KATHOLIEKE UNIVERSITEIT
LEUVEN

Faculty of Business and Economics

Warping functional data in R and C via a Bayesian multiresolution approach

Leen Slaets, Gerda Claeskens and Bernard Silverman

DEPARTMENT OF DECISION SCIENCES AND INFORMATION MANAGEMENT (KBI)

KBI 1013

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

Leen Slaets
ORSTAT and Leuven Statistics Research Center
K.U.Leuven
3000 Leuven, Belgium
Leen.Slaets@econ.kuleuven.be

Gerda Claeskens
ORSTAT and Leuven Statistics Research Center
K.U.Leuven
3000 Leuven, Belgium
Gerda.Claeskens@econ.kuleuven.be

Bernard W. Silverman
University of Oxford
Oxford, United Kingdom
Bernard.Silverman@stats.ox.ac.uk

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 a transformation, the warping function τ, to the function argument. 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 MRwarp 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. Examples include growth curves in biology, volatility of assets in economy, ecg curves in cardiology and black and white images for face recognition. The different observations come naturally structured; they correspond to a certain time point, grid point, etc. Although there is a similarity with longitudinal data (see Hall et al., 2006), functional data require a separate treatment. Ramsay and Silverman (2002) provide a clear overview oriented towards practise, facilitating the transfer of functional data methodology from the academic circuit to society and industry.

An important aspect of a functional data analysis is the recognition of phase variation. Most statistical methodology is designed to seek structure in the response values. That is, they study the variation in amplitude in the data. When complex processes are observed over time, however, another source of variation, so-called phase variation can arise. Figure 1 illustrates this problem for a curve sample of total ion counts (TIC) of a liquid chromatography - mass spectrometry (LC-MS) data set (Listgarten et al., 2005). In the original sample (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 would be 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 a transformation, the warping function τ, to the function argument. This transformation needs to be a smooth monotonically increasing function in order to 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).

Many models have been considered in 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 certain 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.

In Claeskens et al. (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 resembling a multiresolution structure with a clear interpretation in the warping framework. The spline basis functions in the amplitude structure however, are not estimated in the model, but need to be specified by the user. In Gervini and Gasser (2005), the warping structure is estimated without amplitude components, while the latter are estimated using the estimated registered curves.

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

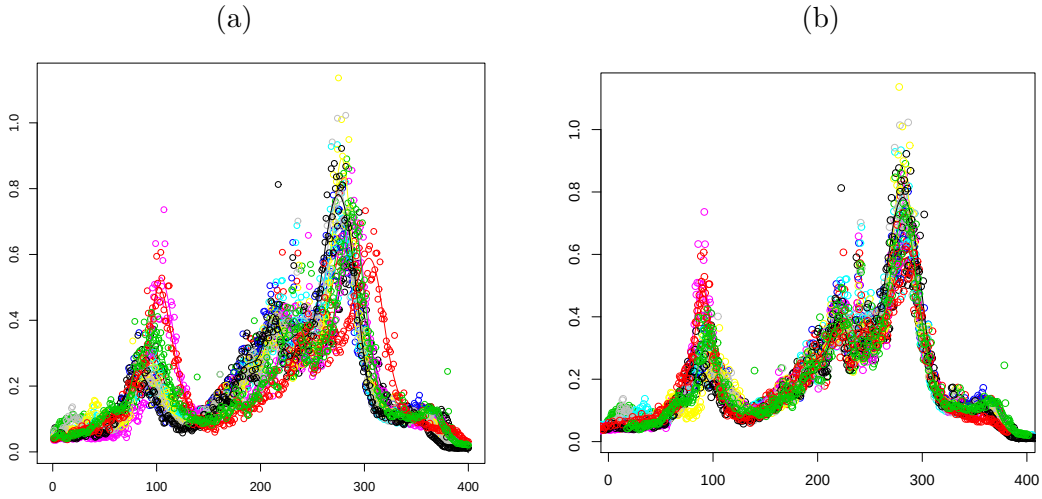

Figure 1: (a) Original LC-MS data which vary both in phase and in amplitude (together with penalized spline smoothed curves), (b) the penalized spline smoothed LC-MS data after warping using four warplets.

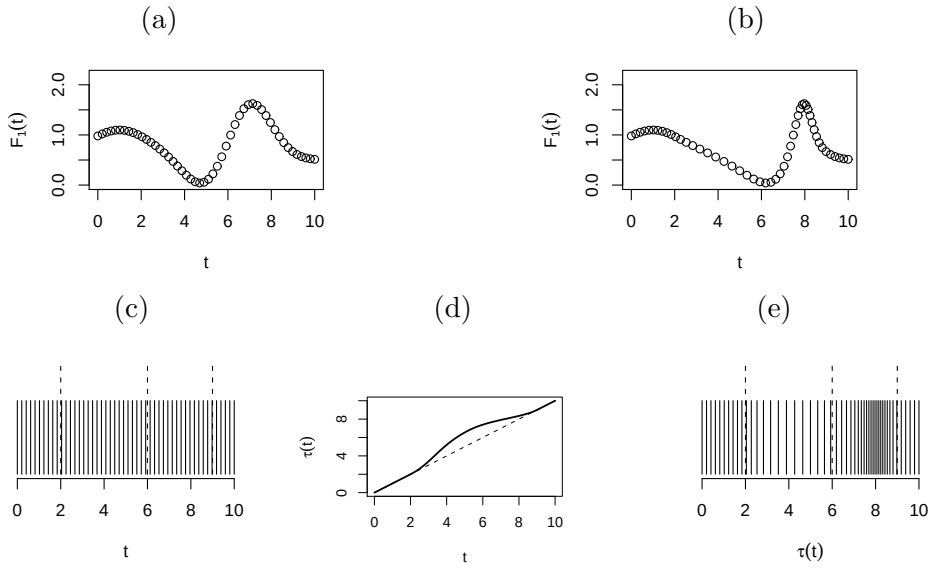

Figure 2: (a) Original data, (b) warped data, (c) original time points, (d) warping function, (e) warped time points.

a faster function 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 3, together with the precise formulation of the model.

The Achilles heel of the method is the estimation of the model. There are many parameters to be estimated and on top of that, 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 Section 4, which gathers the most important warping actions in the first components of the multiresolution structure. A step-by-step estimation routine provides the user output of a gradually extended model with additional warping components that eliminate less and less phase variation. Two stopping methods are available, see Section 4.4. Since we use Markov chain Monte Carlo (MCMC) computations the method is bound to be time consuming. For this reason we perform adaptive MCMC calculations and program them in **C** rather than **R** (R Development Core Team, 2010). 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**.

2 Software overview

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 sets contains TIC counts on 11 curve observations each at the same 400 time points.

Multiresolution warping is made available for easy usage via an **R**-**C** interface. It can be executed by using the **R**-library MRwarp and by running the following command lines in **R**, where the first call to MRwarp names the available input parameters, and the second call only includes the required input arguments:

```
library("MRwarp")
MRwarp(Xdata,Ydata,chain,thin,burnin,kernel.s,components,
       selection,thresh,threshd,prepr,outputfit,outputmcmc,alpha)
MRwarp(Xdata,Ydata,kernel.s,selection)
```

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 **R** output is structured as a list with arguments as listed in Table 2. Additional to the parameters in Table 2, which are directly provided in the output vector in **R**, the program creates and uses a number of `.txt` files in the directory of the .dll file. The latter can also be of interest of the user and are summarized in Table 3).

Four `.dll` files are available for different maximum values of N and S: interface100-200.dll (maximum 100 curves and 200 time points), interface50-200.dll (maximum 50 curves and 200 time points), interface100-400.dll (maximum 100 curves and 400 time points) and interface50-400.dll (maximum 50 curves and 400 time points). Computational speed is gained by choosing the file which corresponds the most to the data at hand. The `MRwarp` function does this automatically.

It is possible to manually set the maximum values equal to the actual sample size in the **C** code as explained in the Appendix and change the corresponding lines at the beginning and end of MRwarp to load and unload the .dll file.

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

- 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.

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 number of MCMC iterations (default=100).
thin	The thinning factor of the MCMC algorithm (default=10).
burnin	The (final) fraction of the MCMC chain which needs to be stored (default=0.5).
kernel.s	Vector containing the starting values for the kernel parameters. No default.
selection	Whether we want to estimate a fixed number of warplets ("FIXED"), or evaluate the warping procedure after each component ("STEP"). No default.
components	The number of warping components in the final model (default=1). For selection="FIXED" only.
thresh	Threshold on the warplet's intensity (value in $(0, 1]$, default=1)
threshd	Minimum distance between the center and lower and upper bounds of the warplets relative to the time-domain (value in $(0, 1]$, default=1).
prepr	Vector (p_1, p_2, p_3, p_4) with four 0/1 elements indicating whether a horizontal shift ($p_1 = 1$), a horizontal rescaling ($p_2 = 1$), a vertical shift ($p_3 = 1$) and/or a vertical rescaling ($p_4 = 1$) preprocessing actions should be included (default= $(0, 0, 0, 0)$).
outputfit	1 if the warped curves should be plotted, 0 otherwise (default=1).
outputmcmc	1 if parameter chains (excluding the burnin fraction) for a selected number of parameters should be plotted, 0 otherwise (default=0).
alpha	The significance level to be used in the model selection procedure (see Section 4.4) (default=0.05). For selection="STEP" only.

Table 1: Input of the function MRwarp.

Argument	Description
shift.w	The estimated horizontal shift in the preprocessing step for curves 1 to N.
scale.w	The estimated horizontal rescaling in the preprocessing step for curves 1 to N.
shift.a	The estimated vertical shift in the preprocessing step for curves 1 to N.
scale.a	The estimated vertical rescaling in the preprocessing step for curves 1 to N.
A.prev	Vector containing the estimated warping centers of the one-but-final model $(a_1, \ldots, a_{Q-1})$.
Ll.prev	Vector containing the estimated warping lower bounds of the one-but-final model $(w_{l,1}, \ldots, w_{l,Q-1})$.
Ul.prev	Vector containing the estimated warping upper bounds of the one-but-final model $(w_{u,1}, \ldots, w_{u,Q-1})$.
Lambda.prev	Vector containing the estimated warping intensities of the one-but-final model. Rows $(\lambda_{n,1}, \ldots, \lambda_{n,Q-1})$, n=1...N.
A	Vector containing the estimated warping centers of final model $(a_1, \ldots, a_Q)$.
Ll	Vector containing the estimated warping lower bounds of the final model $(w_{l,1}, \ldots, w_{l,Q})$.
Ul	Vector containing the estimated warping upper bounds of the final model $(w_{u,1}, \ldots, w_{u,Q})$.
Lambda	Vector containing the estimated warping intensities of the final model. Rows $(\lambda_{n,1}, \ldots, \lambda_{n,Q})$, n=1...N.
index	2-dimensional vector containing the iteration numbers (after burn-in) corresponding to the highest posterior pseudo-loglikelihood, for both the one-but final model and the final model.

In case selection="STEP" (see Table 1), the output list contains the estimated parameters for the last estimated model and the ones for the previous estimated model. In case selection="FIXED" or the user wishes to stop the algorithm after 1 component, the items in the list with the .prev suffix, are copies from the ones of the final estimated model (the list items without .prev).

Table 2: Output of the function MRwarp.

File	Description
`preproc.txt`	Estimated shift and scale parameters of the preprocessing step.
`amcmcvals.txt`	Iterated warping and error variance parameter chains (after burn-in) in the last estimated model with a uneven number of components.
`amcmcker.txt`	Iterated kernel parameter chains (after burn-in) in the last estimated model with a uneven number of components.
`amcmcamp.txt`	Iterated parameter chains of variances of the amplitude random effect (after burn-in) in the last estimated model with an uneven number of components.
`amcmcsigma.txt`	Proposal variances used after the burnin stage, in the last estimated model with an uneven number of components.
`UPDATE.txt`	The number of completed iterations (to keep track of progress) when it is a multiple of 100.
When these files have a suffix 2 (eg. `amcmcvals2.txt`) they correspond to the last estimated model with an even number of components.	

Table 3: Additional output files.

- Similar time and amplitude domains are required. Although the method can account for horizontal and vertical shifts and rescalings, these are intended for small global phase and amplitude 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.

3 Multi-resolution warping

3.1 Model components

Multiresolution warping distinguishes itself from existing warping methods, in that it allows for flexible domain transformations, in which 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. Multiresolution warping is built around warplets, these 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.

3.1.1 Warplets

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 τ. It are strictly increasing functions that deviate from the identity 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. This is achieved by rotating a rescaled quartic kernel function alongside the first diagonal, as is illustrated in panels (a) and (b) of Figure 3. For the use of other kernel functions, see Claeskens et al. (2010). For a positive value of λ, the warplet will cause a dilation directly followed by a compression with an intensity determined by the

value of λ. When λ is negative, a compression is followed by a dilation. The parameter λ can take values in $(-1, 1)$. The component center a divides the warping intensity in a compression and dilation part, allowing for asymmetric actions. It turns out (see Claeskens et al., 2010) that such asymmetric components require less warplets, and hence less parameters, than would be the case with symmetric components.

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

$$\begin{aligned} \tilde{\tau}\,(a, \lambda, w_l, w_u; t) &= \tilde{\tau}\,(a, \lambda, a - r_1, a + r_2; t) \\ &= \begin{cases} a + r_1 \cdot g\left(\lambda \frac{r}{r_2}; (t-a)/r_1\right), & t \in [a - r_1, a - \frac{3\sqrt{3}}{8}\lambda r] \\ a + r_2 \cdot g\left(\lambda \frac{r}{r_2}; (t-a)/r_2\right), & t \in [a - \frac{3\sqrt{3}}{8}\lambda r, a + r_2] \\ t, & \text{otherwise,} \end{cases} \end{aligned} \tag{1}$$

with $r_1, r_2 > 0$, $r = \min(r_1, r_2)$, $\lambda \in (-1, 1)$, $g(\lambda; y) = z + \lambda K^q(z)$ in which z is the solution to $z - \lambda K^q(z) = y$, and with the quartic warplet kernel K^q:

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

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 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.

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

```
t <- seq(0,10,length.out=1000)
tau.t <- warp(c(5,2),c(0.6,0.4),c(2,1.5),c(3,2),t)
plot(t,tau.t)
```

There are two arguments in the **MRwarp** function related to the warplets. Together they control the maximum severity of the deformations. The value `thresh` denotes the threshold on the λ parameter of all warplets in the model, while `threshd` is the minimum value of r_1 and r_2 relative to the observed time domain. The lower `thresh` and the larger `threshd`, the less extreme are the warplets. In practise, extreme deformations can translate into a loss of smoothness and a decreased robustness against misspecification of the model, depending on the underlying data. The following settings offer a good balance between flexibility and smoothness in the LC-MS example.

```
thresh <- 0.5
threshd <- 1/15
```

These values are incorporated in the estimation routine of section 4.1.

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

The amplitude components have a straightforward parametrization as can be seen in Figure 3 (c) and allow to specify the center $\tilde{a}$, lower (left) boundary a_l and upper (right) boundary a_u. This makes it possible for the user to provide good starting values for the estimation procedure. Moreover, similar to the warplets, the amplitude components can parsimoniously model a number of local areas of variation. For use in the model, see (4), we define

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

Even though the kernel parameters are estimated in the model, the number of kernels and starting values for the parameters need to be provided via the `kernel.s` argument. 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. A maximum of four kernels is allowed, but this can be altered in the `.c` file (see Appendix). For the example with the TIC responses, inspection of Figure 1 makes us choose three regions of local amplitude variation, related to the heights of the peaks around $t = 100$, $t = 285$ and the bumbs around t=30:

```
kernel.s <- c(0,20,40,70,100,130,270,285,300)
```

3.2 Preprocessing (shift and rescaling)

The main model is stated in (4). In case it would be necessary to shift and/or rescale the data before the actual application of the warping, a preprocessing step is performed. In this case we work with a simplified model. We make the assumption that the unobserved warping functions are surjective on $[l, u]$, by restricting the lower and upper warping bounds in the priors (8). The observations $y_n(t_j) = y_{n,j}$ arise from N random curves F_n that originate from one underlying mean function $\mu(t)$. For simplicity of notation we consider a fixed set of time points $t_{n,j} = t_j$, $j = 1, \ldots, T$, for $n = 1, \ldots, N$ on the interval $[l, u]$. Since model (4) focusses on local effects with respect to amplitude, we incorporate possible shifts and rescalings in a pre-processing step, that estimates the following simple model with four parameters $(w_{\text{shift},n}, w_{\text{scale},n}, a_{\text{shift},n}, a_{\text{scale},n})$ for each curve $n = 1, \ldots, N-1$. Denote $[l.y, u.y]$ the domain of the responses.

$$y_n(t_j) = F_n(t_j) + e_{n,j} = a_{\text{scale},n}\left[\mu\left(w_{\text{scale},n}\{t_j - c_t\} + c_t + w_{\text{shift},n}\right) - c_y\right] + c_y + a_{\text{shift},n} + e_{n,j}, \tag{3}$$

with $c_t = (l + u)/2$ and $c_y = (l.y + u.y)/2$ and $e_{n,j}$ independent realizations of $\mathcal{N}(0, \sigma^2)$.

Both the horizontal shifts $w_{\text{shift},n}$ and the vertical shifts $a_{\text{shift},n}$ $(n = 1, \ldots, N)$ sum to zero:

$$w_{\text{shift},N} = -\sum_{i=1}^{N-1} w_{\text{shift},i}, \quad a_{\text{shift},N} = -\sum_{i=1}^{N-1} a_{\text{shift},i},$$

and both the horizontal scales $w_{\text{scale},n}$ and vertical scales $a_{\text{scale},n}$ $(n = 1, \ldots, N)$ are constrained to be 1 on average:

$$w_{\text{scale},N} = N - \sum_{i=1}^{N-1} w_{\text{scale},i}, \quad a_{\text{scale},N} = N - \sum_{i=1}^{N-1} a_{\text{scale},i}.$$

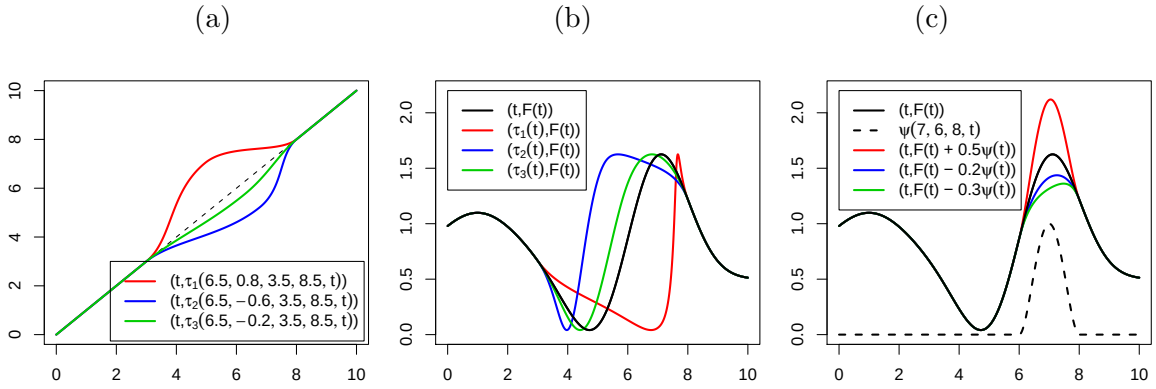

Figure 3: (a) warplets, (b) the effect of applying the warplets to the curve $(t, F(t))$ and (c) kernel (dashed) and curves with one amplitude component.

For estimation of the model in (3), the same pseudo-likelihood principle (6) as for model (4) in Section 3.3 can be applied, since we can construct pairwise submodels by eliminating $\mu(t)$ as in (5):

$$\begin{aligned} y_n(t_j) &= F_{n,j} + e_{n,j} = a_{\text{scale},n}\left[\mu\left(w_{\text{scale},n}\{t_j - c_t\} + c_t + w_{\text{shift},n}\right) - c_y\right] + c_y + a_{\text{shift},n} + e_{n,j}, \\ \mu(t) &= \frac{F_{n,j}\left[\frac{t - c_t - w_{\text{shift},n}}{w_{\text{scale},n}} + c_t\right] - c_y - a_{\text{shift},n}}{a_{\text{scale},n}} + c_y. \end{aligned}$$

Shifts are limited to 1/10th of the time domain (horizontally) and to 1/10th of the response domain (vertically) while scalings are limited to the interval $[1 - 1/10, 1 + 1/10]$. For editing these settings we refer to the Appendix. This preprocessing step is optional and the user can select which effects to include by adding the argument `prepr`$= (w_{\text{shift}}, w_{\text{scale}}, a_{\text{shifts}}, a_{\text{scale}})$, where a one implies that a horizontal (w) or a vertical (a) shift or rescaling takes place, a zero entry indicates no such transformation.

Returning to the LC-MS example, since the TIC response values more or less coincide in the beginning and end of the domain, we only include a horizontal shift and scale:
`prepr <- c(1,1,0,0)`.

3.3 Model formulation

Using the definitions of the warping functions in (1) and the amplitude components in (2), the warping model with amplitude adjustment is phrased as

$$y_{n,j} = F_{n,j} + e_{n,j} = \mu(\tau_n(t_j)) + \sum_{k=1}^{K} b_{n,k}\psi_k(\tau_n(t_j)) + e_{n,j}, \tag{4}$$

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 $n = 1, \ldots, N$, $j = 1, \ldots, T, k = 1, \ldots, K$. Since the main goal is to warp the data and since the amplitude variation is considered a nuisance effect, the amplitude coefficients $b_{n,k}$ are modeled as random effects. The curve-specific warping functions τ_n are modeled as follows,

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

The maximum amount of components is set to 6 in the `.dll` files, but this can be altered in the `.c` file (see Appendix). The curve-specific parameters are the intensities $\lambda_{n,q}$ (and the shift and rescaling parameters in case a preprocessing step is performed). As an averaging constraint, the intensity parameters satisfy that $\lambda_{N,q} = -\sum_{n=1}^{N-1} \lambda_{n,q} = 0$ for $q = 1, \ldots, Q$. Choosing the same warping center, and the same lower and upper bound for all curves will result in a more parsimonious model in terms of the number of parameters without limiting the flexibility. Indeed, curve samples tend to display phase variation on only a few locations, which can be taken to be common for all curves. A fine-tuning of the warping procedure is contained in adding further components.

3.4 Model estimation

A sum of weighted pairwise log-likelihoods is used to estimate the parameters in the N warping functions τ_n, the K kernels ψ_k and the variances of the random amplitudes and error term. The number of kernels K is specified by the user. Claeskens et al. (2010) introduced a Bayesian estimation strategy to estimate and select the number of warping components, see also Section 4. The estimation of $\mu(t)$ can be avoided by exploiting the invertibility of the warping functions. This holds since for every possible combination of values $n_1 \neq n_2$ in $\{1, \ldots, N\}$ we have that

$$\begin{aligned}
\mu(t) &= F_{n_2}(\tau_{n_2}^{-1}(t)) - \sum_{k=1}^{K} b_{n_2,k}\psi_k(t), \text{ and thus:} \\
F_{n_1}(t) &= F_{n_2}(\tau_{n_2}^{-1}(\tau_{n_1}(t))) - \sum_{k=1}^{K} b_{n_2,k}\psi_k(\tau_{n_1}(t)) + \sum_{k=1}^{K} b_{n_1,k}\psi_k(\tau_{n_1}(t)) \\
y_{n_1,j} &= F_{n_2}(\tau_{n_2}^{-1}(\tau_{n_1}(t_j))) + \sum_{k=1}^{K} \left(b_{n_1,k} - b_{n_2,k}\right) \psi_k\left(\tau_{n_1}(t_j)\right) + e_{n_1,j}. \qquad (5)
\end{aligned}$$

A pseudo-log-likelihood is constructed by summing the weighted pairwise log-likelihoods corresponding to the $N(N-1)$ pairwise models (5) with $n_1 \neq n_2$.
Denote $\boldsymbol{\alpha}_\tau = \{a_q, w_{l,q}, w_{u,q}, \lambda_{n,q}; q = 1, \ldots, Q, n = 1, \ldots, N-1\}$ 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

$$\begin{aligned}
\log L(\boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2) &= \frac{-1}{(N-1)N} \sum_{n_1=1}^{N} \sum_{n_2=1, n_2 \neq n_1}^{N} \sum_{j=1}^{T} \left[\log\left(\sqrt{2\pi(\sum_{k=1}^{K} 2\psi_i^2(\tau_{n_1}(t_{n_1,j}))\sigma_k^2 + \sigma^2)} \right) \right. \\
&\quad \left. + \frac{\left(y_{n_1,j} - f_{n_2}(\tau_{n_2}^{-1} \circ \tau_{n_1}(t_{n_1,j}))\right)}{2(\sum_{k=1}^{K} 2\psi_k^2(\tau_{n_1}(t_{n_1,j}))\sigma_k^2 + \sigma^2)} \right], \qquad (6)
\end{aligned}$$

where $f_n(t)$ are predicted values of $F_n(t)$ based on an interpolation of the data $\{t_{n,j}, y_{n,j}\}$. Extrapolation in the preprocessing step is done by carrying over the response value of the nearest observation. When the data display a lot of variation, linear interpolation is not a desirable method to predict intermediate values. As a solution the data can be smoothed beforehand and a new data set can be created based on the smoothed curves. This is also what we did for

the LC-MS data, in which `TIC` is the original data matrix and `TICx` and `TICy` the new data, smoothed by using the package SemiPar (Wand, 2010).

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

4 Algorithm

We adopted the Bayesian framework to overcome problems with numerical optimization which would occur when directly maximizing the pseudo log-likelihood (6). The possibility to incorporate prior information is exploited in a strategy in which we gradually extend the model by adding warping components as explained in section 4.2. It give also give rise to a natural model selection criterion (section 4.4). But first, section 4.1 summarizes the estimation of model (4) by means of adaptive MCMC.

4.1 Adaptive MCMC

In the Bayesian philosophy, model parameters are random rather then fixed entities. They have a prior distribution $f\left(\boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right)$ which is updated in the posterior distribution f_{post} (7) by the incorporation of the data likelihood $f\left(\{y_n(t_j)\}_{j=1\ldots T}^{n=1\ldots N} | \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right) = L(\boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2)$. We use the pseudo-likelihood of (6). This leads to the following expression for the posterior distribution,

$$\begin{aligned} f_{post}\left(\boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right) &= f\left(\boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2 \middle| \{y_n(t_j)\}_{j=1\ldots T}^{n=1\ldots N}\right) \\ &= \frac{f\left(\{y_n(t_{ij})\}_{j=1\ldots T}^{i=1\ldots N} | \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right) f\left(\boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right)}{\int f\left(\{y_n(t_j)\}_{j=1\ldots T}^{n=n\ldots N} | \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right) f\left(\boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right) d(\boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2)}. \end{aligned} \quad (7)$$

When we have no information on the parameters beforehand (which is usually the case) the priors are desired to be uninformative, such that the data (likelihood) completely shapes the posterior. We use the following non-informative priors for the warping and kernel parameters in $f\left(\boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_\psi, \sigma^2\right)$. Let $U(x_1, x_2)$ denote the uniform distribution on the interval (x_1, x_2).

$$\begin{aligned} &a_q \sim U(l,u), w_{l,q} \sim U(l,u), w_{u,q} \sim U(l,u), \\ &\lambda_{n,q} \sim U(-1,1),\ q = 1,\ldots,Q; n = 1,\ldots,N-1, \\ &\bar{a}_k \sim U(l,u), a_{l,k} \sim U(l-(u-l)/v), u), a_{u,k} \sim U(l, u+(l-u)/v),\ k = 1,\ldots,K, \\ &\text{with } v = (u-l)/100 \text{ to enable increased amplitude variation near the borders of } [l,u], \end{aligned} \quad (8)$$

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. An 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: $\{a_q^{(1)}, \lambda_{n,q}^{(1)}, w_{l,q}^{(1)}, w_{u,q}^{(1)}, \bar{a}_k^{(1)}, a_{l,k}^{(1)}, a_{u,k}^{(1)}, \sigma_k^{2(1)}, \sigma^{2(1)}\}_{q=1,\ldots,Q,\, k=1\ldots K}^{n=1\ldots N-1} = \{\boldsymbol{\alpha}_\tau^{(1)}, \boldsymbol{\alpha}_\psi^{(1)}, \sigma^{2(1)}\}$. The parameter values in iteration i are denoted by $\{\boldsymbol{\alpha}_\tau^{(i)}, \boldsymbol{\alpha}_\psi^{(i)}, \sigma^{2(i)}\}$.

The Metropolis-Hastings algorithm is a type of MCMC method that generates a new proposal $(i+1)$ based on the previous one (i) by means of a certain proposal density P. The drawn sample will be approved with probability

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

Here, $P^{(i),(i+1)}$ denotes the proposal density with mean $\{\alpha_\tau^{(i)}, \alpha_\psi^{(i)}, \sigma^{2(i)}\}$, 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 $\{\boldsymbol{\alpha}_\tau^{(i+1)}, \boldsymbol{\alpha}_\psi^{(i+1)}, \sigma^{2(i+1)}\} = \{\boldsymbol{\alpha}_\tau^{(i)}, \boldsymbol{\alpha}_\psi^{(i)}, \sigma^{2(i)}\}$. 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 throughout the algorithm when more information on the posterior comes available through the chain.

Different in our AMCMC scheme, as compared to that of Roberts and Rosenthal (2009), is, first, the use of truncated normal proposal densities for each of the parameters. This guarantees that the generated parameter proposals indeed give rise to valid warping functions and amplitude components. 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 (9) after a value has been drawn for a particular parameter. We explain this in more detail below.

P is taken to be a product of densities P_p as in (10) for each of the parameters p in the model. When sampling from P_{p_1} in iteration (i), the other parameters are left unchanged (thus P_p is set to degenerate distributions located in the current value in the chain for $p \neq p_1$ in step i). In iteration $(i+1)$ a new value is drawn from P_{p_2} as given by the ordering in (10) while the

other values are carried over, and so on.

$$\begin{cases} a_Q^{(i+1)} & \text{drawn from } \bar{\mathcal{N}}\left(a_Q^{(i)},\ \sigma^2_{a_Q}, w_{l,Q}^{(i)}, w_{u,Q}^{(i)}\right), \\ w_{l,Q}^{(i+2)} & \text{drawn from } \bar{\mathcal{N}}\left(w_{l,Q}^{(p+1)},\ \sigma^2_{w_{l,Q}}, l, a_Q^{(i+1)}\right), \\ w_{u,Q}^{(i+3)} & \text{drawn from } \bar{\mathcal{N}}\left(w_{u,Q}^{(i+2)},\ \sigma^2_{w_{u,Q}}, a_Q^{(i+2)}, u\right), \\ \lambda_{n,Q}^{(i+3+n)} & \text{drawn from } \bar{\mathcal{N}}\left(\lambda_{n,Q}^{(i+3+n-1)},\ \sigma^2_{\lambda_{n,Q}}, -thresh, thresh\right), n = 1, \ldots, N-1, \end{cases}$$

$$\sigma^{(i+3+N)} \text{ drawn from } \bar{\mathcal{N}}\left(\sigma^{(i+3+(N-1))},\ \sigma^2_\sigma, 0, (l_y - u_y)\right). \tag{10}$$

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

$$\begin{cases} a_{l,k}^{(i+4+N)} & \text{drawn from } \begin{cases} \bar{\mathcal{N}}\left(a_{l,k}^{(i+3+N)},\ \sigma^2_{a_{l,k}}, l - r, \bar{a}_k^{(i+3+N)}\right), & k = 1 \\ \bar{\mathcal{N}}\left(a_{l,k}^{(i+3+N)},\ \sigma^2_{a_{l,k}}, a_{u,(k-1)}^{(i+4+N)} + r, \bar{a}_k^{(i+3+N)}\right), & k > 1 \end{cases} \\ \bar{a}_k^{(i+4+N)} & \text{drawn from } \bar{\mathcal{N}}\left(\bar{a}_k^{(i+3+N)},\ \sigma^2_{\bar{a}_s}, a_{l,k}^{(i+4+N)}, a_{u,k}^{(i+3+N)}\right), \\ a_{u,k}^{(i+4+N)} & \text{drawn from } \begin{cases} \bar{\mathcal{N}}\left(a_{u,k}^{(i+3+N)},\ \sigma^2_{a_{u,k}}, \bar{a}_k^{(i+4+N)} + r, a_{l,k+1}^{(i+3+N)} - r\right), & k < K \\ \bar{\mathcal{N}}\left(a_{u,k}^{(i+3+N)},\ \sigma^2_{a_{u,k}}, \bar{a}_k^{(i+4+N)} + r, u + r\right), & k = K \end{cases} \end{cases}$$

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

$$\sigma_k^{(i+5+N)} \text{ drawn from } \bar{\mathcal{N}}\left(\sigma_k^{(i+5+N)},\ \sigma^2_{\sigma_k}, 0, (l_y - u_y)\right),$$

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

$$\begin{cases} a_q^{(i+6+S)} & \text{drawn from } \bar{\mathcal{N}}\left(a_q^{(i+5+N)},\ \sigma^2_{a_q}, w_{l,q}^{(p)}, w_{u,q}^{(i)}\right), \\ w_{l,q}^{(i+6+N)} & \text{drawn from } \bar{\mathcal{N}}\left(w_{l,q}^{(p+5+N)},\ \sigma^2_{w_{l,q}}, l, a_q^{(i+6+N)}\right), \\ w_{u,q}^{(i+6+N)} & \text{drawn from } \bar{\mathcal{N}}\left(w_{u,q}^{(p+5+N)},\ \sigma^2_{w_{u,q}}, a_q^{(i+6+N)}, u\right), \\ \lambda_{n,q}^{(i+6+N)} & \text{drawn from } \bar{\mathcal{N}}\left(\lambda_{n,q}^{(i+5+N)},\ \sigma^2_{\lambda_{n,q}}, -thresh, thresh\right), n = 1, \ldots, N-1, \end{cases}$$

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 $r = (u - l)/100$, the minimum distance between kernels parameters.

We not always perform the decision rule after each single parameter proposal, as already indicated by the iteration numbers in (10). To increase computation speed, new values can be generated in clusters. This is the 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 from the all but latest component $\{\bar{a}_q, w_{l,q}, w_{u,q}, \lambda_{n,q}; q = 1, \ldots, Q-1, n = 1, \ldots, N-1\}$. The reason why the last component is treated differently is explained in section 4.2.

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 (9) of roughly 44% is targeted by Roberts and Rosenthal (2009). The proposal variances in (10) can thus be adjusted differently for each of the parameters, to better approximate the target density. Concretely this is done after $b = 30$ thinnings for the parameters. When we have more than $0.5b$ acceptances of that particular parameter or cluster, the corresponding proposal variances are increased by 25%, when it is lower than $0.4b$ 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.

In case of 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 `thinning` argument in the function call indicates after how many iterations the values need to be stored, `chain` denotes the total number of stored values (burn-in included) and `burn-in` the final fraction of the stored values that will be available in the output. The following settings are applied to the LC-MS example:

```
chain=700
thin=10
burnin=0.5
```

When the estimation with the provided number of iterations is executed, the **C** code writes a number of files (for a complete listing, see Table 3) which contain all the important information. The file `amcmcvals.txt` contains the following vectors:

$$(a_1, \dots a_Q, w_{l,1}, \dots, w_{l,Q}, w_{u,1}, \dots, w_{u,Q}, \lambda_{1,1}, \dots, \lambda_{N-1,1}, \dots, \lambda_{1,Q}, \dots, \lambda_{N-1,Q}, \sigma^2), \qquad (11)$$

where each row corresponds to a stored iteration. In the above example 250 values are available in the output file, while 5000 iterations were performed. The file `amcmcamp.txt` consists of the 250 iterated variances of the random amplitude coefficients, that is, $\sigma_1^2, \dots, \sigma_K^2$. The file `amcmcker.txt` contains the kernel parameters $(a_{l,1}, \bar{a}_1, a_{u,1}, \dots, a_{l,K}, \bar{a}_K, a_{u,K})$. The output file `amcmcsigmas.txt` consists of a single line, namely the proposal variances corresponding to each of the parameters in (11), since these values are no longer updated after the burn-in stage. During the estimation routine these files are accessed by the program, they should therefore only be opened, renamed or removed afterwards.

Usually we are interested in a point estimate of the warping parameters in the model rather than in a sample from the posterior. The output list in **R** contains the warping parameters in the iteration corresponding to the highest posterior density value (as shown in Table 2).

To facilitate the decision on whether to stop the warping process or to further improve the warp by adding a new warplet, we wish to keep the information (and in particular the produced output files) of the previous model as well. Therefore we work with two series of files (e.g. `amcmcvals.txt` and `amcmcvals2.txt`), in which `amcmcvals.txt` contains the vectors for the last estimated model in case of an odd number of components and `amcmcvals2.txt` the values for the last estimated model in case of an even number of components. In the output list, the point estimates of the previous model are included and indicated by means of the '.prev' suffix).

4.2 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 highly 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 (10) a separate Metropolis-Hastings evaluation step (9) is not necessary and rather a cluster of proposals is created as in Section 4.1.

4.3 Model selection

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 the latter scenarios, the model selection step suggests to drop this additional component and opts for a reprise of the previously estimated model.

We use 95% highest posterior density intervals to represent the posterior densities in this criterion which are calculated using the function `hpd` in the **R**-library `boa` (Smith, 2007).

The function `MRwarp` offers a choice between two strategies regarding the number of warplets. First, a fixed number of warplets can be selected (e.g. 3) and the program will perform the prior-posterior tranfer estimation procedure, without displaying any intermediate results. After completion the R-output vector contains only information on the last estimated model (as also explained in Table 2). This strategy is selected by choosing:

```
selection="FIXED"
components=3
```

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 each estimated model. To facilitate this decision the current warped curves (including horizontal

preprocessing) are plotted. After the first model estimation (1 warplet) also the horizontally scaled and shifted curves and the horizontally and vertically scaled and shifted curves of the preprocessing step are plotted. In addition, the user can consult the hpd intervals, which will be automatically activated.

4.4 Example

This section contains the complete *R*-code for the LC-MS example.

Reading the data

```
TIC <- as.matrix(TICdata)
```

Smoothing the LC-MS data

```
library("SemiPar")

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

Multiresolution warping options

```
Xdata=TICx
Ydata=TICy
chain=700
thin=10
burnin=0.5
selection="STEP"
components=1
kernel.s=c(0,20,40,70,100,130,270,285,300)
prepr=c(1,1,0,0)
thresh=0.5
threshd=1/15
alpha=0.05
output <-MRwarp(Xdata,Ydata,chain,thin,burnin,kernel.s,components,selection,
            thresh,threshd,prepr,alpha=0.05)
```

Plot of the warped data

```
x <- 1:400
TIC.plot <- matrix(0,11,400)
```

```
WX <- t(matrix(x,400,11))
shift <- output$shift.w
scale <- output$scale.w

WX <- (WX -200)*scale +200 +shift

for (i in 1:11)
{
r1 <- output$A.prev - output$Ll.prev
r2 <- output$Ul.prev - output$A.prev
WX[i,] <- warp(output$A.prev,output$Lambda.prev[i,],r1,r2,WX[i,])
wx <- WX[i,]
TIC.sm <- spm(TIC[i,]~f(wx))
TIC.plot[i,] <- TIC.sm$fit$fitted
}

windows()
plot(WX[1,],TIC[1,],xlab="",ylab="",ylim=c(min(TIC),max(TIC)))
lines(WX[1,],TIC.plot[1,])
for (i in 2:11)
{
points(WX[i,],TIC[i,],xlab="",ylab="",col=i)
lines(WX[i,],TIC.plot[i,],col=i)
}
```

Output and answers provided to the program during the stepwise procedure

```
"start C loop"
"finish C loop"
Do you want to continue and add a component? (y/n)y
"start C loop"
"finish C loop"
Do you want to continue and add a component? (y/n)y
"start C loop"
"finish C loop"
Do you want to continue and add a component? (y/n)y
"start C loop"
"finish C loop"
Do you want to continue and add a component? (y/n)y
"start C loop"
"finish C loop"
"automatic model selection selects model with  4  components"
Do you want to continue and add a component? (y/n)n
"program stopped after  5  components"
```

After preprocessing and estimating the first model, containing one warplet, we receive the three plots (a), (b) and (c) in Figure 4 and the following question: `Do you want to continue and`

`add a component? (y/n)` Since we did not include any amplitude preprocessing, plot (a) and (b) in Figure 4 are exactly the same. In case one would include a vertical shift or rescaling, this will be included while estimating but will only be illustrated in plot (b). 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. We continue until we are satisfied with the result, which in this example is at $Q = 4$. This is also indicated by the model selection criteria after including a fifth component. Since this model has an even number of components, its corresponding output can be found in the files with the suffix 2. In the output list both the warping parameters for the model with four (.prev suffix) and five components are provided. The code for the plot of the warped data (Figure 1 (b)) illustrates how to use the output list to obtain warped data curves.

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.

Thresholding on the warping intensities and the warping domain can be used to avoid too severe transformations and to promote 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 in the output files and can be consulted when desired, while the most important output is directly transferred to **R**.

A Additional settings in `interface.C`

Table 4 summarizes a selection of additional settings that can be easily altered in the **C** file. The C-file is extensively documented, hence more adjustments can always be made. To compile the `.c` file and obtain the `.dll` file in Windows, we have used Rtools, a collection of packages put together by Prof. B. Ripley, currently maintained by D. Murdoch. We refer to the extensive documentation on `http://www.murdoch-sutherland.com/Rtools/`. For use with Linux, this is not necessary.

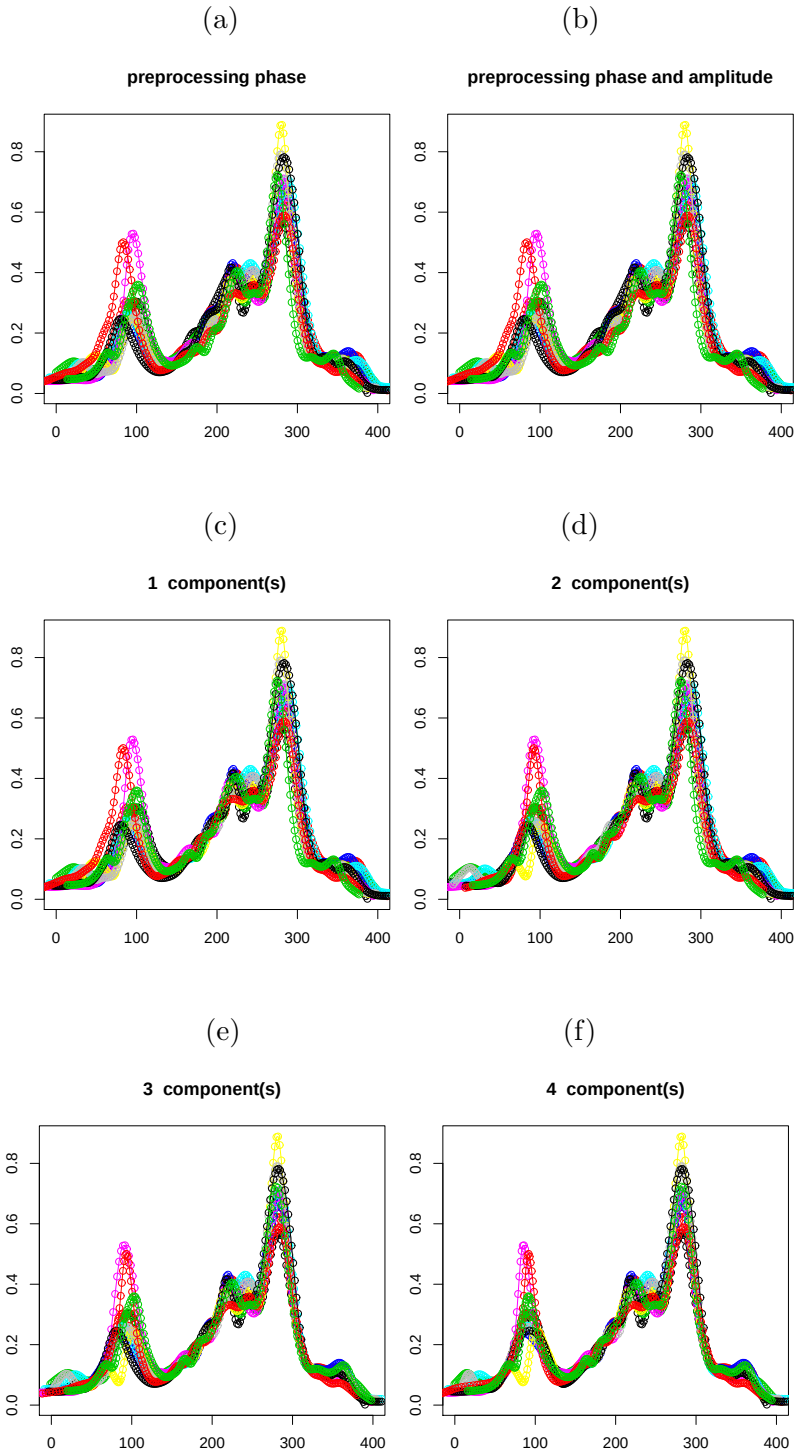

Figure 4: (a) Horizontally shifted an rescaled curves, (b) transformed curves after the entire preprocessing step (c)-(d)-(e)-(f) stepwise warped curves (horizontally shifted, rescaled and warping components). The plotted lines linearly interpolate the data.

Parameter	Line	Description
Nmax	11	The maximum number of time points N.
Smax	12	The maximum number of curves S.
MAXc	17	The maximum number of components Q.
MAXk	19	The maximum number of kernels K.
wshiftr	176	The maximum horizontal shift in the preprocessing step.
wscaler	177	The maximum horizontal rescaling in the preprocessing step.
ashiftr	558	The maximum vertical shift in the preprocessing step.
ascaler	561	The maximum vertical rescaling in the preprocessing step.
u	553	The number of thinning loops after which the proposal variances are updated (currently 30).
Npreproc	554	The number of iterations in the preprocessing step (currently $S \times 100$).
*gamma	564-565	The scale and shape parameter of the inverse gamma prior for the variances (currently 0.01) .

Table 4: Some easily adapted additional settings in the **C** file.

References

Claeskens, G., B. W. Silverman, and L. Slaets (2010). A multiresolution approach to time warping achieved by a bayesian prior-posterior transfer fitting strategy. *J. R. Stat. Soc. Ser. B Stat. Methodol.* x(x), x. to appear.

Gervini, D. and T. Gasser (2005). Nonparametric maximum likelihood estimation of the structural mean of a sample of curves. *Biometrika 92*(4), 801–820.

Hall, P., H.-G. Müller, and J.-L. Wang (2006). Properties of principal component methods for functional and longitudinal data analysis. *Ann. Statist. 34*(3), 1493–1517.

James, G. M. (2007). Curve alignment by moments. *The Annals of Applied Statistics 1*(2), 480–501.

Kneip, A. and T. Gasser (1992). Statistical tools to analyze data representing a sample of curves. *The Annals of Statistics 20*(3), 1266–1305.

Listgarten, J., R. M. Neal, S. T. Roweis, and A. Emili (2005). Multiple alignment of continuous

time series. In L. K. Saul, Y. Weiss, and L. Bottou (Eds.), *Advances in Neural Information Processing Systems 17*, pp. 817–824. Cambridge, MA: MIT Press.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Ramsay, J. O. and X. Li (1998). Curve registration. *Journal of the Royal Statistical Society, Series B 60*(2), 351–363.

Ramsay, J. O. and B. W. Silverman (2002). *Applied Functional Data Analysis* (2 ed.). Spinger-Verlag, New York.

Roberts, G. O. and J. S. Rosenthal (2009). Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics 18*(2), 349–367.

Rønn, B. B. (2001). Nonparametric maximum likelihood estimation for shifted curves. *J. R. Stat. Soc. Ser. B Stat. Methodol. 63*(2), 243–259.

Silverman, B. W. (1995). Incorporating parametric effects into functional principal components analysis. *Journal of the Royal Statistical Society, Series B 57*(4), 673–689.

Smith, B. J. (2007). boa: An r package for mcmc output convergence assessment and posterior inference. *Journal of Statistical Software 21*(11), 1–37.

Wand, M. (2010). *SemiPar: Semiparametic Regression*. R package version 1.0-3.

Wang, K. and T. Gasser (1997). Alignment of curves by dynamic time warping. *Ann. Statist. 25*(3), 1251–1276.