Journal of Statistical Software

October 2013, Volume 55, Issue 3. *http://www.jstatsoft.org/*

Warping Functional Data in R and C via a Bayesian Multiresolution Approach

Leen Slaets
KU Leuven

Gerda Claeskens
KU Leuven

Bernard W. Silverman
University of Oxford

Abstract

Phase variation in functional data obscures the true amplitude variation when a typical cross-sectional analysis of these responses would be performed. Time warping or curve registration aims at eliminating the phase variation, typically by applying transformations, the warping functions τ_n, to the function arguments. We propose a warping method that jointly estimates a decomposition of the warping function in warping components, and amplitude components. For the estimation routine, adaptive MCMC calculations are performed and implemented in C rather than R to increase computational speed. The R-C interface makes the program user-friendly, in that no knowledge of C is required and all input and output will be handled through R. The R package **MRwarping** contains all needed files.

Keywords: functional data, time warping, curve registration, adaptive MCMC, C, R.

1. Introduction

Functional data analysis involves the analysis of a set of curves or images e.g., brain potentials (Kneip and Gasser 1992), (2D) facial shape data in biology (Barry and Bowman 2008), bidding patterns in online auctions (Peng and Müller 2008) and market penetration data (Sood, James, and Tellis 2009) in economics. Although there is a similarity with longitudinal data (see Hall, Müller, and Wang 2006), functional data are considered through a different conceptual approach. Ramsay and Silverman (2002) provide a clear overview oriented towards practice, facilitating the transfer of functional data methodology from the academic context to society and industry.

An important aspect of functional data is the recognition of phase variation. Most statistical methodology is designed to seek cross-sectional structure in the response values. That is, they study the variation in amplitude in the data. When complex processes are observed over

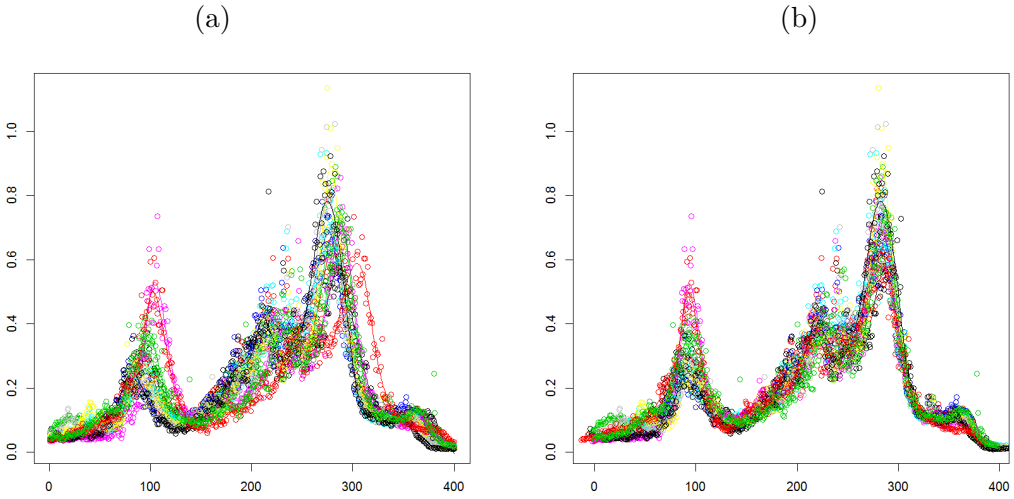

Figure 1: (a) Original LC-MS data which vary both in phase and in amplitude (together with penalized spline smoothed curves). (b) The penalized spline smoothed LC-MS data after warping using four warplets.

time or some other domain, however, another source of variation, so-called phase variation can arise. Figure 1 illustrates this for a curve sample of total ion counts (TIC) of a liquid chromatography - mass spectrometry (LC-MS) data set (Listgarten, Neal, Roweis, and Emili 2005). In the original sample, see Figure 1 (a), the time axes are misaligned in a non-trivial way, due to variable conditions (temperature, pressure,...) in the LC step that cannot be remedied during the experiment. This obscures the true amplitude variation when a typical cross-sectional analysis of these responses is performed. In other situations the phase variation could be of interest itself, e.g., the fact that a data peak is delayed might contain important information for the further analysis of the data.

Time warping or curve registration aims at eliminating the phase variation in a functional sample. It achieves this goal by applying transformations, the warping functions τ_n, to the function arguments. Many models have been considered in the literature aimed to capture phase variation as it is intuitively perceived by the data analyst. Landmark registration (Kneip and Gasser 1992) is one of the earliest methods and requires the identification of curve features or landmarks. The approaches by Silverman (1995), later extended in Ramsay and Li (1998) to continuous monotone registration, and Wang and Gasser (1997) are not based on landmarks but on the minimization of a distance measure between the curves. More recent are likelihood-based methods by Rønn (2001) and Gervini and Gasser (2005), and curve alignment by moments (James 2007), the latter combining advantages of landmark and continuous monotone registration. Warping or registration of the functional observations takes place before nearly any further analysis. Explicit examples include a study of leg growth velocities (Gervini and Gasser 2004), and of the geometries of the internal carotid artery (Sangalli, Secchi, Vantini, and Veneziani 2009; Vantini 2012).

In Claeskens, Silverman, and Slaets (2010) a model is proposed for time warping that also takes the amplitude variability into account. Similar to Gervini and Gasser (2005), a warping function is applied to transform the time domain and a random effects structure is added

to represent amplitude variation. The main novelty of the model in Claeskens *et al.* (2010) is that the warping function is constructed through a multiresolution structure with a clear interpretation in the warping framework. The spline basis functions in the amplitude structure of that model, however, are not estimated in the model, but need to be specified by the user.

In this paper the model in Claeskens *et al.* (2010) is extended to jointly estimate the warping and amplitude components. Instead of B-spline basis functions, a limited number of asymmetric rescaled kernel functions are used to indicate modes of amplitude variation. Apart from faster functional evaluation, these kernels have the advantage that their parameters have an easy graphical interpretation and make it possible for the user to provide good starting values. The amplitude and warping components are presented in Section 2, together with the precise formulation of the model.

In fitting the model, there are many parameters to be estimated and, in addition, the decomposition structure of the warping function does not have a unique parameterization. To deal with this, we have developed a Bayesian estimation method (Claeskens *et al.* 2010), see Appendix A, which gathers the most important warping actions in the first components of the multiresolution structure. A step-by-step estimation routine provides a gradually extended model with additional warping components that progressively eliminate remaining phase variation. Two stopping methods are available, see Section A.4. To reduce the time required by the Markov chain Monte Carlo (MCMC) computations, we perform adaptive MCMC calculations and program them in C (Kernighan and Ritchie 1988) rather than R (R Core Team 2013). The R-C interface makes the program user-friendly, in that no knowledge of C is required and all input and output will be handled through R. The R package **MRwarping** contains all needed functionality and is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=MRwarping`.

2. Multi-resolution warping

Multiresolution warping allows for flexible domain transformations where the parameters of the transformation have a meaningful interpretation in the context of warping. This is in contrast with the use of spline basis functions, for example, which are no warping functions, and hence require constraints on the parameters to ensure the monotonicity of the resulting warping functions.

2.1. The warping model with amplitude adjustments

The functional data sample consisting of N curves has function values $y_{n,j}$ $(n = 1, \ldots, N)$ corresponding to a fixed set of T ordered discrete time points t_j $(j = 1, \ldots, T)$. The main model that we use for estimation consists of an overall mean function $\mu(\cdot)$, that potentially needs to be warped for a better alignment. The model allows for local amplitude variation, plus some random error. The time domain is warped using warplets τ_n, one for each curve. Each τ_n is composed of basic warplet component functions; a precise definition is given in Equation 2. The model also includes a horizontal shift parameter $w_{\text{shift},n}$ for each curve, which serves as a global warping action prior to applying the local warplets. To prevent inappropriate extreme warping, the shifts are restricted to 1/4 of the range of the time points in both directions. Thus $-(t_T - t_1)/4 \leq w_{\text{shift},n} \leq (t_T - t_1)/4$. While the warplets take care of the phase variation, the functions $\psi_k(\cdot)$ are used to model possible local amplitude variation

in the curves. Each $\psi_k(\cdot)$ is parameterized by a center, a lower and an upper bound. For a precise definition of these functions, see Equation 5. The warping model with amplitude adjustment is therefore defined as

$$y_{n,j} = F_{n,j} + e_{n,j} = \mu(\tau_n(t_j + w_{\text{shift},n})) + \sum_{k=1}^{K} b_{n,k}\psi_k(\tau_n(t_j + w_{\text{shift},n})) + e_{n,j}, \qquad (1)$$

with $b_{n,k}$ and $e_{n,j}$ independent realizations of respectively $\mathcal{N}(0, \sigma_k^2)$ and $\mathcal{N}(0, \sigma^2)$ for $n = 1, \ldots, N$, $j = 1, \ldots, T$ and $k = 1, \ldots, K$, the number of amplitude kernels. The function τ_n performs the warping action and is decomposed into warplets, see (3) with an explicit inverse, see (4). In model (1), K is considered to be a fixed constant and needs to be chosen by the user based on inspection of the data. For example, for the data in Figure 1, the smoothed curves in Figure 6 panel (a) reveal two locations of substantial amplitude variation: the areas around $t = 100$ and $t = 280$. A sensible choice would be $K = 2$. Since the main goal is to warp the data and since the amplitude variation is considered to be a nuisance effect, the amplitude coefficients $b_{n,k}$ are modeled as random effects. For the fixed effect shift parameters the following constraint is used, $w_{\text{shift},N} = -\sum_{i=1}^{N-1} w_{\text{shift},i}$. The recent works by Bigot and Gadat (2010) and Vimond (2010) also provide a model (the shape invariant model) for the estimation of shifts (and scales) under similar constraints.

During the course of the paper, the method and all the function arguments will be explained and illustrated by means of the LC-MS example (Figure 1). This data set contains TIC counts on $N = 11$ curve observations each at the same $T = 400$ time points.

Multiresolution warping by fitting model (1) is made available for easy usage via an R-C interface in the R package **MRwarping**, according to the Bayesian estimation procedure as described in Appendix A. We here present the function call in R, with only the required input arguments,

```
R> library("MRwarping")
R> MRwarp(Xdata, Ydata, kernel.s)
```

The details about this function follow, see Section 3. The input quantity `kernel.s` provides starting values for the location and scale of the amplitude kernels ψ_k, $k = 1, \ldots, K$, by specifying their centers, lower and upper boundaries.

2.2. Warplets

Multiresolution warping is built around warplets, which are local warping components that concentrate the warping action to a certain domain. Warplets are composed, one after the other, to form the final warping function. Warplets have a clear interpretation in terms of both location and intensity of the warp.

Warping functions need to be smooth strictly monotonically increasing functions in order to ensure they define a bijection of the original function domain and respect the natural ordering of the time points. Figure 2 illustrates the warping process. It shows the original curve (a), the original equally spaced time points (c), the warping function in (d) and the warped curve and warped observation points in (b), resp. (e). The warplets or warping components are designed to only warp a local area. Warplets are denoted in full by $\tilde{\tau}((a, \lambda, w_l, w_u); t)$, or abbreviated by $\tilde{\tau}(t)$. They are strictly increasing functions that deviate from the identity

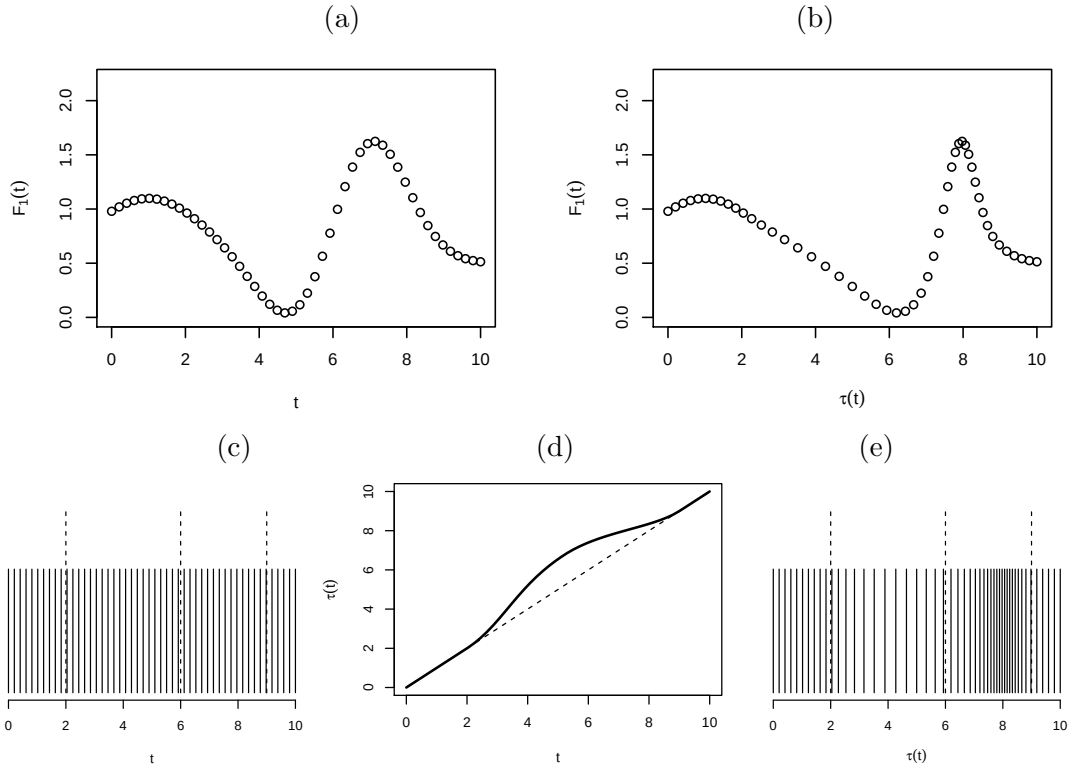

Figure 2: (a) Original data, (b) warped data, (c) original time points, (d) warping function, (e) warped time points.

function in a smooth manner on the interval $[a - r_1, a + r_2] = [w_l, w_u]$, the area where the warplet is active. In what follows we will mainly use the notation with the upper and lower bound (w_l,w_u) instead of the radii (r_1,r_2). The intensity parameter λ can take values in $(-1, 1)$. For a positive value of λ, the warplet will cause a dilation directly followed by a compression. When λ is negative, a compression is followed by a dilation. For $\lambda = 0$, no warping takes place. The intensity of the warping action increases with the absolute value of λ, as can be seen in Figure 4 (a) and (b). The component center a divides the warping intensity in a compression and dilation part, allowing for asymmetric actions. Similar as with the shift parameters, we restrict the domain of the warplets to the range of the time points, with the exception that w_l can be smaller than t_1, but not smaller than $t_1 - (t_T - t_1)/10$ and w_u can be larger than t_T but not larger than $t_T + (t_T - t_1)/10$. The latter exceptions accommodate phase variation near the borders of the time domain.

The following definition introduces the warplets more formally (see Definition 2.2 of Claeskens *et al.* 2010). Define the warplet

$$\begin{aligned} \tilde{\tau}\left(a, \lambda, w_l, w_u; t\right) &= \tilde{\tau}\left(a, \lambda, a - r_1, a + r_2; t\right) \\ &= \begin{cases} a + r_1 \cdot g\left(\lambda \frac{r}{r_1}; (t-a)/r_1\right), & t \in [a - r_1, a - \frac{3\sqrt{3}}{8}\lambda r] \\ a + r_2 \cdot g\left(\lambda \frac{r}{r_2}; (t-a)/r_2\right), & t \in [a - \frac{3\sqrt{3}}{8}\lambda r, a + r_2] \\ t, & \text{otherwise,} \end{cases} \end{aligned} \tag{2}$$

with $r_1, r_2 > 0$, $r = \min(r_1, r_2)$, $\lambda \in (-1, 1)$, $g(\lambda; y) = z + \lambda K(z) = y + 2\lambda K(z)$ where z is

the solution to $z - \lambda K(z) = y$, and with the quartic warplet kernel K:

$$K(z) = \begin{cases} \frac{3\sqrt{3}}{8}(1-z^2)^2, & z \in [-1,1] \\ 0, & \text{otherwise.} \end{cases}$$

The function $g(\lambda; y)$ rotates the rescaled quartic kernel function $K(z)$ alongside the first diagonal, as is illustrated in panels (a) and (b) of Figure 4. This construction ensures that the warplets are monotone and guarantees that the inverse warplet, see (4), is again a warplet. For the use of other kernel functions, see Claeskens *et al.* (2010). For each curve n ($n = 1, \ldots, N$), the warplets $\tilde{\tau}_{n,q}$ ($q = 1, \ldots, Q$) are composed in a warping function $\tau_n = \tilde{\tau}_{n,Q} \circ \ldots \circ \tilde{\tau}_{n,2} \circ \tilde{\tau}_{n,1}$, where $\tilde{\tau}_{n,1}$ is executed first, then $\tilde{\tau}_{n,2}$, etc. The warping functions τ_n in model (1) are curve-specific and allow for different locations and different intensities,

$$\tau_n(t_j) = \tilde{\tau}(a_Q, \lambda_{n,Q}, w_{l,Q}, w_{u,Q}) \circ \ldots \circ \tilde{\tau}(a_1, \lambda_{n,1}, w_{l,1}, w_{u,1})(t_j). \tag{3}$$

The composition of monotone warplets ensures the monotonicity of the overall warping function and moreover it has the attractive property that the inverse transformation has an easy, explicit formula,

$$\tau_n^{-1} = \tilde{\tau}_{n,1}^{-1} \circ \ldots \circ \tilde{\tau}_{n,Q}^{-1}, \text{ with } \tilde{\tau}_{n,q}^{-1}(a, \lambda, w_l, w_u; t) = \tilde{\tau}_{n,q}(a, -\lambda, w_l, w_u; t). \tag{4}$$

The inverse warplets are used frequently, in the estimation routine, but also to plot the warped curves. For a single warplet, choosing for all N curves the same warping center and the same lower and upper bound results in a more parsimonious model in terms of the number of parameters. This is motivated since many curve samples tend to display phase variation on only a few joint locations. Different warplets will in general have different centers and bounds. The intensities $\lambda_{n,q}$ are curve-specific with $\lambda_{N,q} = -\sum_{n=1}^{N-1} \lambda_{n,q}$ for $q = 1, \ldots, Q$.

The function `warp` evaluates a warping function τ in a vector of time points. E.g., to obtain a plot of $\tau(t) = \tilde{\tau}(2, 0.4, 2 - 1.5, 2 + 2) \circ \tilde{\tau}(5, 0.6, 5 - 2, 5 + 3)(t)$, as in Figure 3 (a), execute:

```
R> t <- seq(0, 10, length.out = 1000)
R> tau.t <- warp(c(5, 2), c(0.6, 0.4), c(2, 1.5), c(3, 2), t)
R> plot(t, tau.t, type = "l", ylab = expression(tau(t)))
```

The same plot can be obtained by composing the warplets:

```
R> tau.t1 <- warp(5, 0.5, 2, 3, t)
R> tau.t <- warp(2, 0.4, 1.5, 2, tau.t1)
R> plot(t, tau.t, type = "l", ylab = expression(tau(t)))
```

Figures 3 (b) and (c) illustrate the effect of applying this warping function on a normal density function. The corresponding code is given below. Figure 4 (a) contains some other examples of warplets and Figure 4 (b) shows how they act on a curve.

```
R> y <- dnorm(t, mean = 5, sd = 1)
R> plot(t, y, type = "l")
R> plot(tau.t, y, type = "l", xlab = expression(tau(t)))
```

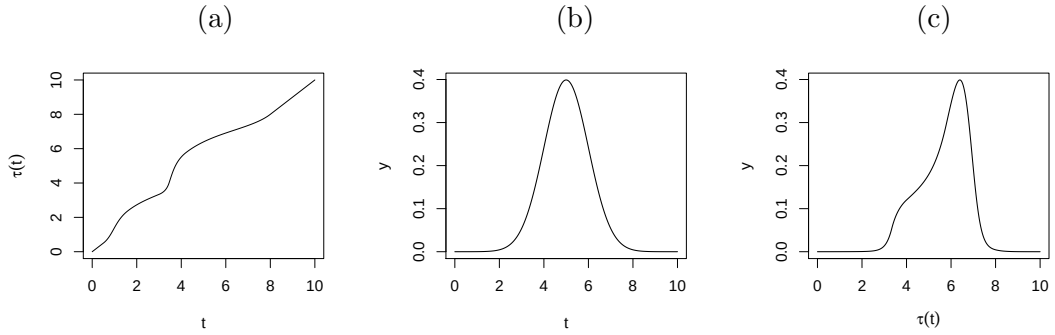

Figure 3: (a) warping function, (b) normal density curve (with mean 5 and variance 1) and (c) warped normal density curve.

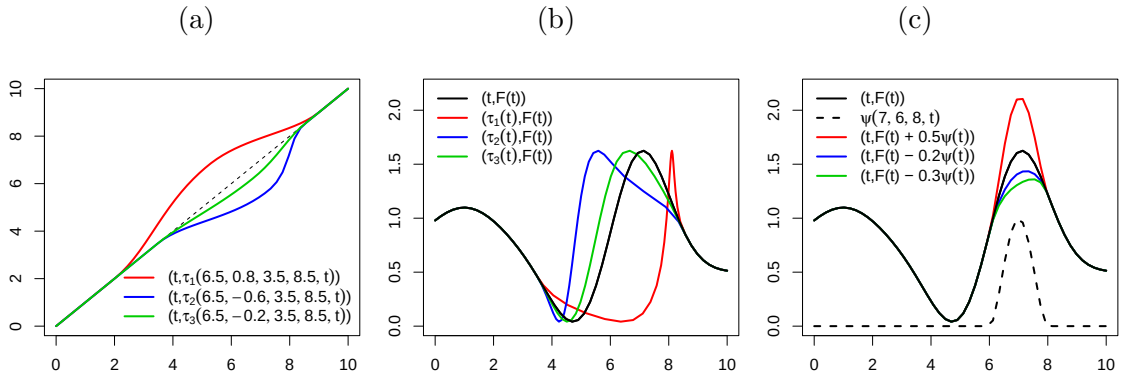

Figure 4: (a) warplets, (b) the effect of applying the warplets to the curve $(t, F(t))$ and (c) kernel (dashed) and curves with one amplitude component.

2.3. Amplitude components

In line with the choice of the warplets, we use rescaled asymmetric quartic kernels $\psi(\bar{a}, a_l, a_u; t)$ to model the amplitude variability in model (1),

$$\psi(\bar{a}, a_l, a_u; t) = \begin{cases} \left(1 - \left(\frac{(t-\bar{a})}{a_u - \bar{a}}\right)^2\right)^2, & \bar{a} \leq t \leq a_u \\ \left(1 - \left(\frac{(t-\bar{a})}{\bar{a} - a_l}\right)^2\right)^2, & a_l \leq t \leq \bar{a}. \end{cases} \tag{5}$$

Other choices are possible, for example, spline basis functions have been used in the papers by Gervini and Gasser (2005) and Claeskens *et al.* (2010).

Examples of amplitude kernels are given in Figure 4 (c). Even though the kernel parameters are estimated in the model, the number of the kernels and for each kernel starting values (rough guesses) of its center $\tilde{a}$, its lower (left) boundary a_l and its upper (right) boundary a_u need to be provided via the input quantity `kernel.s`, with $a_l \leq \bar{a} \leq a_u$. This vector is coded as follows `kernel.s` $= (a_{l,1}, \bar{a}_1, a_{u,1}, \ldots, a_{l,K}, \bar{a}_K, a_{u,K})$ and hence the length of `kernel.s` should be a multiple of 3.

For the example with the TIC responses, inspection of Figure 1 makes us choose two regions of local amplitude variation, related to the heights of the peaks around $t = 100$ and $t = 280$:

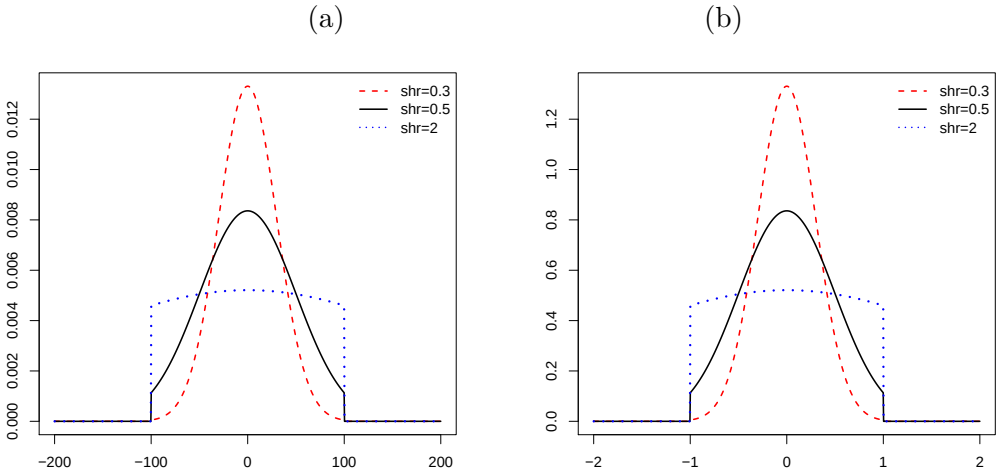

Figure 5: The effect of `shr` on the prior distributions on (a) the shifts with range $[-200, 200]$ and (b) warplet intensities with range $(-1, 1)$ in the LC-MS example.

`kernel.s <- c(70, 100, 130, 250, 280, 300)`. It is advisable to not choose more kernels than clearly suggested by visual inspection. Using too many kernels increases the computation time, our current experience is that any reasonable number will have limited influence on the main results.

2.4. Shrinkage of the warping action

The intensity of the warplets ($\lambda_{n,q}$) and magnitude of the shifts ($w_{shift,n}$) is governed by the input parameter `shr` of `MRwarp`. By default, the warplet intensities and shift parameters all have truncated normal prior distributions with zero means in the Bayesian estimation routine, which favors a low intensity warping action. The parameter `shr` corresponds to the standard deviation of the normal priors for the warplet intensities $\lambda_{n,q}$, and $0.5 \cdot$ `shr` $\cdot (t_T - t_1)$ is the standard deviation of the prior for the shifts. Figure 5 plots these priors for the intensities (a) and shifts (b) in the LC-MS example for the values `shr = 0.3` (the default value), `shr = 0.5` and `shr = 2`. The truncation of the normal prior for the $\lambda_{n,q}$ parameters guarantees that the intensities are contained within the $(-1, 1)$ interval. For the shifts these truncation points correspond to the extremal values for the allowed shifts ($-(t_T - t_1)/2$ and $(t_T - t_1)/2$), as mentioned in Section 2.1. The priors become less informative and the amount of shrinkage decreases with increased values of `shr`. In practice, extreme deformations can translate into a loss of smoothness and without informative priors on the intensities, it generally decreases the robustness against misspecification of the model. The setting of `shr=0.1`, offers a good balance between flexibility and smoothness in the LC-MS example.

2.5. Adding warplets stepwise or fixing Q beforehand

The function `MRwarp` offers a choice between two strategies regarding the number of warplets. With `selection = "FIXED"` and `components = 3` a fixed user-determined number of warplets (3 in this case) can be selected. The program displays intermediate results of the estimation procedure. The warped curves are plotted for 1, 2 and 3 components, up to the specified

number of components. After completion the R output vector contains the parameter values for the last estimated model (with 3 components in this example) and the one-but-last model (with 2 components in this example).

A second use of the program is by user interaction. The option `selection = "STEP"` makes the program ask the user whether or not he/she wants to continue to add an extra component after viewing the warped curves of each estimated model.

Additionally, the model selection information of Claeskens *et al.* (2010), see Section A.4, is displayed after each fit. It can facilitate the decision on whether or not to continue the fitting procedure, however visual inspection of the data is always advisable. The model selection criterion is based on $(1 - \alpha)\%$ highest posterior density intervals for the warping parameters of the newest component. See Section A.4 for more information. The default value of `alpha = 0.1` can be adjusted.

3. Software overview

Three R functions are contained in this package: `MRwarp`, `warp` and `comp`. The main function is `MRwarp`, which performs the actual warp by linking with C and calls the other two functions. The function `comp` computes a single quartic warplet, while the function `warp` is used to evaluate a composition of warplets. Table 1 presents an overview of the input arguments of the function `MRwarp`. Denote N the number of curves and T the number of time points. The

Argument	Description
`Xdata`	$N \times T$ matrix containing the x-coordinates or time points of the curve observations. Each row corresponds to a particular subject. No default.
`Ydata`	$N \times T$ matrix containing the y-coordinates or response values of the curve observations. Each row corresponds to a particular subject. No default.
`chain`	The (total) number of MCMC iterations (default = `400`).
`thin`	The thinning factor of the MCMC algorithm (default = `5`).
`burnin`	The number of MCMC iterations which are discarded (default = `200`).
`kernel.s`	Vector containing the starting values for the kernel parameters (see Section 2.3). No default.
`selection`	`"FIXED"` when we want to estimate a fixed number of warplets, `"STEP"` when evaluating the warping procedure after each component (default = `"FIXED"`).
`components`	The number of warping components in the final model (default = `1`). This value is ignored when `selection = "STEP"`.
`shr`	Determines the variance of the prior on the warplet intensities and shifts (see Section 2.4, default = `0.3`).
`outputfit`	`1` if the warped curves (based on the parameter values in the adaptive MCMC chain which give rise to the highest posterior density) should be plotted after each estimated fit, 0 otherwise (default = `1`).
`alpha`	The significance level to be used in the model selection procedure (see Section A.4) (default = `0.1`).

Table 1: Input of the function `MRwarp`.

Argument	Description
`$shift`	Adaptive MCMC chain of the estimated horizontal shifts $(w_{\text{shift},1}, \ldots, w_{\text{shift},N})$.
`$warping$lower`	Adaptive MCMC chains of the estimated warping lower bounds $(w_{l,1}, \ldots, w_{l,Q})$.
`$warping$A`	Adaptive MCMC chains of the estimated warping centers $(a_1, \ldots, a_Q)$.
`$warping$upper`	Adaptive MCMC chains of the estimated warping upper bounds $(w_{u,1}, \ldots, w_{u,Q})$.
`$warping$Intensities`	Adaptive MCMC chains of the estimated warping intensities $(\lambda_{1,1}, \ldots, \lambda_{N,1}, \lambda_{1,2}, \ldots, \lambda_{N,2}, \ldots \lambda_{1,Q}, \ldots, \lambda_{N,Q})$.
`$kernels`	Adaptive MCMC chains of the estimated kernel lower bounds, centers and upper bounds $(a_{l,1}, \bar{a}_1, a_{u,1}, \ldots, a_{l,K}, \bar{a}_K, a_{u,K})$.
`$error.variance`	The estimated value of the error variance σ^2.
`$max.post.dens`	The row in the parameter chain vectors/matrices corresponding to the highest posterior density.

Table 2: Output of the function `MRwarp`, both output lists `$last` and `$previous` consist of the components stated in this table.

R output is structured as a list with elements as listed in Table 2.

The output is given in the form of a list containing two components, `last` and `previous`, which can be accessed in the usual way by using the dollar sign. Both of the output components, for the last fitted model in `output$last` and for the one but last fitted model in `output$previous` (where `output` is the name given to the result of the call to the function `MRwarp`), are lists, each containing the components `shift`, `warping`, `kernels`, `error.variance` and `max.post.dens`.

The warping parameters in `warping` are grouped in a list containing the components `lower`, `A`, `upper`, `Intensities`, representing, respectively, the lower bounds $w_{l,q}$, the centers a_q, the upper bounds $w_{u,q}$ and the intensities $\lambda_{n,q}$, for $q = 1, \ldots, Q$ and $n = 1, \ldots, N$.

The default settings provide a good starting point for first time users. A simple model (one component) is fitted, the warped curves are plotted and the AMCMC chains are not too big to minimize the waiting period for completion of the routine. Inspecting these initial results (e.g., plotting adaptive MCMC parameter chains) immediately gives the user an idea of whether a longer adaptive MCMC chain is needed.

Manual preprocessing of the data is required in the following cases:

(i) An unequal number of observations for the different subjects. Data points can be omitted or interpolated for certain subjects or the data can be smoothed and predicted in a vector of time points of equal size.

(ii) Similar time and amplitude domains are required. Although the method can account for horizontal shifts, these are intended for relatively small global phase effects in the data, not to adjust different observation domains. E.g., when the data constitute a process observed for a month, cut into daily curves, these curves need to shifted to a one-day frame, prior to the analysis.

4. Example

This section contains the complete R code for the LC-MS example. Computation time for this entire example is ± 30–60 minutes, provided the user immediately supplies the needed input. Changing the `MRwarp` settings to `components = 3` and `selection = "FIXED"` yields the same final results, without requiring intermediate user input. Faster results are obtained with shorter adaptive MCMC chains (e.g., `chain = 100`; `burnin = 50`).

Reading the data

```
R> data("TICdata")
R> TIC <- as.matrix(TICdata)
```

Smoothing the LC-MS data

This is done to improve the linear interpolation performed to evaluate the pseudo-log-likelihood (see Section 2.1) at time points that are not originally observed.

```
R> index <- 1:200 * 2 - 1
R> TICy <- t(matrix(index, 200, 11))
R> x <- 1:400
R> for(i in 1:11) {
+    TIC.sm <- spm(TIC[i, ] ~ f(x))
+    TICy[i,] <- TIC.sm$fit$fitted[index]
+  }
R> TICx <- t(matrix(index,200, 11))
```

Multiresolution warping: options and estimation

See also Table 1.

```
R> output <- MRwarp(Xdata = TICx, Ydata = TICy, chain = 1000, thin = 5,
+    burnin = 500, kernel.s = c(70, 100, 130, 250, 280, 300), components = 1,
+    selection = "STEP", shr = 0.1, outputfit = 1, alpha = 0.1)
```

Output and answers provided to the program during the stepwise procedure.

```
"start C loop"
"C loop finished"
"Bayesian model selection criterion includes this component"
Do you want to continue and add a component? (y/n)y
"start C loop"
"C loop finished"
"Bayesian model selection criterion suggests not to include this component"
Do you want to continue and add a component? (y/n)y
"start C loop"
"C loop finished"
```

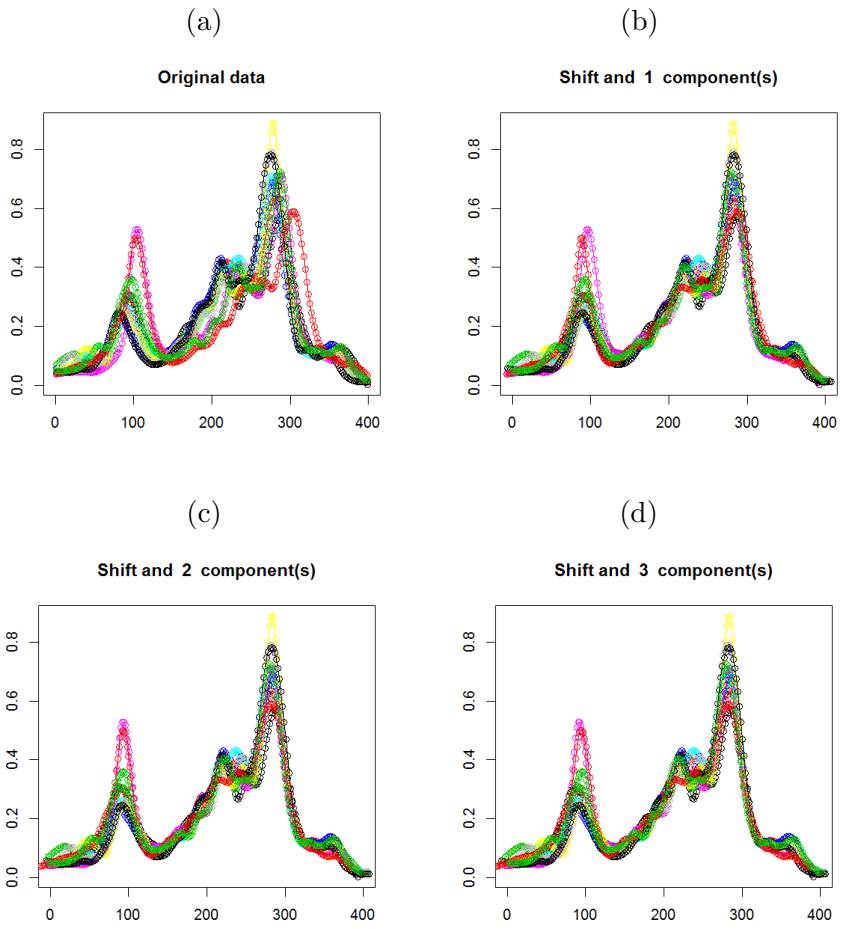

Figure 6: (a) Original data, and (b)–(d) stepwise warped curves (horizontal shifts and warping components). The plotted lines linearly interpolate the data.

```
"Bayesian model selection criterion suggests not to include this component"
Do you want to continue and add a component? (y/n)n
"program stopped after  3  components"
```

After estimation of the first model consisting of a shift and one warplet only, we receive plot (a) in Figure 6 and the following question: `Do you want to continue and add a component? (y/n)`. The first model already eliminates a substantial part of phase variation. To investigate whether this is further improved by including a second component, we answer the output question with `y` and enter. The program now extends the model by adding a second warplet (while updating the priors of the warping parameters by using the posterior information of the first model, see Section A.3). We continue until we are satisfied with the result, which in this example is at $Q = 2$. A third component was still added, but did not offer much improvement. The model selection information suggests to use just one components, but in this example we prefer to rely on a visual inspection of the data.

Extracting parameters of the final model ($Q = 2$) *from the output*

Three components have been fitted in the final model. Since we prefer the model with two components (the previous one), we use the values stored in the 'previous' part of the list. See also Table 2.

```
R> index <- output$previous$max.post.dens
R> index

 [1] 157
```

Output $(w_{\text{shift},1}, \ldots, w_{\text{shift},N})$ $(N = 11)$:

```
R> shift <- output$previous$shift[index,]
R> shift

 [1]  -9.607246   3.214980  -0.544125   5.686913   7.518277  -2.307322
 [7]   3.519495   4.971258   6.998859 -15.701871  -3.749218

R> A <- as.matrix(output$previous$warping$A)[index, ]
R> A
```

Output $(a_1, \ldots, a_Q)$ $(Q = 2)$:

```
 [1] 143.34798  28.67765
```

Output $(w_{u,1}, \ldots, w_{u,Q})$ $(Q = 2)$:

```
R> Wl <- as.matrix(output$previous$warping$lower)[index,]
R> Wl

 [1] 56.852001  8.224291
```

Output $(w_{u,1}, \ldots, w_{u,Q})$ $(Q = 2)$:

```
R> Wu <- as.matrix(output$previous$warping$upper)[index,]
R> Wu

 [1] 353.5670 222.5567
```

Output of $(\lambda_{1,1}, \ldots, \lambda_{N,1}, \lambda_{1,2}, \ldots, \lambda_{N,2}, \ldots \lambda_{1,Q}, \ldots, \lambda_{N,Q})$ $(Q = 2, N = 11)$

```
R> Intensities <- output$previous$warping$Intensities[index,]
R> Intensities

 [1]  0.1681539226  0.0489868217  0.0506884156  0.0646273158 -0.0625221529
 [6] -0.0777187035  0.0008774359 -0.0247585172  0.0366047925 -0.1124949934
[11] -0.0477684531  0.0977758034 -0.0851257317  0.0750820117 -0.0808696189
[16]  0.1231458679 -0.2354372868 -0.0926394026  0.1070670289 -0.0460199249
[21]  0.1847897063 -0.0477684531
```

Output $(a_{l,1}, \bar{a}_1, a_{u,1}, \ldots, a_{l,K}, \bar{a}_K, a_{u,K})$ $(K = 2)$:

```
R> kernels <- output$previous$kernels[index,]
R> kernels
```

```
 [1]  28.53527 104.91836 136.15320 190.15830 279.66676 318.79581
```

Plot of the warped data

While the function `MRwarp` provides plots of the warped curves corresponding to the (smoothed) input data $(\{(\widehat{\tau}_n(TICx[n,]), TICy[n,])\}, n = 1, \ldots, 11)$, it is also possible to use the estimated warping functions on other curves by means of the function `warp`. Below we illustrate how to use the output to apply the warping functions to the original, unsmoothed data TIC $(\{(\widehat{\tau}_n(x), TIC[n,])\}, n = 1, \ldots, 11)$. The resulting plot is shown in Figure 1 (b). For each curve the values that need to be supplied to the `warp` function (see Section 2.2) are extracted from the output.

```
R> x <- 1:400
R> T <- 400
R> N <- 11
R> Q <- length(A)
R> TIC.plot <- matrix(0, N, T)
R> WX <- t(matrix(x, T, N))
R> WX <- (WX) + shift
R> r1 <- A - Wl
R> r2 <- Wu - A
R> for(i in 1:N) {
+    ints <- Intensities[seq(from = i, to = (Q - 1) * N + i, by = N)]
+    WX[i,] <- warp(A, ints, r1, r2, WX[i, ])
+    wx <- WX[i,]
+    TIC.sm <- spm(TIC[i, ] ~ f(wx))
+    TIC.plot[i, ] <- TIC.sm$fit$fitted
+  }
```

In this example we have two warplets, the output vectors `r1`, `r2` and `A` give us their lower radius, upper radius and center. For these data the first warplet has $r_1 = 85.7061$, $r_2 = 202.88084$ and center $a = 155.51366$. The second warplet has $r_1 = 53.6720$, $r_2 = 103.2508$, and the center at value 62.80259.

```
R> plot(WX[1, ], TIC[1, ], xlab = "", ylab = "", ylim = range(TIC))
R> lines(WX[1, ], TIC.plot[1, ])
R> for(i in 2:N) {
+    points(WX[i, ], TIC[i, ], col = i)
+    lines(WX[i, ], TIC.plot[i, ], col = i)
+  }
```

5. Conclusion

The multiresolution warping method (Claeskens *et al.* 2010) has been extended to incorporate joint amplitude estimates in the form of rescaled kernel functions. The latter were chosen because of their parametrization which makes it easy for the user to interpret the parameters and to provide proper starting values.

Shrinking the warping intensities and the warping domain avoids too severe transformations and promotes data smoothness after warping.

The R-C interface for multiresolution warping combines the computational efficiency of C with the graphical features and user-friendliness of R. It provides the user with a several options to monitor the warping stage. Extensive output is available and can be consulted when desired.

Acknowledgments

The authors wish to express their thanks to all the reviewers of this paper, whose helpful comments have resulted in a cleaner software code and in a better structured paper. Financial support from the Research Fund KU Leuven (GOA) is also gratefully acknowledged.

References

Barry SJE, Bowman AW (2008). "Linear Mixed Models for Longitudinal Shape Data with Applications to Facial Modeling." *Biostatistics*, **9**, 555–565.

Bigot J, Gadat S (2010). "A Deconvolution Approach to Estimation of a Common Shape in a Shifted Curves Model." *The Annals of Statistics*, **38**, 2422–2464.

Claeskens G, Silverman BW, Slaets L (2010). "A Multiresolution Approach to Time Warping achieved by a Bayesian Prior-Posterior Transfer Fitting Strategy." *Journal of the Royal Statistical Society B*, **72**(5), 673–694.

Gervini D, Gasser T (2004). "Self-Modelling Warping Functions." *Journal of the Royal Statistical Society B*, **66**(4), 959–971.

Gervini D, Gasser T (2005). "Nonparametric Maximum Likelihood Estimation of the Structural Mean of a Sample of Curves." *Biometrika*, **92**(4), 801–820.

Hall P, Müller HG, Wang JL (2006). "Properties of Principal Component Methods for Functional and Longitudinal Data Analysis." *The Annals of Statistics*, **34**(3), 1493–1517.

James GM (2007). "Curve Alignment by Moments." *The Annals of Applied Statistics*, **1**(2), 480–501.

Kernighan BW, Ritchie DM (1988). *The C Programming Language*. 2nd edition. Prentice-Hall, Englewood Cliffs.

Kneip A, Gasser T (1992). "Statistical Tools to Analyze Data Representing a Sample of Curves." *The Annals of Statistics*, **20**(3), 1266–1305.

Listgarten J, Neal RM, Roweis ST, Emili A (2005). "Multiple Alignment of Continuous Time Series." In LK Saul, Y Weiss, L Bottou (eds.), *Advances in Neural Information Processing Systems 17*, pp. 817–824. MIT Press, Cambridge.

Peng J, Müller HG (2008). "Distance-Based Clustering of Sparsely Observed Stochastic Processes, with Applications to Online Auctions." *The Annals of Applied Statistics*, **2**, 1056–1077.

Ramsay JO, Li X (1998). "Curve Registration." *Journal of the Royal Statistical Society B*, **60**(2), 351–363.

Ramsay JO, Silverman BW (2002). *Applied Functional Data Analysis*. 2nd edition. Spinger-Verlag, New York.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Roberts GO, Rosenthal JS (2009). "Examples of Adaptive MCMC." *Journal of Computational and Graphical Statistics*, **18**(2), 349–367.

Rønn BB (2001). "Nonparametric Maximum Likelihood Estimation for Shifted Curves." *Journal of the Royal Statistical Society B*, **63**(2), 243–259.

Sangalli LM, Secchi P, Vantini S, Veneziani A (2009). "A Case Study in Exploratory Functional Data Analysis: Geometrical Features of the Internal Carotid Artery." *Journal of the American Statistal Association*, **104**(485), 37–48.

Silverman BW (1995). "Incorporating Parametric Effects into Functional Principal Components Analysis." *Journal of the Royal Statistical Society B*, **57**(4), 673–689.

Smith BJ (2007). "**boa**: An R Package for MCMC Output Convergence Assessment and Posterior Inference." *Journal of Statistical Software*, **21**(11), 1–37. URL http://www.jstatsoft.org/v21/i11/.

Sood A, James G, Tellis G (2009). "Functional Regression: A New Model for Predicting Market Penetration of New Products." *Marketing Science*, **28**, 36–51.

Vantini S (2012). "On the Definition of Phase and Amplitude Variability in Functional Data Analysis." *Test*, **21**, 676–696.

Vimond M (2010). "Efficient Estimation for a Subclass of Shape Invariant Models." *The Annals of Statistics*, **38**, 1885–1912.

Wand M (2013). ***SemiPar**: Semiparametic Regression*. R package version 1.0-4, URL http://CRAN.R-project.org/package=SemiPar.

Wang K, Gasser T (1997). "Alignment of Curves by Dynamic Time Warping." *The Annals of Statistics*, **25**(3), 1251–1276.

A. Bayesian estimation method

We now give some details about the pseudo-log-likelihood and the Bayesian methods that are used in the estimation routine.

A.1. A weighted pairwise log-likelihood function

A sum of weighted pairwise log-likelihoods is used to estimate all the parameters. The estimation of the common underlying mean function $\mu(\cdot)$ can be avoided by exploiting the invertibility of the warping functions. Indeed, for every possible combination of values $n_1 \neq n_2$ in $\{1, \ldots, N\}$ it holds that

$$\mu(t) = F_{n_2}(\tau_{n_2}^{-1}(t - w_{\text{shift},n_2})) - \sum_{k=1}^{K} b_{n_2,k}\psi_k(t + w_{\text{shift},n_1}),$$

and thus

$$\begin{aligned} F_{n_1}(t) &= F_{n_2}(\tau_{n_2}^{-1}(\tau_{n_1}(t + w_{\text{shift},n_1}) - w_{\text{shift},n_2})) - \sum_{k=1}^{K} b_{n_2,k}\psi_k(\tau_{n_1}(t + w_{\text{shift},n_1})) \\ &\quad + \sum_{k=1}^{K} b_{n_1,k}\psi_k(\tau_{n_1}(t + w_{\text{shift},n_1})) \\ y_{n_1,j} &= F_{n_2}(\tau_{n_2}^{-1}(\tau_{n_1}(t_j + w_{\text{shift},n_1}) - w_{\text{shift},n_2})) \\ &\quad + \sum_{k=1}^{K} \left(b_{n_1,k} - b_{n_2,k}\right) \psi_k\left(\tau_{n_1}(t_j + w_{\text{shift},n_1})\right) + e_{n_1,j}. \end{aligned} \tag{6}$$

The pseudo-log-likelihood is defined as the sum of the weighted pairwise log-likelihoods corresponding to the $N(N-1)$ pairwise models, see Equation 6, with $n_1 \neq n_2$. Denote $\boldsymbol{\alpha}_{\text{shift}} = \{w_{\text{shift},n}; n = 1, \ldots, N\}$ the shift parameters, $\boldsymbol{\alpha}_\tau = \{a_q, w_{l,q}, w_{u,q}, \lambda_{n,q}; q = 1, \ldots, Q, n = 1, \ldots, N\}$ the parameters of the warplet expansions of the warping functions, $\boldsymbol{\alpha}_\psi = \{\bar{a}_k, a_{l,k}, a_{u,k}, \lambda_i^\psi, \sigma_k^2; k = 1, \ldots, K\}$ the kernel parameters and the variances of the random amplitudes, and σ^2 the variance of the noise $e_{n,j}$. The pseudo-log-likelihood is given by

$$\begin{aligned} &\log L(\boldsymbol{\alpha}_{\text{shift}}, \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2) \\ &= \frac{-1}{(N-1)N} \sum_{n_1=1}^{N} \sum_{n_2=1, n_2 \neq n_1}^{N} \sum_{j=1}^{T} \left[\log\left(\sqrt{2\pi(\sum_{k=1}^{K} 2\psi_k^2(\tau_{n_1}(t_j + w_{\text{shift}}))\sigma_k^2 + \sigma^2)} \right) \right. \\ &\quad \left. + \frac{\left(y_{n_1,j} - f_{n_2}(\tau_{n_2}^{-1} \circ \tau_{n_1}(t_j + w_{\text{shift},n_1}) - w_{\text{shift},n_2})\right)}{2(\sum_{k=1}^{K} 2\psi_k^2(\tau_{n_1}(t_j + w_{\text{shift},n_1}))\sigma_k^2 + \sigma^2)} \right], \end{aligned} \tag{7}$$

where $f_n(t)$ are predicted values of $F_n(t)$ based on an interpolation of the data $\{t_j, y_{n,j}\}$. The number of kernels K is specified by the user.

The evaluation of this pairwise log-likelihood function requires function evaluations at time points at which the original observations might not have been observed. When the data display a lot of variation, rather than using linear interpolation to predict the intermediate

values, we smooth the data and create a new data set based on the smoothed curves. For the LC-MS data, `TIC` is the original data matrix and `TICx` and `TICy` are the new data, smoothed by using the package **SemiPar** (Wand 2013).

A.2. Adaptive MCMC

In the Bayesian philosophy, model parameters are random rather then fixed entities. They have a prior distribution $f\left(\boldsymbol{\alpha}_{\text{shift}}, \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right)$ which is used to obtain the posterior distribution f_{post}, see Equation 8. We use the pseudo-likelihood of Equation 7, that is, $f\left(\{y_n(t_j)\}_{j=1\ldots T}^{n=1\ldots N} | \boldsymbol{\alpha}_{\text{shift}}, \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right) = L(\boldsymbol{\alpha}_{\text{shift}}, \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2)$. This leads to the following expression for the posterior distribution,

$$\begin{aligned} & f_{post}\left(\boldsymbol{\alpha}_{\text{shift}}, \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right) = f\left(\boldsymbol{\alpha}_{\text{shift}}, \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2 \middle| \{y_n(t_j)\}_{j=1\ldots T}^{n=1\ldots N}\right) \\ = & \frac{f\left(\{y_n(t_{ij})\}_{j=1\ldots T}^{i=1,\ldots,N} | \boldsymbol{\alpha}_{\text{shift}}, \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right) f\left(\boldsymbol{\alpha}_{\text{shift}}, \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right)}{\int f\left(\{y_n(t_j)\}_{j=1,\ldots,T}^{n=n,\ldots,N} | \boldsymbol{\alpha}_{\text{shift}}, \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right) f\left(\boldsymbol{\alpha}_{\text{shift}}, \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right) d(\boldsymbol{\alpha}_{\text{shift}}, \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2)}. \end{aligned} \quad (8)$$

The program takes the following priors on each newly added component. Let $U(x_1, x_2)$ denote the uniform distribution on the interval (x_1, x_2).

$$\begin{aligned} & w_{\text{shift}} \sim \bar{\mathcal{N}}(0, (0.5 \cdot \texttt{shr} \cdot (t_T - t_1))^2, -0.5 \cdot (t_T - t_1), 0.5 \cdot (t_T - t_1)) \\ & a_q \sim U(t_1, t_T), w_{l,q} \sim U(t_1, t_T), w_{u,q} \sim U(t_1, t_T), \\ & \lambda_{n,q} \sim \bar{\mathcal{N}}(0, \texttt{shr}^2, -1, 1),\ q = 1, \ldots, Q; n = 1, \ldots, N, \\ & \bar{a}_k \sim U(t_1, t_T), a_{l,k} \sim U(t_1 - (t_T - t_1)/v), t_T), \\ & a_{u,k} \sim U(t_1, t_T + (t_1 - t_T)/v),\ k = 1, \ldots, K, \end{aligned}$$

with $v = 100$ to allow for an increased amplitude variation near the boundary of $[t_1, t_T]$, `shr` the shrinkage parameter (see Section 2.4), and an inverse gamma prior on σ and σ_k with shape and scale equal to 0.01, $k = 1, \ldots, K$.

Since the true posterior distribution is not tractable, numerical methods are used to obtain an informative sample. A Markov chain Monte Carlo (MCMC) procedure generates chains of dependent samples which converge to the equilibrium distribution, that is, the posterior distribution of the model parameters. The initial part of the chain, the burn-in period, is disregarded. The chain starts with a proper starting value which must be determined in accordance to the prior distribution

$$\{w_{\text{shift},n}^{(1)}, a_q^{(1)}, \lambda_{n,q}^{(1)}, w_{l,q}^{(1)}, w_{u,q}^{(1)}, \bar{a}_k^{(1)}, a_{l,k}^{(1)}, a_{u,k}^{(1)}, \sigma_k^{2(1)}, \sigma^{2(1)}\}_{q=1,\ldots,Q,\ k=1\ldots K}^{n=1\ldots N} = \{\boldsymbol{\alpha}_{\text{shift}}^{(1)}, \boldsymbol{\alpha}_\tau^{(1)}, \boldsymbol{\alpha}_\psi^{(1)}, \sigma^{2(1)}\}.$$

The parameter values in iteration i are denoted by $\{\boldsymbol{\alpha}_{\text{shift}}^{(i)}, \boldsymbol{\alpha}_\tau^{(i)}, \boldsymbol{\alpha}_\psi^{(i)}, \sigma^{2(i)}\}$.

We use a Metropolis-Hastings algorithm to generate a new proposal of parameter values at iteration $(i + 1)$ based on the previous iteration (i) by means of a proposal density P. The drawn sample will be approved with probability

$$P(\text{acceptance}) = \min\left\{\frac{f_{post}\left(\boldsymbol{\alpha}_{\text{shift}}^{(i+1)}, \boldsymbol{\alpha}_\tau^{(i+1)}, \boldsymbol{\alpha}_\psi^{(i+1)}, \sigma^{2(i+1)}\right)}{f_{post}\left(\boldsymbol{\alpha}_{\text{shift}}^{(i)}, \boldsymbol{\alpha}_\tau^{(i)}, \boldsymbol{\alpha}_\psi^{(i)}, \sigma^{2(i)}\right)} \cdot \frac{P^{(i),(i+1)}}{P^{(i+1),(i)}}, 1\right\}. \quad (9)$$

Here, $P^{(i),(i+1)}$ denotes the proposal density with mean $\{\boldsymbol{\alpha}_{\text{shift}}^{(i)}, \boldsymbol{\alpha}_{\tau}^{(i)}, \boldsymbol{\alpha}_{\psi}^{(i)}, \sigma^{2(i)}\}$, evaluated in the newly drawn values at step $(i+1)$. If the proposed value is rejected, the previous value is carried over unchanged and thus $\{\boldsymbol{\alpha}_{\text{shift}}^{(i+1)}, \boldsymbol{\alpha}_{\tau}^{(i+1)}, \boldsymbol{\alpha}_{\psi}^{(i+1)}, \sigma^{2(i+1)}\} = \{\boldsymbol{\alpha}_{\text{shift}}^{(i)}, \boldsymbol{\alpha}_{\tau}^{(i)}, \boldsymbol{\alpha}_{\psi}^{(i)}, \sigma^{2(i)}\}$. The algorithm works best (read: the chain converges fastest to a sample of the posterior) if the proposal density matches the shape of the target distribution, namely, the posterior distribution. Since the latter is unknown, we use adaptive MCMC (AMCMC), comparable to the adaptive Metropolis-within-Gibbs algorithm in Roberts and Rosenthal (2009), which updates the proposal density at regular times throughout the algorithm when more information on the posterior becomes available.

Our AMCMC scheme differs from that of Roberts and Rosenthal (2009), in two ways. First, the use of truncated normal proposal densities for each of the parameters. This way we make sure the generated parameter proposals give rise to valid warping functions and amplitude components, before evaluating the priors. Second, the updating of the variance of these densities is only done during the burn-in stage and, third, in our algorithm we do not always evaluate Equation 9 after a value has been drawn for a particular parameter. We explain this in more detail below.

When sampling in iteration (i), the other parameters are left unchanged. In iteration $(i+1)$ a new value is drawn from the distributions as given by the ordering in Equation 10 while the other values are carried over, and so on. Note that because $\lambda_{N,q} = -\sum_{n=1}^{N-1} \lambda_{n,q}$ for $q = 1, \ldots, Q$ and $w_{\text{shift},N} = -\sum_{i=1}^{N-1} w_{\text{shift},i}$, the intensities and shift of the Nth curve are not generated. Schematically,

$$
\begin{cases}
w_{\text{shift},n}^{(i+n)} \text{ drawn from } \bar{\mathcal{N}}\left(w_{\text{shift},n}^{(i)},\ (t_T - t_1)/100, -0.5 \cdot (t_T - t_1), 0.5 \cdot (t_T - t_1)\right), \\
n = 1, \ldots, N-1,
\end{cases}
$$

$$
\begin{cases}
a_Q^{(i+N)} & \text{drawn from } \bar{\mathcal{N}}\left(a_Q^{(i)},\ \sigma_{a_Q}^2, w_{l,Q}^{(i)}, w_{u,Q}^{(i)}\right), \\
w_{l,Q}^{(i+N+1)} & \text{drawn from } \bar{\mathcal{N}}\left(w_{l,Q}^{(i)},\ \sigma_{w_{l,Q}}^2, t_1 - 0.1 \cdot (t_T - t_1), a_Q^{(i+N)}\right), \\
w_{u,Q}^{(i+N+2)} & \text{drawn from } \bar{\mathcal{N}}\left(w_{u,Q}^{(i)},\ \sigma_{w_{u,Q}}^2, a_Q^{(i+N)}, t_T + 0.1 \cdot (t_T - t_1)\right), \\
\lambda_{n,Q}^{(i+N+2+n)} & \text{drawn from } \bar{\mathcal{N}}(\lambda_{n,Q}^{(i)},\ \sigma_{\lambda_{n,Q}}^2, \max\{-1, -1 - \sum_{j=1}^{n-1} \lambda_{j,q}^{(i+N+2+j)} \\
& \qquad - \sum_{j=n+1}^{N-1} \lambda_{j,q}^{(i)}\}, \min\{1, 1 - \sum_{j=1}^{n-1} \lambda_{j,q}^{(i+N+2+j)} - \sum_{j=n+1}^{N-1} \lambda_{j,q}^{(i)}\}), \\
& \qquad n = 1, \ldots, N-1,
\end{cases}
$$

$$
\sigma^{(i+2N+2)} \text{ drawn from } \bar{\mathcal{N}}\left(\sigma^{(i)},\ \sigma_{\sigma}^2, 0, (\max_{n,j} y_n(t_j) - \min_{n,j} y_n(t_j))\right). \tag{10}
$$

For $q = 1, \ldots, Q-1$:

$$
\begin{cases}
a_q^{(i+2N+3)} & \text{drawn from } \bar{\mathcal{N}}\left(a_q^{(i)},\ \sigma_{a_q}^2, w_{l,q}^{(i)}, w_{u,q}^{(i)}\right), \\
w_{l,q}^{(i+2N+3)} & \text{drawn from } \bar{\mathcal{N}}\left(w_{l,q}^{(i)},\ \sigma_{w_{l,q}}^2, t_1, a_q^{(i+2N+3)}\right), \\
w_{u,q}^{(i+2N+3)} & \text{drawn from } \bar{\mathcal{N}}\left(w_{u,q}^{(i)},\ \sigma_{w_{u,q}}^2, a_q^{(i+2N+3)}, t_T\right), \\
\lambda_{n,q}^{(i+2N+3)} & \text{drawn from } \bar{\mathcal{N}}(\lambda_{n,q}^{(i)},\ \sigma_{\lambda_{n,q}}^2, \max\{-1, -1 - \sum_{j=1}^{n-1} \lambda_{j,q}^{(i+2N+3)} \\
& \qquad - \sum_{j=n+1}^{N-1} \lambda_{j,q}^{(i)}\}, \min\{1, 1 - \sum_{j=1}^{n-1} \lambda_{j,q}^{(i+2N+3)} - \sum_{j=n+1}^{N-1} \lambda_{j,q}^{(i)}\}), \\
& \qquad n = 1, \ldots, N-1,
\end{cases}
$$

For $k = 1, \ldots, K$:

$$
\begin{aligned}
a_{l,k}^{(i+2N+3+k)} &\text{ drawn from } \begin{cases} \bar{\mathcal{N}}\left(a_{l,k}^{(i)}\ ,\ \sigma^2_{a_{l,k}}, t_1 - r, \bar{a}_k^{(i+2N+3+(k-1))}\right) & \text{if } k=1, \\ \bar{\mathcal{N}}\left(a_{l,k}^{(i)}\ ,\ \sigma^2_{a_{l,k}}, a_{u,(k-1)}^{(i+2N+3+(k-1))} + r, \bar{a}_k^{(i+2N+3+(k-1))}\right), & k>1, \end{cases} \\
\bar{a}_k^{(i+2N+3+k)} &\text{ drawn from } \bar{\mathcal{N}}\left(\bar{a}_k^{(i)}\ ,\ \sigma^2_{\bar{a}_s}, a_{l,k}^{(i+2N+3+k)}, a_{u,k}^{(i+2N+3+(k-1))}\right), \\
a_{u,k}^{(i+2N+3+k)} &\text{ drawn from } \begin{cases} \bar{\mathcal{N}}\left(a_{u,k}^{(i)}\ ,\ \sigma^2_{a_{u,k}}, \bar{a}_k^{(i+2N+3+k)} + r, a_{l,k+1}^{(i+2N+3+k)} - r\right) & \text{if } k<K, \\ \bar{\mathcal{N}}\left(a_{u,k}^{(i)}\ ,\ \sigma^2_{a_{u,k}}, \bar{a}_k^{(i+2N+3+k)} + r, t_T + r\right) & \text{if } k=K, \end{cases} \\
\sigma_k^{(i+2N+3+k+1)} &\text{ drawn from } \bar{\mathcal{N}}\left(\sigma_k^{(i)}\ ,\ \sigma^2_{\sigma_k}, 0, (\max_{n,j} y_n(t_j) - \min_{n,j} y_n(t_j))\right),\ k=1,\ldots,K
\end{aligned}
$$

where $\bar{\mathcal{N}}(x_1, x_2, x_3, x_4)$ denotes the truncated normal distribution on the interval (x_3, x_4) with mean x_1, variance x_2 and and with $r = (t_T - t_1)/100$, the minimum distance between the kernel parameters.

After one run through the iterations in Equations 10, only the final parameter values (corresponding to iteration $(i+2N+3+k+1)$) are considered for storage. This means that a new proposal has been considered for all parameters. Whether or not the values are stored depends on the `thinning` argument in the function call. The latter indicates after how many runs the values need to be stored. For `thinning = 1` the parameter values are stored after each run. The `chain` argument denotes the total number of stored values (burn-in included) and `burnin` the number of stored values that are thrown away. Thus the output chains (see Table 2) will contain `chain` $-$ `burnin` values for each parameter, corresponding to `chain` $\cdot$ `thinning` runs and `chain` $\cdot$ `thinning` $\cdot\,(i+2N+3+k+1)$ iterations.

To increase the computational speed, new values can be generated in clusters. We define as a cluster $\{\sigma_k^2\}_{k=1\ldots K}$, $\boldsymbol{\alpha}_\psi = \{\bar{a}_k, a_{l,k}, a_{u,k}, k = 1,\ldots,K\}$ and the warping parameters except those from the latest component, that is, $\{\bar{a}_q, w_{l,q}, w_{u,q}, \lambda_{n,q}; q = 1,\ldots,Q-1, n = 1,\ldots,N-1\}$. The reason why the last component is treated differently is explained in Section A.3.

The advantage of updating values one by one or in smaller clusters and not having one big cluster, is that we can monitor the acceptance probabilities of the parameters that are altered and evaluated separately. In order for the algorithm to converge sufficiently fast, an acceptance rate during the Metropolis-Hastings step in Equation 9 of roughly 44% is targeted by Roberts and Rosenthal (2009). The proposal variances in Equation 10 can thus be adjusted differently for each of the parameters, to better approximate the target density. Concretely this is done after 30 thinnings for the parameters. When we have more than $0.5 \cdot 30$ acceptances of that particular parameter or cluster, the corresponding proposal variances are increased by 25%, when it is lower than $0.4 \cdot 30$ they are increased by the same amount. We use a relatively large initial choice for the variances of the proposal densities to benefit the exploration phase.

At convergence, the iterated parameter values are still dependent draws from the posterior distribution. Their information content is therefore smaller than that of a sample of independent draws. For this reason thinning is applied, that is, we only store the estimates after several iterations.

The following settings are applied to the LC-MS example: `chain = 1000`, `thin = 5`, `burnin = 500`.

A.3. A prior-posterior transfer

Instead of building the model stepwise, immediately starting with a large number of warplets, say 6 components, would be problematic. The fact that the decomposition of τ is not unique can give rise to a multimodal posterior density of the warping parameters, which makes it difficult to detect convergence. The non-uniqueness is easily explained. For example, if the true unobserved warping function has two components, a model with with four warplets can simply take an arbitrary $\tau_{n,3}$ and have $\tau_{n,4} = (\tau_{n,3})^{-1}$. Or, in the case of warplets with a non-overlapping domain their order can be reversed. Such a multimodal posterior density is not only difficult to use to judge convergence, but more importantly, it disables any sort of interpretation of the parameters.

The solution that we offer is to build the model gradually. We start with a model with a single warplet and extend the model with one warplet at a time. The information gathered after estimating each such model is incorporated in the extended model in the next step in the form of an updated prior. As a result, we estimate a sequence of models in which each additional warplet is stimulated to eliminate the remaining phase variation only while the previous components take care of the warping actions that were already achieved in the simpler model.

The joint posterior distribution of the vector $\boldsymbol{\alpha}_\tau$ in a model with a single warplet is summarized by means of marginal histograms of the MCMC chains for each of the parameters. While more advanced methods could be used at this stage, we found the information contained in the histograms sufficient.

Since the adaptive MCMC has adjusted the proposal variances of the warping parameters in Equation 10 a separate Metropolis-Hastings evaluation step in Equation 9 is not necessary and rather a cluster of proposals is created as in Section A.2.

A.4. Selection of the number of warplets

Because each new warplet contributes less to the warping action than the already present warplets, a natural model selection procedure arises. When the newest warplet (indexed by Q) can not sufficiently improve the model, it will either operate on a very small domain, which results in an overlap of the highest posterior density (`hpd`) intervals of $w_{l,Q}$ and $w_{u,Q}$ and/or act with a low intensity, in which case all the highest posterior density intervals of $\lambda_{Q,n}$ contain 0. In case of one of these scenarios, the model selection step suggests to drop this additional component and opts for a reprise of the previously estimated model.

The $(1 - \texttt{alpha})$% highest posterior density intervals are calculated using the function `hpd` in the R package **boa** (Smith 2007).

Affiliation:

Leen Slaets, Gerda Claeskens
ORSTAT and Leuven Statistics Research Center
KU Leuven
Naamsestraat 69
3000 Leuven, Belgium
E-mail: Leen.Slaets@kuleuven.be, Gerda.Claeskens@kuleuven.be

Bernard W. Silverman
University of Oxford, United Kingdom
E-mail: Bernard.Silverman@stats.ox.ac.uk

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 55, Issue 3 *Submitted:* 2013-01-02
October 2013 *Accepted:* 2011-08-31