning gas chromatography mass spectrometry (GC-MS) data and first used in a biomarker discovery related to obesity (Clifford *et al.* 2007, 2009). VPdtw has also been used for aligning GC-MS data in other application areas including the creation of meat quality indicators (Watkins *et al.* 2010) and prediction of wine origin (Berna *et al.* 2009). It is also applicable to other forms of chromatography including liquid chromatography mass spectrometry (LC-MS).

Other methods for aligning full signal chromatography data exist including parametric time warping (Eilers 2004; Bloemberg *et al.* 2010) in the **ptw** package (Gerretzen *et al.* 2010), other spline based techniques (Daszykowski *et al.* 2010), wavelet based techniques (Zhang *et al.* 2012) and correlation optimized warping (Nielsen *et al.* 1998; Christin *et al.* 2008). Another quite different approach to tackling this problem involves first picking peaks from chromatograms before clustering the peaks together and aligning the signals, (America *et al.* 2006; Robinson *et al.* 2007; Smith *et al.* 2006; Voss *et al.* 2011). Giskeodegard *et al.* (2010) compared several of these methods in the context of alignment of magnetic resonance spectra.

In this paper we highlight the uses of VPdtw using the implementation found in the **VPdtw** package. In Section 2 we outline the basic functions of **VPdtw**. Following this we look at some examples in Section 3 and compare **VPdtw** with **ptw** and **dtw** in terms of quality of alignment and speed of computation in Section 4.

2. About the VPdtw package

2.1. Package details and functions

The **VPdtw** package is available from CRAN at <http://CRAN.R-project.org/package=VPdtw> under a GPL-2 license. The alignment code has the efficiency of banded dynamic time warping $O(bn)$ where b is the width of the band and n is the length of the signal. The code is written in C++ and dynamically loaded into R to avoid nested for-loops in R and achieve fast optimisation of the warping path.

The principal function **VPdtw** performs a Sakoe-Chiba dynamic time warping with variable penalty for expansion or contraction moves using the L_1 norm to measure distance between signals. At its most basic, **VPdtw** takes a reference and query as input and aligns the query

to the reference. Other arguments passed include a penalty vector, and the width of the Sakoe-Chiba window size. The penalty is a vector whose length is equal to that of the signal and by default the values of the penalty are zero giving the usual Sakoe-Chiba DTW.

The versions of R and **VPdtw** discussed here are as follows:

```
R> version$version.string
[1] "R version 2.13.1 (2011-07-08)"

R> library("VPdtw")
R> packageDescription("VPdtw")$Version
[1] "2.1-9"

R> objects(2)
[1] "dilation"      "plot.VPdtw"    "print.VPdtw"   "VPdtw"
```

VPdtw produces an object of class **VPdtw** and functions **plot.VPdtw** and **print.VPdtw** summarise each alignment. A sample reference and query are also provided. These signals have lengths on the order of 10000 each. The output from **VPdtw** is a list including the reference, the warped query, the shift required to achieve the alignment and x-values for plotting the results.

2.2. Data supplied with VPdtw

Reference and query GC-MS signals from wine samples are provided with this package (Berna *et al.* 2009). In the examples here, we apply a log transformation to the chromatogram intensities prior to alignment. We work on the log scale due to the large dynamic range of the chromatogram signals. Figure 1 shows two chromatograms which have a large dynamic range after log-transform.

Figure 1: Reference and Query signals from GC-MS data collected from wine samples. The signals are plotted before alignment.

Figure 2: Dilation of the reference intensity values plotted together with the reference for comparison. We recommend using a penalty that is proportional to the dilation.

```
R> data("query")
R> data("reference")
R> reference <- log(reference)
R> query <- log(query)
```

2.3. Dilation and the penalty

A dilation of the reference signal r over a window length of $2k + 1$ is defined by

$$\text{dilation}(r(t_i), k) = \max_{i-k \leq j \leq i+k} \{r(t_j)\}.$$

We recommend using a penalty that is proportional to a dilation of the reference signal

$$\lambda_i \propto \text{dilation}(r(t_i), k)$$

for some value of k . The choice of k and the constant of proportionality are subjective and left to the user at this point but research on objective choices for the λ are ongoing. An external cross-validation scheme may be useful for optimization of these parameters. **VPdtw** can be used to compare the results of using different penalties for aligning a query to a reference.

A function **dilation** is provided to compute such penalties. Dilations are often used in mathematical morphology in the analysis of images (Soille 1999). See Figure 2 for an example of the result of the **dilation** function.

2.4. Ways of using VPdtw

One can also pass more than one penalty to see how different penalties produce different results. Our first example illustrates this aspect of the code. There, a penalty matrix is passed to **VPdtw** and each column of the penalty matrix is used to produce an alignment. Plotting the results of such a call to **VPdtw** show all alignments at once.

One can also pass more than one query to **VPdtw**. In this setting only one penalty should be passed. Simply pass a query matrix and each column of the matrix is aligned to the reference using the same penalty (but each with its own alignment). This is highlighted in our second example.

2.5. Choice of reference

Often choosing a reference sample is not straightforward. In some problems the reference exists naturally but often we are presented with a collection of query signals only. In the arguments for **VPdtw** if **reference** = **NULL** then by default **VPdtw** chooses one of the query signals by simple random sampling to serve as the reference. Other times one may not wish to use one query as a reference, instead using a pointwise mean or median of the queries as the reference. A **Reference.type** argument to **VPdtw** can be set to **"median"**, **"mean"** or **"trimmed"** (mean) but one needs queries that are not badly mis-aligned to use these options properly otherwise the shapes of peaks in the reference will be broader than they are in individual queries.

2.6. Background correction

A background correction functionality is not provided by the **VPdtw** package as we have made no advances to this aspect of the analysis of chromatogram data. Choosing to do a baseline correction will have an effect on the kind of penalty required to achieve a good alignment. As such, one should choose a penalty with baseline corrected data in mind if that is how one plans to analyse the data. Likewise, should the user decide not to log-transform the intensities this should be kept in mind when the penalties are chosen.

3. Examples using VPdtw

3.1. Example 1: Alignment via VPdtw

In this first example we use **VPdtw** to align the query to the reference using two penalties. The first penalty is a zero penalty and the second is based on a dilation of the reference. The syntax for computing both of these alignments is straightforward and the two calls to **plot** show the results of the alignments (Figure 3) and the shifts required (Figure 5) to achieve those alignments.

This summary of the alignments shows the total number of expansion and contraction moves, the maximum observed shift (with a warning printed to the screen if this exceeds three quarters of the maximum allowed shift). The cost of the alignment (which includes the penalty) is also outputted to the screen.

```
R> penalty <- rep(0, sum(!is.na(reference)))
R> penalty <- cbind(penalty, dilation(reference, 350)/5)
R> result <- VPdtw(reference, query, penalty = penalty, maxshift = 350)
R> print(result)
```

Reference supplied by user.

Query vector of length 10018.

Figure 3: Results of VPdtw alignment using a zero penalty and using 20% of a dilation as penalty. Compare the signals shown here with the original signal shown in Figure 1.

Figure 4: The pointwise difference between the reference and query after alignment using two different penalties.

Penalty matrix made up of 2 penalties supplied by user.
Max allowed shift is 350.

	Cost	Overlap	Max Obs	Shift #	Diag	Moves #	Expanded #	Dropped
Penalty #1:	682.4	9617		239		2488	4173	3750
Penalty #2:	2372.9	9617		128		9780	0	104

```
R> plot(result, "After")
```

From the output we can see a dramatic reduction in the number of expansions and contractions under penalty #2 compared with penalty #1.

Figure 5: Comparison of the warping paths for our two penalties.

Figure 6: Detail before alignment.

In Figure 3 it is not possible to directly compare the two alignments with each other, but overall we see that both alignments appear to be very good compared with the raw data in Figure 1.

Figure 4 shows the difference between the query and reference after alignment using the two penalties. In Figure 5 we can compare the warping paths associated with each penalty, this plot can be created with the command:

```
R> plot(result, "Shift")
```

Under penalty #1, when we do not penalise expansion or contraction moves the warping path is much more complex compared with the path found when a non-zero penalty vector is used. For the first half of the signal (up to index 5000) this additional complexity does

Figure 7: Detail after alignment. Note the peak distortion in the red signal compared with the absence of distortion in the green signal.

Figure 8: Detail of difference between reference and query after alignment.

not improve the final alignment. For the latter half of the signals, the two warping paths are quite different.

```
R> plot(result, xlim = c(800, 1600), type = "Before")
```

A more detailed examination of the quality of the alignment can be made using Figure 6 and Figure 7 which shows a subsection of the data before and after alignment. We can see clear differences in the final aligned signals in Figure 7. We can see evidence of distortion of the peaks when a penalty term is not used in the dynamic time warping. Figure 8 confirms that the differences between the reference and query signals are often removed entirely when no penalty is used in the alignment. One needs to choose a penalty that prevents this overwarping

but also allows enough warping to give good alignment across the full signal.

Additional examples of the artificial features introduced by unpenalised or improperly penalised dynamic time warping can be found in [Clifford *et al.* \(2009\)](#).

3.2. Example 2: Aligning many query signals at the same time

In this section we use the **gaschrom** dataset provided as part of the **ptw** package. This dataset is made up of sixteen GC signals each of length 5000. No reference is provided with the dataset but our aim is to align the signals to each other somehow. Figure 9 shows an image of the signals, each signal corresponding to a row of the image. The raw data appears to be very well aligned on the left hand side of Figure 9 but there is a need for aligning the last few peaks on the right hand side. As before we do the alignment on the log intensity scale.

```
R> data("gaschrom", package = "ptw")
R> query <- log(t(gaschrom))
R> image(1:nrow(query), 1:ncol(query), query, xlab = "Index",
+       ylab = "Sample", main = "gaschrom data")
R> box()
```

When we try to align these signals we need to decide on a penalty and a reference. Our strategy in the code that follows is to figure out the appropriate penalty term first. From Figure 9 it is clear that aligning the first and last sequence in the **gaschrom** data will require the largest shift. We use these two signals to investigate what kind of penalty is required. We specify four different penalties that are proportional to a dilation of the median signal and each successive penalty is lower than the preceding one. This kind of exploratory analysis is necessary to set penalty terms for the alignment of all signals in this example.

Figure 9: This is an image of the **gaschrom** data from the **ptw** package. The curvature in the lines shows that these signals need to be aligned.

Figure 10: A plot of the results of aligning one query to a reference signal using four different penalties. Clear differences in the results can be seen at the three large peaks above index 3500.

```
R> penalty <- dilation(apply(query, 1, median), 150)
R> penalty <- cbind(penalty/3, penalty/4, penalty/5, penalty/6)
R> result0 <- VPdtw(reference = query[, 1], query = query[,
+   ncol(query)], penalty = penalty, maxshift = 150)
```

Warning: Observed shift more than three quarters of maxshift

```
R> plot(result0, "After")
```

Figure 10 shows the resulting alignments. Penalty #1 fails to align the three large peaks to the right of index 3500 to the corresponding peaks in the reference sample. Penalties #2 and #3 manage to align two of the three peaks. Only penalty #4 aligns all three of these peaks. The other three penalties are too large and prevent these peaks from aligning properly. We proceed with our alignment using penalty #4. Figure 11 shows the difference between the reference and each of the four warped queries.

Having found an appropriate penalty we use it in the next stage to align all sixteen signals. When no reference is specified `VPdtw` chooses one of the query signals to be the reference by simple random sampling. The results of this alignment can be seen in Figure 12. Noting the vertical bands in Figure 12 across all samples we conclude that the alignment was a success.

```
R> penalty <- dilation(apply(query, 1, median), 150)/6
R> result1 <- VPdtw(reference = NULL, query = query, penalty = penalty,
+   maxshift = 150)
R> plot(result1, "After")
```

Figure 11: A plot of the difference between the reference and warped query associated with each of the four penalties.

Figure 12: Results of aligning the `gaschrom` data. Straight vertical lines in this figure indicate a high-quality alignment. Compare to Figure 9.

4. Comparisons with other methods

R software for other methods are available for alignment including non-penalised DTW using the **dtw** package and parametric time warping using the **ptw** package. Another commonly used method is correlation optimised warping but R software is not available at this time; MATLAB implementations are available (Skov *et al.* 2006). We only compare methods implemented in R in this paper.

Comparison of different methods based on the quality of alignment is not a straightforward task as each method involves several decisions on the part of the user to obtain decent alignment results. In our case that may involve decisions on the bandwidth and on the penalty. One might also wish to explore whether or not a background correction is advantageous, or whether one should divide by the signal mean. Many of these issues affect the final result. While we noted deficiencies in quality of the alignments from **ptw** in Clifford *et al.* (2009), we believe comparable alignment using **ptw** can be achieved by an experienced user. As such, here we only compare methods implemented in R in this paper in terms of the time to computation.

4.1. Comparisons in terms of speed of computation

We proceed to compare **VPdtw** with **dtw** and **ptw** in terms of speed of computation. We use a Sakoe-Chiba window and an asymmetric step pattern with **dtw** with the same window size to make the comparison fair in terms of speed of computation. For signals of various lengths up to length 5000, we extract a random subsection of the data supplied with **VPdtw** and perform alignments using each of these methods. This is carried out five times for a particular length. Figure 13 is a plot of the computation times in seconds (on a log-scale) as a function of the length of the signals being aligned. Quantile smoothing (Koenker 2012) is employed to model the median time as a function of the length of the signals. A dashed black line showing the results of this is included for each method. As to be expected the computation times increase

Figure 13: Comparison of the computation times for the three methods. Time of computation (seconds) is plotted on a log scale against the signal length.

with signal length. In general **dtw** is the slowest method and **VPdtw** provides the fastest method and scales linearly with signal length. The performance of **ptw** lies in between and is a lot more variable than the alternatives. For signals of length 5000, the median computation times are 0.14s, 0.78s and 14.8s for **VPdtw**, **ptw** and **dtw** respectively.

While a runtime of 15s may not seem very large, in high-throughput applications involving hundreds of signals there is great importance placed on minimizing total run-time. The time savings gained by using **VPdtw** or **dtw** instead of **ptw** are considerable. As such, it is clear that **dtw** is a less viable option compared with **ptw** and **VPdtw** for alignment of large datasets.

5. Conclusions

The **VPdtw** package provides users with the means of aligning chromatograms in a fast and efficient manner. The software implements variable penalty dynamic time warping which is a fundamental variation to a standard and well understood alignment method. This approach avoids peak distortion by design. Comparisons with competing methods here and in Clifford *et al.* (2009) show that **VPdtw** works at faster speeds while also returning high quality results with minimal peak distortion. **VPdtw** provides the means for investigation of the appropriate penalty as well as a means for aligning multiple samples at once. The syntax is straightforward and summary information and plotting capabilities highlight the results in an intuitive format. **VPdtw** is now available on CRAN under a GPL-2 license.

References

- America AHP, Cordewener JHG, van Geffen MHA, Lommen A, Vissers JPC, Bino RJ, Hall RD (2006). "Alignment and Statistical Difference Analysis of Complex Peptide Data Sets Generated by Multidimensional LC-MS." *Proteomics*, **6**(2), 641–653.
- Berna AZ, Trowell S, Clifford D, Cynkar W, Cozzolino D (2009). "Geographical Origin of Sauvignon Blanc Wines Predicted by Mass Spectrometry and Metal Oxide Based Electronic Nose." *Analytica Chimica Acta*, **648**(2), 146–152.
- Bloemberg TG, Gerretzen J, Wouters HJP, Gloerich J, van Dael M, Wessels HJCT, van den Heuvel LP, Eilers PHC, Buydens LMC, Wehrens R (2010). "Improved Parametric Time Warping for Proteomics." *Chemometrics and Intelligent Laboratory Systems*, **104**(1), 65–74.
- Christin C, Smilde AK, Hoefsloot HCJ, Suits F, Bischoff R, Horvatovich PL (2008). "Optimized Time Alignment Algorithm for LC-MS Data: Correlation Optimized Warping Using Component Detection Algorithm-Selected Mass Chromatograms." *Analytical Chemistry*, **80**(18), 7012–7021.
- Clifford D, Stone G, Lovell D, Ramadan Z, Kochhar S (2007). "Issues in Aligning Multiple-MS Spectra." In S Barber, PD Baxter, KV Mardia (eds.), *Systems Biology and Statistical Bioinformatics*, pp. 105–107. Leeds, Leeds University Press.
- Clifford D, Stone G, Montoliu I, Rezzi S, Martin FP, Guy P, Bruce S, Kochhar S (2009). "Alignment Using Variable Penalty Dynamic Time Warping." *Analytical Chemistry*, **81**(3), 1000–1007.

- Daszykowski M, Heyden YV, Boucon C, Walczak B (2010). “Automated Alignment of One-Dimensional Chromatographic Fingerprints.” *Journal of Chromatography A*, **1217**(40), 6127–6133.
- Eilers PHC (2004). “Parametric Time Warping.” *Analytical Chemistry*, **76**(2), 404–411.
- Gerretzen J, Eilers P, Wouters H, Bloemberg T, Wehrens R (2010). *ptw: Parametric Time Warping*. R package version 1.0-1, URL <http://CRAN.R-project.org/package=ptw>.
- Giorgino T (2009). “Computing and Visualizing Dynamic Time Warping Alignments in R: The **dtw** Package.” *Journal of Statistical Software*, **31**(7), 1–24. URL <http://www.jstatsoft.org/v31/i07/>.
- Giskeodegard GF, Bloemberg TG, Postma G, Sitter B, Tessem MB, Gribbestad IS, Bathen TF, Buydens LMC (2010). “Alignment of high resolution magic angle spinning magnetic resonance spectra using warping methods.” *Analytica Chimica Acta*, **683**(1), 1–11.
- Juhász Z (2007). “Analysis of Melody Roots in Hungarian Folk Music Using Self-Organizing Maps with Adaptively Weighted Dynamic Time Warping.” *Applied Artificial Intelligence: An International Journal*, **21**(1), 35–55.
- Keogh E, Pazzani M (2001). “Derivative Dynamic Time Warping.” In *First SIAM International Conference on Data Mining*. Chicago.
- Koenker R (2012). *quantreg: Quantile Regression*. R package version 4.79, URL <http://CRAN.R-project.org/package=quantreg>.
- Nielsen NPV, Carstensen JM, Smedsgaard J (1998). “Aligning of Single and Multiple Wavelength Chromatographic Profiles for Chemometric Data Analysis Using Correlation Optimised Warping.” *Journal of Chromatography A*, **805**(1-2), 17–35. ISSN 0021-9673.
- R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org/>.
- Roberts K, Lawrence P, Eisen A, Hoirch M (1987). “Enhancement and Dynamic Time Warping of Somatosensory Evoked Potential Components Applied to Patients with Multiple Sclerosis.” *IEEE Transactions on Biomedical Engineering*, **34**(6), 397–405. ISSN 0018-9294.
- Robinson M, Souza DD, Keen W, Saunders E, McConville M, Speed T, Likic V (2007). “A Dynamic Programming Approach for the Alignment of Signal Peaks in Multiple Gas Chromatography-Mass Spectrometry Experiments.” *BMC Bioinformatics*, **8**(1), 419. ISSN 1471-2105.
- Sakoe H, Chiba S (1971). “A Dynamic Programming Approach to Continuous Speech Recognition.” In *Proceedings of the Seventh International Congress on Acoustics*, volume 3, pp. 65–69.
- Skov T, van den Berg F, Tomasi G, Bro R (2006). “Automated Alignment of Chromatographic Data.” *Journal of Chemometrics*, **20**(11-12), 484–497.

- Smith CA, Want EJ, O'Maille G, Abagyan R, Siuzdak G (2006). “**XCMS**: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification.” *Analytical Chemistry*, **78**(3), 779–787.
- Soille P (1999). *Morphological Image Analysis: Principles and Applications*. Springer-Verlag.
- Sun H, Lui JCS, Yau DKY (2006). “Distributed Mechanism in Detecting and Defending Against the Low-Rate TCP Attack.” *Computer Networks*, **50**(13), 2312–2330. ISSN 1389-1286.
- Tormene P, Giorgino T, Quaglini S, Stefanelli M (2008). “Matching Incomplete Time Series With Dynamic Time Warping: An Algorithm and an Application to Post-Stroke Rehabilitation.” *Artificial Intelligence in Medicine*, **45**(1), 11–34.
- Vintsyuk TK (1968). “Speech Discrimination by Dynamic Programming.” *Kibernetika*, **4**(2), 81–88.
- Voss B, Hanselmann M, Renard, Y B, Lindner MS, Köthe U, Kirchner M, Hamprecht FA (2011). “SIMA: Simultaneous Multiple Alignment of LC/MS Peak Lists.” *Bioinformatics*, **27**(7), 987–93.
- Wang CP, Isenhour TL (1987). “Time-Warping Algorithm Applied to Chromatographic Peak Matching Gas Chromatography/Fourier Transform Infrared/Mass Spectrometry.” *Analytical Chemistry*, **59**(4), 649–654.
- Watkins PJ, Clifford D, Rose G, Allen D, Warner R, Dunshea F, Pethick D (2010). “Sheep Category can be Classified Using Machine Learning Techniques Applied to Fatty Acid Profiles Derivatised as Trimethylsilyl Esters.” *Animal Production Science*, **50**, 782–791.
- Zhang ZM, Liang YZ, Lu HM, Tan BB, Xu XN, Ferro M (2012). “Multiscale Peak Alignment for Chromatographic Datasets.” *Journal of Chromatography A*, **1223**(0), 93–106.

Affiliation:

David Clifford
Division of Mathematics, Informatics and Statistics
Commonwealth Scientific and Industrial Research Organisation (CSIRO)
PO Box 2583
Brisbane, QLD 4001, Australia
E-mail: David.Clifford@csiro.au
URL: <http://www.csiro.au/org/CMIS.html>

Glenn Stone
School of Computing, Engineering and Mathematics
University of Western Sydney

Locked Bay 1797
Penrith, NSW 2751, Australia
E-mail: G.Stone@uws.edu.au
URL: <http://www.uws.edu.au/scem/>