Biostatistics (2014), **15**, 4, *pp.* 603–619
doi:10.1093/biostatistics/kxu006
Advance Access publication on March 11, 2014

Shape outlier detection and visualization for functional data: the outliergram

ANA ARRIBAS-GIL*, JUAN ROMO
Departamento de Estadística, Universidad Carlos III de Madrid, Getafe 28903, Spain
ana.arribas@uc3m.es

Summary

We propose a new method to visualize and detect shape outliers in samples of curves. In functional data analysis, we observe curves defined over a given real interval and shape outliers may be defined as those curves that exhibit a different shape from the rest of the sample. Whereas magnitude outliers, that is, curves that lie outside the range of the majority of the data, are in general easy to identify, shape outliers are often masked among the rest of the curves and thus difficult to detect. In this article, we exploit the relationship between two measures of depth for functional data to help to visualize curves in terms of shape and to develop an algorithm for shape outlier detection. We illustrate the use of the visualization tool, the outliergram, through several examples and analyze the performance of the algorithm on a simulation study. Finally, we apply our method to assess cluster quality in a real set of time course microarray data.

Keywords: Depth for functional data; Outlier visualization; Robust estimation; Time course microarray data.

1. Introduction

In many biomedical and public health studies, individual observations are real functions of time, observed at discrete time points. Each curve provides the evolution of a certain process of interest for a given individual. If the process is known to be continuous and smooth, curves can be treated as functional data. Examples include human growth curves (see, e.g. Ramsay and Silverman, 2005), time-course microarray experiments (Rong and Müller, 2009), mortality and fertility rates (Hyndman and Ullah, 2007), or pollutants' concentration across time (Febrero *and others*, 2008), among others. In this context, as in the univariate or the multivariate case, the presence of atypical observations may affect the statistical analysis of the data. Indeed, in time course microarray analysis, for instance, the sensitivity of regression techniques to outliers when trying to identify genes whose expression time profiles depend on a factor is a known problem (Sohn *and others*, 2010). This reflects the necessity of including an outlier detection step in any preliminary analysis of functional data. The challenge in the functional framework is that outlying observations might be difficult to identify visually. Indeed, an outlier trajectory is not only one that contains atypically high or low values, but also a trajectory that even containing average levels across the whole observation interval may present a different shape or pattern than the rest of the curves of the sample.

*To whom correspondence should be addressed.

Following Hyndman and Shang (2010), we will refer to the first type of atypical curves as magnitude outliers and to the second one as shape outliers. Whereas magnitude outliers might be easily detected even with a simple visual inspection of the data, shape outliers will be *hidden* in the middle of the sample and its identification will not be straightforward. That is, a shape outlier is not an observation that lies far away from the sample in terms of the Euclidean distance. Thus, measures that can deal with shape features need to be considered in order to be able to distinguish them from the rest of the curves. In this article, we combine two sources of information about curve shape given by two measures of depth for functional data, the modified band depth (MBD) (López-Pintado and Romo, 2009) and the modified epigraph index (MEI) (López-Pintado and Romo, 2011). Each one provides an ordering of the sample according to different shape features and the relationship between them sheds light on shape variation across the sample.

Different methods for outlier detection in functional data have been developed during the last years. Among them, several rely on different notions of functional depth (Febrero *and others*, 2008; Sun and Genton, 2011; Gervini, 2012; Martín-Barragán *and others*, 2012), on robust principal components (Hyndman and Shang, 2010), or on random projections of infinite-dimensional data into $\mathbb{R}$ (Fraiman and Svarc, 2013). Also, some distributional approaches have been considered (Gervini, 2009). While some of these methods may only be sensitive to magnitude outliers, most of them work efficiently at detecting different kinds of outlying trajectories. However, the mechanisms they rely on do not always allow one to provide an easy interpretation on why an observation is considered an outlier. In this sense, the objective of this article is 2-fold. On the one hand, we propose an algorithm for shape outlier detection. On the other hand, we give a visualization tool that helps in understanding shape variation across the sample and provides additional information that can be used to correct the output of the algorithm.

The rest of the article is as follows. In Section 2, we review the definitions of MBD and MEI and investigate their relationship on a sample of curves. Based on it, in Section 3 we propose a visualization tool that helps identifying curves with different shape and a rule for shape outliers detection. We evaluate our method and previous existing techniques for outlier detection in functional data through an extensive simulation study in Section 4. In Section 5, we apply our method in the assessment of cluster analysis on a real data set of yeast time course microarrays. Finally, we conclude the article with a discussion in Section 6.

2. MBD AND MEI

Let us introduce the MBD and the MEI, as firstly defined in López-Pintado and Romo (2009) and López-Pintado and Romo (2011), respectively. Both measures provide an idea of how central or deep a curve is with respect to a sample of curves. Let $x_1, \ldots, x_n$ be n continuous functions defined on a given compact real interval $\mathcal{I}$. For any $x \in \{x_1, \ldots, x_n\}$, we define its MBD as

$$\mathrm{MBD}_{\{x_1,\ldots,x_n\}}(x) = \binom{n}{2}^{-1} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{\lambda(\{t \in \mathcal{I} \mid \min(x_i(t), x_j(t)) \leqslant x(t) \leqslant \max(x_i(t), x_j(t))\})}{\lambda(\mathcal{I})},$$

where $\lambda(\cdot)$ stands for the Lebesgue measure on $\mathbb{R}$. If, for each pair of curves x_i and x_j in the sample, we consider the band that they define in $\mathcal{I} \times \mathbb{R}$ as $\{(t, y) \mid t \in \mathcal{I}, \min(x_i(t), x_j(t)) \leqslant y \leqslant \max(x_i(t), x_j(t))\}$, then $\mathrm{MBD}_{\{x_1,\ldots,x_n\}}(x)$ represents the mean over all possible bands of the proportion of time that $x(t)$ spends inside a band. The MBD is an extension of the original band depth that accounts for the proportion of bands in which a curve is entirely contained (see López-Pintado and Romo, 2009, for details).

The MEI of $x \in \{x_1, \ldots, x_n\}$ is defined as

$$\mathrm{MEI}_{\{x_1,\ldots,x_n\}}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{\lambda(\{t \in \mathcal{I} \mid x_i(t) \geqslant x(t)\})}{\lambda(\mathcal{I})},$$

and it stands for the mean proportion of time that x lies below the curves of the sample. As in the case of the MBD, the MEI is a generalization of the epigraph index that accounts for the proportion of curves that lie entirely above x (López-Pintado and Romo, 2011).

It is clear that MBD and MEI are closely related quantities and that investigating the relation between these two measures might shed light on shape characterization of curves. Indeed, consider a curve with an MEI close to 0.5. That would mean that the curve is located in the center of the sample in terms of level variation. Then, if the curve's shape is similar to the shapes of the rest of the curves in the sample, one may expect to get a high value for its MBD, since the curve should be contained in many bands defined by other curves. However, if one gets a low MBD value, that would indicate that the curve is contained in a small number of bands, even if it is placed in the center of the sample in terms of level. That can only mean that the curve exhibits a shape very different from those of the rest of the curves.

In this section, we investigate the relationship between these two measures in order to characterize shape outlyingness. The following lemma gives an expression of the MBD in terms of MEI, whose proof can be found in supplementary material available at *Biostatistics* online.

LEMMA 2.1 Let $x_1, \ldots, x_n$ be n continuous functions on $\mathcal{I}$. Then, for any $x \in \{x_1, \ldots, x_n\}$,

$$\mathrm{MBD}_{\{x_1,\ldots,x_n\}}(x) = a_0 + a_1 \mathrm{MEI}_{\{x_1,\ldots,x_n\}}(x) + a_2 \left[\sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\lambda(E_{i,x} \cap E_{j,x})}{\lambda(\mathcal{I})} \right],$$

with $a_0 = a_2 = -2/n(n-1)$, $a_1 = 2(n+1)/(n-1)$, and $E_{i,x} = \{t \in \mathcal{I} \mid x_i(t) \geqslant x(t)\}$.

THEOREM 2.2 Let $x_1, \ldots, x_n$ be n continuous functions on $\mathcal{I}$. Then, for any $x \in \{x_1, \ldots, x_n\}$:

(a)

$$\mathrm{MBD}_{\{x_1,\ldots,x_n\}}(x) = a_0 + a_1 \mathrm{MEI}_{\{x_1,\ldots,x_n\}}(x) + a_2 n^2 \mathrm{MEI}_{\{x_1,\ldots,x_n\}}(x)^2 + a_2 \left[\sum_{i=1}^{n} \sum_{j=1}^{n} \left(\frac{\lambda(E_{i,x} \cap E_{j,x})}{\lambda(\mathcal{I})} - \frac{\lambda(E_{i,x})\lambda(E_{j,x})}{\lambda(\mathcal{I})^2} \right) \right].$$

(b)

$$\mathrm{MBD}_{\{x_1,\ldots,x_n\}}(x) \leqslant a_0 + a_1 \mathrm{MEI}_{\{x_1,\ldots,x_n\}}(x) + a_2 n^2 \mathrm{MEI}_{\{x_1,\ldots,x_n\}}(x)^2. \tag{2.1}$$

(c) Moreover, if $(x_i(t_1) - x_j(t_1))(x_i(t_2) - x_j(t_2)) > 0$, $t_1, t_2 \in \mathcal{I}$, $i \neq j$, then

$$\mathrm{MBD}_{\{x_1,\ldots,x_n\}}(x) = a_0 + a_1 \mathrm{MEI}_{\{x_1,\ldots,x_n\}}(x) + a_2 n^2 \mathrm{MEI}_{\{x_1,\ldots,x_n\}}(x)^2.$$

Proof. (a) The equality follows from Lemma 2.1 and from the fact that $\mathrm{MEI}_{\{x_1,\ldots,x_n\}}(x) = (1/n) \sum_{i=1}^{n} (\lambda(E_{i,x})/\lambda(\mathcal{I}))$.

(b) Let $a_x(t)$ be the number of curves above or equal to x at time t. Then

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \left(\frac{\lambda(E_{i,x} \cap E_{j,x})}{\lambda(\mathcal{I})} - \frac{\lambda(E_{i,x})\lambda(E_{j,x})}{\lambda(\mathcal{I})^2} \right) = \int_{\mathcal{I}} a_x(t)^2 \frac{1}{\lambda(\mathcal{I})} \, \mathrm{d}t - \left(\int_{\mathcal{I}} a_x(t) \frac{1}{\lambda(\mathcal{I})} \, \mathrm{d}t \right)^2 \geqslant 0,$$

by Jensen's inequality. Now, since $a_2 < 0$, the inequality holds from (a).

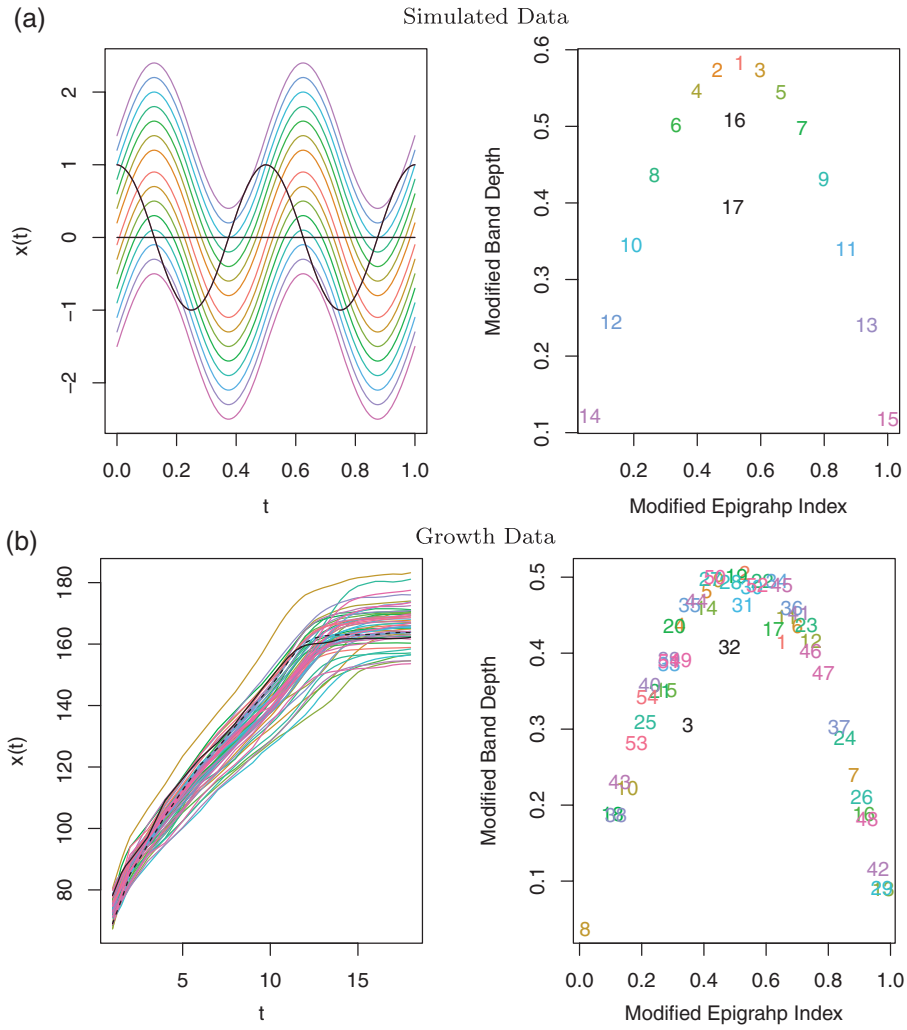

Fig. 1. (a) Left: a sample of 17 curves of the form $x_i(t) = sin(4\pi t) + (-1)^i(i/10)$, $i = 1, \ldots, 15$, $x_{16}(t) = 0$, and $x_{17}(t) = \cos(4\pi t)$, $t \in [0, 1]$. Curves x_{16} and x_{17} are displayed in black. Right: MBD vs. MEI of the 17 curves. (b) Left: height curves of 54 girls during ages between 0 and 18 years. Curves x_3 and x_{32} are displayed in black with solid and dashed lines, respectively. Right: MBD vs. MEI of the 54 curves.

(c) If $(x_i(t_1) - x_j(t_1))(x_i(t_2) - x_j(t_2)) > 0$, $t_1, t_2 \in \mathcal{I}$, $i \neq j$, then $E_{i,x} = \emptyset$ or $E_{i,x} = \mathcal{I}$ for all i and so, $\lambda(E_{i,x} \cap E_{j,x})/\lambda(\mathcal{I})$ and $\lambda(E_{i,x})\lambda(E_{j,x})/\lambda(\mathcal{I})^2$ are equal for all i, j. That is, the fourth term in the right-hand side of (a) is zero and so the equality holds. □

An underlying idea below the statement of Theorem 2.2 is that if, in a sample of perfectly aligned curves with common shape, one introduces a curve with a different pattern, then the $\mathbb{R}^2$ point corresponding to the pair (MEI, MBD) for this new curve will lie far away from the parabola defined by the points corresponding to the rest of the curves. Figure 1(a) provides an example of this phenomenon, in which it is very easy to detect the outlying observations by looking at the MBD vs. MEI characteristics. In general, however, trajectories of a random process will cross many times even if they all exhibit the same trend pattern.

Then, the (MEI, MBD) points will not define a perfect parabola and identifying outlying trajectories will not be straightforward. Let us consider as an illustration the height curves of 54 girls coming from the well-known Berkeley growth data set (see, e.g. Ramsay and Silverman, 2005). They are presented in Figure 1(b) together with their MBD vs. MEI characteristics. The points corresponding to the third and 32nd girls are far from the rest of the points. These two girls exhibit a different growth pattern from the rest of the girls in the sense that they were ones tallest of the sample during childhood, especially true for girl number 3, but they stopped growing earlier than the rest of the girls and ended up with a height in the lower quartile at 18 years of age. Girl number 32, in addition, was one of the smallest girls at birth and exhibits a very high growth rate during the first years of her life. To what extent we can consider these two trajectories as outliers is something that can be addressed by considering the problem of outlier detection in the MBD vs. MEI plane as we do in the next section.

3. Shape outlier detection algorithm and outliergram

Based on the relation between the MBD and the MEI, we now propose to use the MBD vs. MEI plane as a visualization tool and we give an algorithm to detect shape outliers. From Theorem 2.2, we know that all the (MEI, MBD) points lie below the parabola given by (2.1) and that the closest points to the parabola correspond to curves with typical shape, whereas the most distant ones represent outlying curves in terms of shape. This motivates the use of the univariate boxplot rule for outlier detection on the vertical distances to the parabola. That is, given a sample of curves $x_1, \ldots, x_n$ with $mb_i = \mathrm{MBD}_{x_1,\ldots,x_n}(x_i)$ and $me_i = \mathrm{MEI}_{x_1,\ldots,x_n}(x_i)$ for $i = 1, \ldots, n$, we consider the distances $d_i = a_0 + a_1 me_i + n^2 a_2 me_i^2 - mb_i$ and define as shape outliers those curves with $d_i \geqslant Q_{d3} + 1.5\mathrm{IQR}_d$, where Q_{d3} and IQR_d are the third quartile and inter-quartile range of $d_1, \ldots, d_n$. In addition to the outlier detection rule, it might be interesting to assess how distant the outliers are from the rest of the sample or to identify curves that might be close to the outlier region although not inside. To jointly visualize the observations in terms of shape and the boundary between the outlying and non-outlying curves, we propose to represent in $\mathbb{R}^2$ the (MEI, MBD) points together with the parabola shifted downward by $Q_{d3} + 1.5\mathrm{IQR}_d$. We will refer to this graphical representation as the outliergram.

Although the outliergram has not been conceived to detect magnitude outliers, note that some particular magnitude outliers, the ones that lie below or above the majority of the curves along most of the time interval, will appear at the bottom corners of the plot. The gap between them and the contiguous observations (according to the order induced by the MEI) might be an indicator of their outlyingness. However, we do not pretend to give any specific rule to detect them, since very good mechanisms for this purpose already exists, such as the functional boxplot defined in Sun and Genton (2011) (see also Martín-Barragán *and others*, 2012). Indeed, we propose to combine that algorithm with our shape outlier detection rule and we provide code that does so (see supplementary material available at *Biostatistics* online).

It is worth noting that the precedent reasoning for shape outlier detection might fail with curves that lie above or below the majority of the curves in the sample, that is, with MEI values close to 0 or 1. Indeed, for such curves the MBD will always be low, since they are surrounded by very few curves, independently of the fact that they might present an atypical shape or not. However, if the curve presented a typical shape and we shifted it vertically toward the center of the sample, its MBD in the new location should increase (as the MEI increases or decreases). On the other hand, if the curve's shape was atypical, even when placed in the center of the sample, its MBD would remain low. That motivates the addition of a second step in the shape outlier detection procedure in which the more extreme curves (see below for a proper definition) are vertically shifted toward the center of the sample one by one. They would be considered shape outliers if the new (MBD, MEI) point lies in the outlying region previously determined. In that case the outliergram will show both the old and the new (MBD, MEI) points, using a different symbol for the last one to help in distinguishing them (see Figures 2 and 4 for some examples).

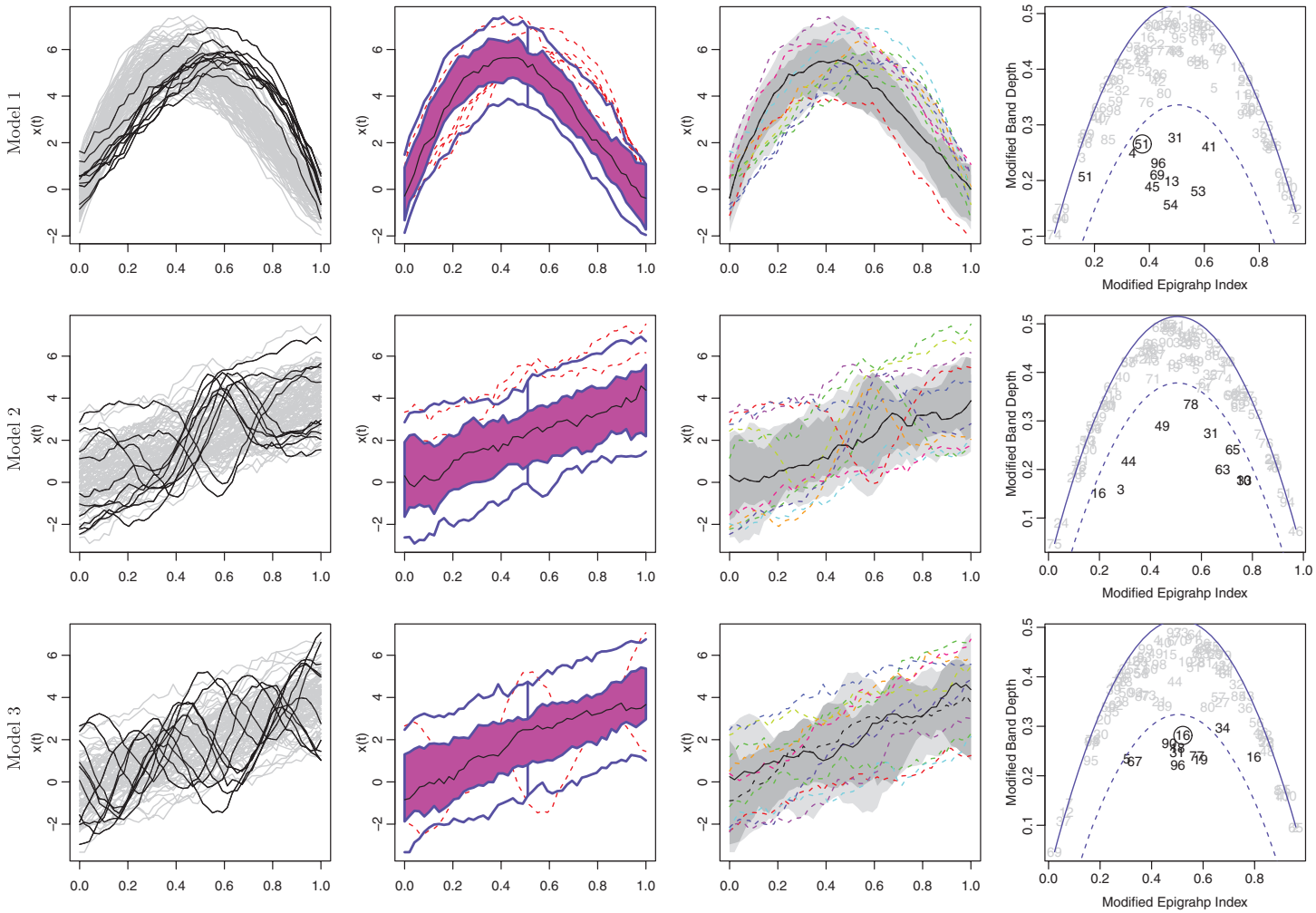

Fig. 2. Three simulation runs from models 1, 2, and 3 with $n = 100$ and $c = 0.1$. First column: curves generated from the main model (gray) and the contamination model (black). Second column: adjusted functional boxplot (outliers are dashed lines). Third column: functional HDR boxplot (outliers are dashed lines, the solid line is the mode). Fourth column: adjusted outliergram (outliers correspond to the points below the dashed parabola; the code color is the same of the first column; circles stand for curves that have been considered outliers after having been shifted vertically toward the center of the sample).

Then, given a sample of curves $x_1, \ldots, x_n$, the whole shape outlier detection algorithm is as follows:

1. Compute $mb_i = \mathrm{MBD}_{x_1,\ldots,x_n}(x_i)$, $me_i = \mathrm{MEI}_{x_1,\ldots,x_n}(x_i)$, and $P_i = a_0 + a_1 me_i + n^2 a_2 me_i^2$, for $i = 1, \ldots, n$, where a_0, a_1, and a_2 are the ones given in Lemma 2.1.
2. Compute $d_i = P_i - mb_i$ for $i = 1, \ldots, n$.
3. Compute the third quartile and inter-quartile range of the sample $d_1, \ldots, d_n$, Q_{d3}, and IQR_d.
4. Shape outlier identification: $\mathrm{SO} = \{i \mid mb_i \leqslant P_i - Q_{d3} - 1.5\mathrm{IQR}_d\}$.
5. For $i \in \{1, \ldots, n\} \setminus \mathrm{SO}$:
   - If $x_i(t) < \min_{j \neq i} x_j(t)$ for some $t \in \mathcal{I}$, define $\tilde{x}_i(t) = x_i(t) - \min_s\{x_i(s) - \min_{j \neq i} x_j(s)\}$.
   - If $x_i(t) > \max_{j \neq i} x_j(t)$ for some $t \in \mathcal{I}$, define $\tilde{x}_i(t) = x_i(t) - \max_s\{x_i(s) - \max_{j \neq i} x_j(s)\}$.
   - Compute $\widetilde{mb}_i = \mathrm{MBD}_{x_1,\ldots,\tilde{x}_i\ldots,x_n}(\tilde{x}_i)$, $\widetilde{me}_i = \mathrm{MEI}_{x_1,\ldots,\tilde{x}_i\ldots,x_n}(\tilde{x}_i)$ and $\widetilde{P}_i = a_0 + a_1\widetilde{me}_i + n^2 a_2 \widetilde{me}_i^2$. If $\widetilde{mb}_i \leqslant \widetilde{P}_i - Q_{d3} - 1.5\mathrm{IQR}_d$, then $\mathrm{SO} = \mathrm{SO} \cup \{i\}$.

4. Simulation study

In this section, we compare the performance of the proposed procedure with several functional and/or multivariate outlier detection methods through an extensive simulation study. Namely, we consider eight different techniques developed during the last decade (the numbers below stand only for reference purposes).

1. Functional boxplot (Sun and Genton, 2011): considering the center outward ordering induced by band depth or MBD in a sample of curves, a boxplot is constructed by defining the envelope of the 50% central region.The maximum non-outlying envelope is obtained by inflating that central region 1.5 times as in a univariate boxplot. Any curve lying partially or globally outside that maximum non-outlying envelope is considered an outlier.
2. Adjusted functional boxplot (Sun and Genton, 2012): The 1.5 constant factor in the functional boxplot (method 1) can be replaced by a different quantity with the aim of controlling the probability of correctly detecting no outliers. The method consists in simulating observations without outliers on the basis of a robust estimator of the covariance function of the data. The factor is then selected as the one for which, when used with the functional boxplot, the probability of detecting no outliers is the closest to 0.993. The selected factor is applied to the functional boxplot of the original data.
3. Functional highest density region (HDR) boxplot (Hyndman and Shang, 2010): a functional boxplot is obtained by constructing a bivariate HDR boxplot (Hyndman, 1996) with the first two robust principal component scores. The coverage probability of the outlying region, that is, the proportion of curves that will be classified as outliers, needs to be prespecified.
4. Robust Mahalanobis Distance: considering the curves as multivariate observations, the robust Mahalanobis distance between each curve and the pointwise sample mean is computed. Outliers are defined as observations that have squared robust Mahalanobis distances greater than the 0.99 quantile of a χ^2 distribution with p degrees of freedom, where p is the fixed number of observation points in every curve (see Hyndman and Shang, 2010 and the references therein for details).
5. Integrated Squared Error (Hyndman and Ullah, 2007): the integrated squared error between each curve in the sample and its projection into a given number K of robust principal components is computed. Outliers are defined as those observations with an integrated squared error greater than a threshold. Throughout this simulation study K is chosen to be equal to 2 and the threshold is set to $s + 3.29\sqrt{s}$, where s is the median of the observed ISEs, as suggested in Hyndman and Shang (2010).

6, 7. Depth-based weighting and trimming (Febrero *and others*, 2008): considering different depth measures for functional data, the authors proposed to define as outliers the curves whose depth levels are below a cutoff. The cutoff is determined by a bootstrap procedure based either on trimming or weighting of the sample. In the first case, the proportion of potential outliers, which is used as the trimming level, needs to be prespecified.

8. Projection-based trimming (Fraiman and Svarc, 2013): the authors propose a multivariate and functional robust estimation procedure that provides an outlier detection method as a by-product. The method consists in trimming the sample based on random projections. The maximum proportion of observations to be trimmed has to be prespecified.

As for the outliergram presented in this article, we are going to consider two versions: the one described in Section 3 (referred to as 9) and an adjusted version (referred to as 10) inspired by Sun and Genton (2012) (see method 2). The idea of this adjusted algorithm is to select the factor that determines the boundary for outlying points in terms of the observed data, in order to be able to control the false positive rate. As in Sun and Genton (2012), we propose to robustly estimate the covariance matrix of the data and simulate data without outliers on the basis of a centered Gaussian process with estimated covariance function. By simulating a large enough number of data sets and applying to each one the outliergram with different factor values, we can then select the factor whose false detection rate is the closest to 0.007, which is the false detection rate of the univariate boxplot when applied to a normal distribution and is often used as a standard in the definition of alternative outlier detection rules (see, for instance, Hubert and Vandervieren, 2008). Although this procedure could be applied to determine the value of F in $Q_{3d} + F \times \mathrm{IQR}_d$, we prefer to use as limiting rule $d_i \geqslant F \times Q_{1d}$ and choose the value of F as described above. The reason for this choice is the following: the distribution of points d_i is right-skewed and outliers only appear at the right tail of it. Then, assuming the observed data contain outliers, Q_{d3} and IQR_d in the uncontaminated simulated samples would probably be smaller than their counterparts in the original sample, whereas it is reasonable to think that Q_{1d} will not significantly change. Hence, it is advisable to use the rule $d_i \geqslant F \times Q_{1d}$ for the selection of the factor on the basis of uncontaminated samples and for its posterior application to the outliergram of the observed data. In both cases, the set of candidate values for the factor needs to be specified. While for the boundary $Q_{3d} + F \times \mathrm{IQR}_d$ it seems logical to search for F around 1.5, it is not clear what the interval of candidate values should be when using the boundary $F \times Q_{1d}$. Given the distribution of points d_i of the data set to be analyzed, it is reasonable to think that in the uncontaminated data sets generated from the estimation of its covariance structure the limit between outlying and non-outlying observations could be somewhere between Q_{3d} and $c \max d_i$, where $c > 1$ to account for the case in which the original data set contains no outliers itself. Then, in the adjusted outliergram we will let the interval of candidate values for F vary for each data set to be analyzed: after computing the quantities d_i, the interval will be set to $[Q_{3d}/Q_{1d}, c \max d_i / Q_{1d}]$. See below for the particular values used through the simulation study.

For the sake of clarity and conciseness, we restrict our simulation study to these 10 methods. Comparison between these and some other related non-parametric procedures can be found in Hyndman and Shang (2010).

We have generated curves from three different models which are described next. In each case, $n - \lceil c \cdot n \rceil$ curves were generated according to the main model and the remaining $\lceil c \cdot n \rceil$ curves according to the contamination model, where, for a real number x, $\lceil x \rceil$ is the smallest integer not less than x.

- Model 1. Main model: $X(t) = 30t(1-t)^{3/2} + \varepsilon(t)$, and contamination model: $X(t) = 30t^{3/2}(1-t) + \varepsilon(t)$, where $t \in [0, 1]$ and $\varepsilon(t)$ is a Gaussian process with zero mean and covariance function $\gamma(s, t) = 0.3 \exp\{-|s-t|/0.3\}$. This model had already been used in Febrero *and others* (2008) and Fraiman and Svarc (2013).

- Model 2. Main model: $X(t) = 4t + \varepsilon(t)$, and contamination model: $X(t) = 4t + (-1)^u 1.8 + (1/\sqrt{2\pi 0.01}) \exp\{-(t-\mu)^2/0.02\} + \varepsilon(t)$, where $t \in [0, 1]$, $\varepsilon(t)$ is a Gaussian process with zero mean and covariance function $\gamma(s, t) = \exp\{-|s - t|\}$, u follows a Bernoulli distribution with probability $\frac{1}{2}$ and μ is uniformly distributed in [0.25, 0.75]. The main model had already been used in Sun and Genton (2011).
- Model 3. Main model: $X(t) = 4t + \varepsilon(t)$, and contamination model: $X(t) = 4t + 2\sin(4(t + \theta)\pi) + \varepsilon(t)$, where $t \in [0, 1]$, $\varepsilon(t)$ is a Gaussian process with zero mean and covariance function $\gamma(s, t) = \exp\{-|s - t|\}$, and θ is uniformly distributed in [0.25, 0.75].

For each one of the three models, we considered two different values for the sample size, $n = 100$ and 200, and five values for the contamination rate, $c = 0$, 0.05, 0.1, 0.15, and 0.2, including the non-contaminated model ($c = 0$). We ran 400 simulations for each combination of n and c. In each case, we sampled the curves on 50 equidistant points over the interval [0, 1]. For the procedures requiring the specification of the coverage probability of the outlying region or trimming proportion, we set those equal to the true value c. For procedures 6 and 7, we set to 200 the number of bootstrap samples and chose as depth function the modal depth (Cuevas *and others*, 2006) as advised in Febrero *and others* (2008). For the procedures 2 and 10, we equally set to 200 the number of sampled data sets for the selection of the factor in the outlier detection rule. For these methods, we used as robust estimator of the covariance the orthogonalized Gnanadesikan–Kettenring estimator proposed by Maronna and Zamar (2002). For method 2, the search interval for the factor value was [0.5, 2.5], with a distance between candidate values of 0.25. For method 10, the search was performed in a grid of 10 points equally spaced on the interval $[Q_{d3}/Q_{d1}, 1.5 \max_i d_i/Q_{d1}]$, where $d_1, \ldots, d_n$ refer to the distances on the original data set. For procedure 8, we set to 100 the maximum number of random projections. In Figure 2, we present the curves generated with each of these models in a single simulation run. In Tables 1 and 2, we show the results for the two versions of the outliergram and the other eight methods in terms of the proportion of correctly identified outliers p_c (number of correctly identified outliers over the number of outliers in the sample, or sensitivity) and the proportion of false positives p_f (number of wrongly identified outliers over the number of non-outlying curves in the sample, or false detection rate). For model 1, the method that achieves the best performance is the Robust Mahalanobis Distance whose p_c and p_f remain very high and low, respectively, across different sample sizes and contamination rates. However, this method's sensitivity for models 2 and 3 is very low. On the contrary, the ISE method performs very well on models 2 and 3 and presents slightly worse results for model 1. Let us point out that, although the sensitivity of this method is in general very high, its false detection rate is often larger than that of most of the methods. The functional boxplot has very low sensitivity in all the models, although its adjusted version correctly identifies a considerably larger proportion of outliers while barely increasing its false detection rate. The HDR functional boxplot detects more outliers than both versions of the functional boxplot, for model 1, and slightly less than the adjusted version for models 2 and 3, but, in general, it exhibits very large false detection rates. With respect to the depth-based methods (6 and 7), the trimming procedure works always better than the weighting procedure (except for $c = 0.05$, where their performances are similar) and they both present better results for $n = 200$ than for $n = 100$. Except for model 1, they seem to be quite resistant since their sensitivity remains almost constant as the contamination rate increases. That is also the case for the projection-based procedure, whose sensitivity is, however, generally low and its false detection rate quite large across all models, sample sizes, and contamination rates. With respect to the outliergram, we can see that it presents very high detection rates in the three models, especially for $c = 0.05$ and $c = 0.1$. For models 2 and 3, the sensitivity remains high for larger contamination rates, but for model 1 it decreases rapidly as the contamination rate increases (especially when going from 0.15 to 0.2). Indeed, the outliergram is not a resistant procedure, since the presence of too many outlying trajectories would decrease the MBD of all the curves in the sample making the (MEI, MBD) cloud of points too spread out to find outliers. With regard to specificity,

the outliergram presents high false detection rates particularly in the presence of none or few outliers, whereas they decrease as contamination rates increase. The adjusted outliergram, however, reduces significantly the proportion of incorrectly identified outliers with values close to the nominal level, 0.007, in the case of uncontaminated samples. This comes at the price of a slightly reduced sensitivity, especially in

Table 1. *Mean and standard deviation* (*in parentheses*) *of the proportion of correctly and falsely identified outliers in the three simulation models over* 400 *simulation runs for* $n = 100$

	Model 1		Model 2		Model 3	
Method	p_c	p_f	p_c	p_f	p_c	p_f
$n = 100, c = 0$						
1. Fun. BP	–	0.001 (0.003)	–	0.001 (0.002)	–	0.001 (0.003)
2. Adj. fun. BP	–	0.006 (0.01)	–	0.005 (0.009)	–	0.007 (0.011)
3. Fun. HDR BP	–	0.000 (0)	–	0.000 (0)	–	0.000 (0)
4. Rob. Mah. dist.	–	0.016 (0.015)	–	0.015 (0.014)	–	0.016 (0.015)
5. ISE	–	0.035 (0.019)	–	0.033 (0.018)	–	0.031 (0.018)
6. DB trimming	–	0.013 (0.008)	–	0.012 (0.007)	–	0.013 (0.007)
7. DB weighting	–	0.014 (0.012)	–	0.014 (0.012)	–	0.015 (0.012)
8. PB trimming	–	0.000 (0)	–	0.000 (0)	–	0.000 (0)
9. Outliergram	–	0.053 (0.024)	–	0.054 (0.024)	–	0.054 (0.023)
10. Adj. outliergram	–	0.011 (0.012)	–	0.011 (0.012)	–	0.012 (0.013)
$n = 100, c = 0.05$						
1. Fun. BP	0.194 (0.208)	0.000 (0.002)	0.193 (0.191)	0.001 (0.002)	0.186 (0.188)	0.000 (0.002)
2. Adj. fun. BP	0.574 (0.3)	0.008 (0.011)	0.604 (0.294)	0.007 (0.01)	0.640 (0.284)	0.008 (0.011)
3. Fun. HDR BP	0.644 (0.216)	0.019 (0.011)	0.474 (0.209)	0.028 (0.011)	0.198 (0.178)	0.042 (0.009)
4. Rob. Mah. dist.	0.973 (0.078)	0.009 (0.01)	0.366 (0.239)	0.010 (0.012)	0.100 (0.142)	0.012 (0.013)
5. ISE	0.906 (0.244)	0.032 (0.018)	1.000 (0)	0.034 (0.021)	1.000 (0)	0.031 (0.019)
6. DB trimming	0.95 (0.157)	0.008 (0.009)	0.995 (0.031)	0.007 (0.007)	1.000 (0)	0.005 (0.007)
7. DB weighting	0.906 (0.212)	0.007 (0.008)	0.994 (0.034)	0.011 (0.01)	1.000 (0)	0.012 (0.011)
8. PB trimming	0.375 (0.199)	0.033 (0.01)	0.394 (0.193)	0.032 (0.01)	0.185 (0.156)	0.043 (0.008)
9. Outliergram	0.998 (0.017)	0.035 (0.019)	1.000 (0)	0.034 (0.02)	1.000 (0)	0.034 (0.021)
10. Adj. outliergram	0.983 (0.063)	0.008 (0.011)	0.984 (0.063)	0.008 (0.01)	1.000 (0)	0.009 (0.012)
$n = 100, c = 0.1$						
1. Fun. BP	0.133 (0.139)	0.000 (0.002)	0.167 (0.142)	0.000 (0.002)	0.170 (0.155)	0.000 (0.002)
2. Adj. fun. BP	0.551 (0.256)	0.006 (0.01)	0.635 (0.24)	0.008 (0.011)	0.661 (0.226)	0.007 (0.01)
3. Fun. HDR BP	0.624 (0.16)	0.042 (0.018)	0.586 (0.134)	0.046 (0.015)	0.321 (0.154)	0.075 (0.017)
4. Rob. Mah. dist.	0.95 (0.103)	0.004 (0.007)	0.367 (0.18)	0.007 (0.009)	0.118 (0.134)	0.010 (0.011)
5. ISE	0.872 (0.277)	0.026 (0.018)	1.000 (0)	0.033 (0.022)	1.000 (0)	0.030 (0.019)
6. DB trimming	0.742 (0.371)	0.009 (0.01)	0.997 (0.017)	0.010 (0.009)	1.000 (0)	0.006 (0.008)
7. DB weighting	0.124 (0.202)	0.002 (0.004)	0.962 (0.092)	0.006 (0.008)	0.992 (0.057)	0.006 (0.008)
8. PB trimming	0.455 (0.136)	0.061 (0.015)	0.513 (0.122)	0.054 (0.014)	0.295 (0.134)	0.078 (0.015)
9. Outliergram	0.989 (0.038)	0.023 (0.017)	0.998 (0.015)	0.016 (0.014)	1.000 (0)	0.021 (0.016)
10. Adj. outliergram	0.923 (0.106)	0.005 (0.009)	0.983 (0.049)	0.006 (0.009)	1.000 (0)	0.008 (0.011)
$n = 100, c = 0.15$						
1. Fun. BP	0.091 (0.104)	0.000 (0.001)	0.135 (0.114)	0.000 (0.002)	0.131 (0.11)	0.000 (0.002)
2. Adj. fun. BP	0.498 (0.206)	0.005 (0.008)	0.623 (0.208)	0.006 (0.01)	0.618 (0.192)	0.006 (0.01)

Continued

Table 1. *Continued*

Method	Model 1		Model 2		Model 3	
	p_c	p_f	p_c	p_f	p_c	p_f
3. Fun. HDR BP	0.633 (0.127)	0.065 (0.022)	0.669 (0.103)	0.058 (0.018)	0.432 (0.134)	0.100 (0.024)
4. Rob. Mah. dist.	0.935 (0.119)	0.001 (0.004)	0.335 (0.168)	0.003 (0.007)	0.136 (0.14)	0.007 (0.01)
5. ISE	0.753 (0.344)	0.027 (0.018)	0.999 (0.011)	0.041 (0.026)	1.000 (0)	0.030 (0.02)
6. DB trimming	0.446 (0.391)	0.010 (0.012)	0.998 (0.01)	0.012 (0.011)	1.000 (0)	0.009 (0.009)
7. DB weighting	0.021 (0.038)	0.001 (0.003)	0.747 (0.276)	0.002 (0.005)	0.682 (0.372)	0.002 (0.005)
8. PB trimming	0.495 (0.117)	0.089 (0.021)	0.576 (0.103)	0.075 (0.018)	0.353 (0.1)	0.114 (0.018)
9. Outliergram	0.895 (0.107)	0.012 (0.013)	0.988 (0.033)	0.008 (0.01)	1.000 (0.003)	0.008 (0.01)
10. Adj. outliergram	0.66 (0.198)	0.003 (0.006)	0.967 (0.074)	0.005 (0.008)	1.000 (0.005)	0.008 (0.013)
$n = 100, c = 0.2$						
1. Fun. BP	0.048 (0.066)	0.000 (0)	0.111 (0.105)	0.000 (0.002)	0.100 (0.088)	0.000 (0.001)
2. Adj. fun. BP	0.356 (0.205)	0.003 (0.006)	0.564 (0.187)	0.005 (0.008)	0.567 (0.185)	0.004 (0.008)
3. Fun. HDR BP	0.607 (0.108)	0.098 (0.027)	0.721 (0.088)	0.070 (0.022)	0.538 (0.125)	0.115 (0.031)
4. Rob. Mah. dist.	0.856 (0.173)	0.000 (0.001)	0.299 (0.162)	0.001 (0.004)	0.143 (0.143)	0.005 (0.009)
5. ISE	0.536 (0.387)	0.027 (0.021)	0.998 (0.014)	0.043 (0.026)	1.000 (0)	0.029 (0.021)
6. DB trimming	0.168 (0.221)	0.009 (0.012)	0.997 (0.017)	0.012 (0.01)	0.998 (0.021)	0.009 (0.009)
7. DB weighting	0.013 (0.027)	0.001 (0.004)	0.232 (0.232)	0.000 (0.003)	0.096 (0.152)	0.000 (0.002)
8. PB trimming	0.514 (0.096)	0.121 (0.024)	0.619 (0.089)	0.095 (0.022)	0.401 (0.086)	0.150 (0.022)
9. Outliergram	0.367 (0.201)	0.002 (0.005)	0.919 (0.134)	0.002 (0.005)	0.994 (0.023)	0.001 (0.004)
10. Adj. outliergram	0.251 (0.163)	0.002 (0.005)	0.972 (0.058)	0.004 (0.007)	1.000 (0.003)	0.008 (0.011)

model 1. Finally, let us recall that the HDR functional boxplot, the depth-based trimming method and the projection-based method had the advantage of having been given the *true* proportion of outliers in each sample.

Details about the implementation of the different methods and comparison of their computing times can be found in supplementary material available at *Biostatistics* online.

5. Assessing cluster quality in time course microarray analysis

In time course microarray experiments, the expression levels of thousands of genes are measured over time. The observed trajectories are then analyzed to characterize the gene expression profiles. This may allow the identification of genes whose expression depends on an experimental or phenotypic factor or those that are associated to certain biological processes. Sohn *and others* (2010) point out the necessity of using robust techniques to perform these analyses due to the common presence of outliers in this kind of experiments.

In this section, we illustrate the use of the outliergram as a diagnostic tool in clustering of time course microarray series by using the yeast data analyzed in the paper of Gillespie *and others* (2010), which provides a review of statistical methods for time course microarray data with Bioconductor (Gentleman *and others*, 2004). The data set is publicly available and it is accessible at https://github.com/csgillespie/bmc-microarray. It contains the expression level over time of a wild-type yeast strain and a mutant yeast strain sampled at five different time points. In this experiment, interest lay in finding differences in gene expression over time between the two strains. After selecting a group of highly differentially expressed genes (647), Gillespie *and others* (2010) conducted cluster analysis on the

"expression changes" of this subset to gain biological insight by determining groups of genes with a similar expression pattern. These expression changes are the standardized expression level differences between the mutant strain and the wild-type strain. The result of their cluster analysis is presented in Figure 3. They set the number of clusters to 8 and performed soft clustering with the R package `Mfuzz` (Futschik, 2013).

Table 2. *Mean and standard deviation* (*in parentheses*) *of the proportion of correctly and falsely identified outliers in the three simulation models over* 400 *simulation runs for* $n = 200$

	Model 1		Model 2		Model 3	
Method	p_c	p_f	p_c	p_f	p_c	p_f
$n = 200$, $c = 0$						
1. Fun. BP	–	0.000 (0)	–	0.000 (0)	–	0.000 (0.001)
2. Adj. fun. BP	–	0.004 (0.006)	–	0.003 (0.004)	–	0.004 (0.005)
3. Fun. HDR BP	–	0.000 (0)	–	0.000 (0)	–	0.000 (0)
4. Rob. Mah. dist.	–	0.013 (0.009)	–	0.013 (0.009)	–	0.012 (0.009)
5. ISE	–	0.029 (0.013)	–	0.030 (0.013)	–	0.030 (0.013)
6. DB trimming	–	0.012 (0.004)	–	0.012 (0.004)	–	0.012 (0.005)
7. DB weighting	–	0.016 (0.009)	–	0.016 (0.009)	–	0.016 (0.008)
8. PB trimming	–	0.000 (0)	–	0.000 (0)	–	0.000 (0)
9. Outliergram	–	0.049 (0.017)	–	0.050 (0.017)	–	0.046 (0.017)
10. Adj. outliergram	–	0.009 (0.007)	–	0.009 (0.007)	–	0.009 (0.007)
$n = 200$, $c = 0.05$						
1. Fun. BP	0.059 (0.084)	0.000 (0.001)	0.069 (0.086)	0.000 (0.001)	0.061 (0.083)	0.000 (0)
2. Adj. fun. BP	0.483 (0.197)	0.003 (0.005)	0.501 (0.19)	0.003 (0.005)	0.500 (0.209)	0.003 (0.005)
3. Fun. HDR BP	0.621 (0.148)	0.020 (0.008)	0.495 (0.142)	0.027 (0.007)	0.183 (0.123)	0.043 (0.006)
4. Rob. Mah. dist.	0.982 (0.045)	0.007 (0.007)	0.383 (0.178)	0.009 (0.007)	0.084 (0.099)	0.011 (0.009)
5. ISE	0.916 (0.235)	0.028 (0.012)	1.000 (0)	0.029 (0.015)	1.000 (0)	0.028 (0.014)
6. DB trimming	0.978 (0.056)	0.010 (0.005)	0.998 (0.02)	0.009 (0.005)	0.999 (0.011)	0.008 (0.005)
7. DB weighting	0.959 (0.091)	0.010 (0.007)	0.994 (0.048)	0.014 (0.009)	0.996 (0.054)	0.015 (0.009)
8. PB trimming	0.413 (0.132)	0.031 (0.007)	0.456 (0.134)	0.029 (0.007)	0.216 (0.118)	0.041 (0.006)
9. Outliergram	0.998 (0.015)	0.033 (0.013)	1.000 (0.005)	0.031 (0.013)	1.000 (0)	0.031 (0.013)
10. Adj. outliergram	0.972 (0.057)	0.006 (0.007)	0.983 (0.047)	0.006 (0.007)	1.000 (0.005)	0.008 (0.007)
$n = 200$, $c = 0.1$						
1. Fun. BP	0.04 (0.05)	0.000 (0)	0.054 (0.056)	0.000 (0)	0.058 (0.062)	0.000 (0)
2. Adj. fun. BP	0.442 (0.215)	0.004 (0.006)	0.520 (0.218)	0.004 (0.007)	0.565 (0.231)	0.005 (0.008)
3. Fun. HDR BP	0.592 (0.106)	0.045 (0.012)	0.598 (0.101)	0.045 (0.011)	0.324 (0.112)	0.075 (0.012)
4. Rob. Mah. dist.	0.974 (0.043)	0.003 (0.005)	0.361 (0.137)	0.005 (0.006)	0.111 (0.091)	0.008 (0.008)
5. ISE	0.902 (0.242)	0.025 (0.012)	1.000 (0.003)	0.031 (0.017)	1.000 (0)	0.026 (0.013)
6. DB trimming	0.874 (0.273)	0.011 (0.007)	0.998 (0.014)	0.013 (0.007)	1.000 (0.004)	0.011 (0.006)
7. DB weighting	0.074 (0.101)	0.001 (0.003)	0.984 (0.066)	0.008 (0.007)	0.994 (0.07)	0.008 (0.007)
8. PB trimming	0.469 (0.096)	0.059 (0.011)	0.542 (0.091)	0.051 (0.01)	0.310 (0.086)	0.077 (0.01)
9. Outliergram	0.986 (0.028)	0.022 (0.012)	0.995 (0.017)	0.017 (0.01)	1.000 (0.003)	0.018 (0.011)
10. Adj. outliergram	0.904 (0.084)	0.004 (0.006)	0.967 (0.056)	0.005 (0.007)	0.999 (0.007)	0.006 (0.007)
$n = 200$, $c = 0.15$						
1. Fun. BP	0.023 (0.036)	0.000 (0)	0.046 (0.045)	0.000 (0)	0.039 (0.043)	0.000 (0)
2. Adj. fun. BP	0.369 (0.214)	0.003 (0.005)	0.656 (0.231)	0.007 (0.008)	0.675 (0.227)	0.007 (0.008)

Continued

Table 2. *Continued*

Method	Model 1		Model 2		Model 3	
	p_c	p_f	p_c	p_f	p_c	p_f
3. Fun. HDR BP	0.598 (0.081)	0.071 (0.014)	0.661 (0.073)	0.060 (0.013)	0.434 (0.101)	0.100 (0.018)
4. Rob. Mah. dist.	0.948 (0.072)	0.001 (0.003)	0.317 (0.117)	0.003 (0.004)	0.118 (0.09)	0.006 (0.007)
5. ISE	0.779 (0.32)	0.023 (0.013)	1.000 (0.003)	0.038 (0.02)	1.000 (0)	0.025 (0.013)
6. DB trimming	0.479 (0.393)	0.010 (0.008)	0.999 (0.007)	0.016 (0.008)	1.000 (0)	0.013 (0.007)
7. DB weighting	0.019 (0.027)	0.001 (0.002)	0.838 (0.198)	0.002 (0.004)	0.726 (0.343)	0.002 (0.004)
8. PB trimming	0.499 (0.078)	0.088 (0.014)	0.590 (0.075)	0.072 (0.013)	0.378 (0.075)	0.110 (0.013)
9. Outliergram	0.889 (0.089)	0.009 (0.007)	0.988 (0.025)	0.008 (0.007)	1.000 (0.004)	0.007 (0.006)
10. Adj. outliergram	0.637 (0.15)	0.002 (0.004)	0.964 (0.053)	0.005 (0.006)	0.999 (0.004)	0.005 (0.007)
$n = 200$, $c = 0.2$						
1. Fun. BP	0.011 (0.023)	0.000 (0)	0.028 (0.033)	0.000 (0)	0.026 (0.031)	0.000 (0)
2. Adj. fun. BP	0.317 (0.218)	0.003 (0.005)	0.679 (0.161)	0.006 (0.007)	0.710 (0.143)	0.007 (0.007)
3. Fun. HDR BP	0.588 (0.074)	0.103 (0.019)	0.718 (0.062)	0.070 (0.015)	0.521 (0.093)	0.120 (0.023)
4. Rob. Mah. dist.	0.878 (0.116)	0.000 (0.001)	0.286 (0.108)	0.001 (0.003)	0.124 (0.092)	0.004 (0.006)
5. ISE	0.449 (0.367)	0.023 (0.014)	1.000 (0.004)	0.041 (0.022)	1.000 (0)	0.024 (0.014)
6. DB trimming	0.186 (0.231)	0.008 (0.008)	0.998 (0.009)	0.017 (0.007)	0.999 (0.01)	0.013 (0.006)
7. DB weighting	0.015 (0.021)	0.001 (0.003)	0.199 (0.159)	0.000 (0.002)	0.061 (0.062)	0.000 (0.001)
8. PB trimming	0.521 (0.075)	0.120 (0.019)	0.638 (0.06)	0.091 (0.015)	0.421 (0.064)	0.145 (0.016)
9. Outliergram	0.341 (0.164)	0.002 (0.003)	0.921 (0.084)	0.001 (0.003)	0.994 (0.017)	0.001 (0.003)
10. Adj. outliergram	0.235 (0.134)	0.001 (0.003)	0.964 (0.056)	0.004 (0.006)	0.999 (0.005)	0.008 (0.008)

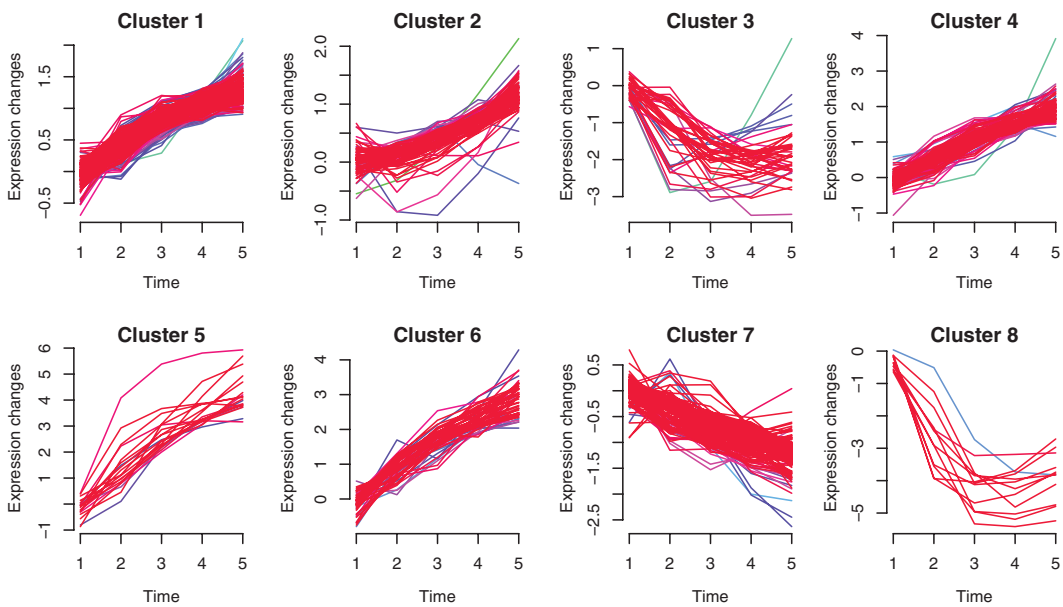

Fig. 3. Eight clusters of expression changes in the yeast microarray data set obtained by Gillespie *and others* (2010).

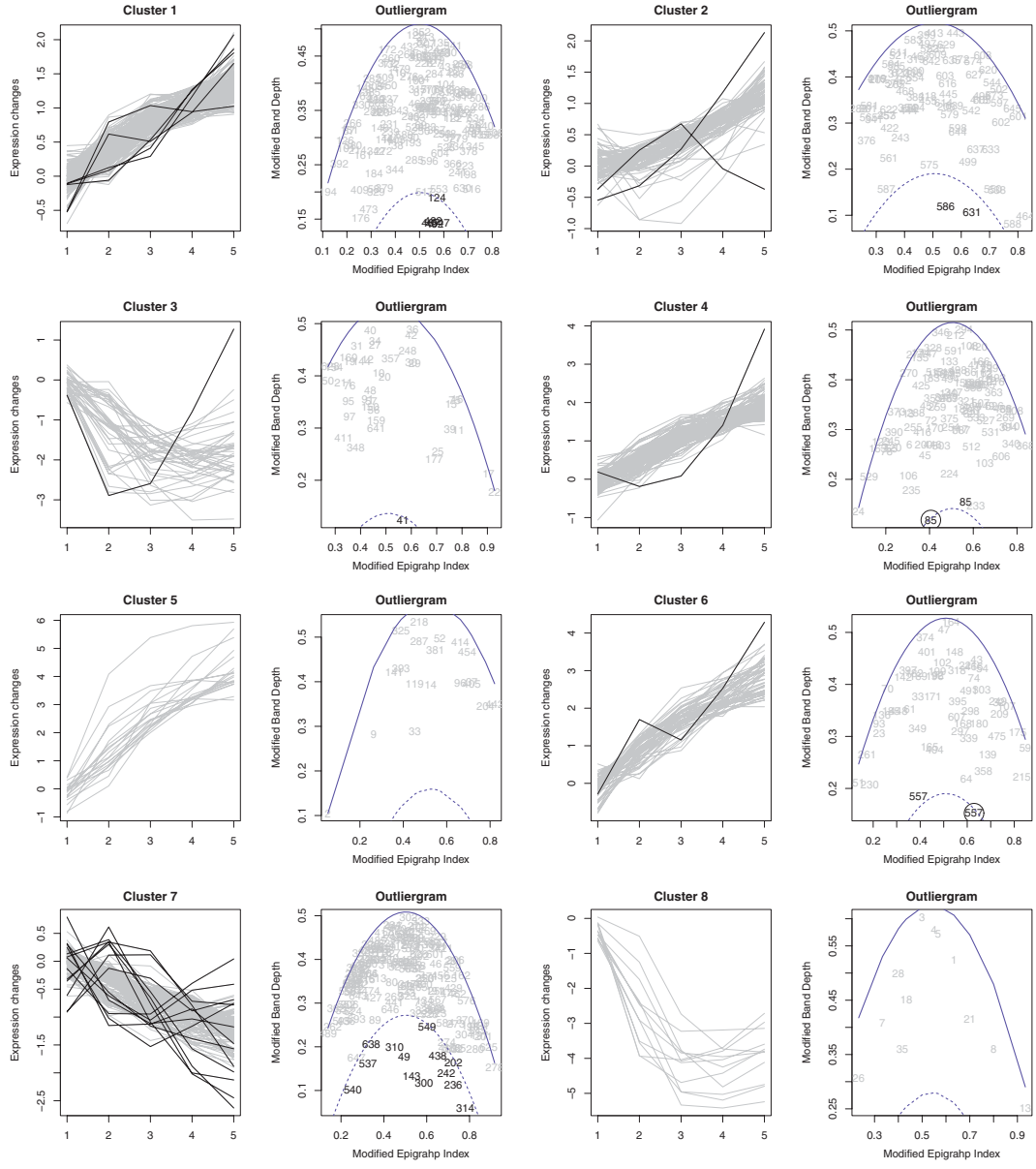

Fig. 4. Eight clusters of expression changes in the yeast microarray data set obtained by Gillespie *and others* (2010) with corresponding outliergrams. Shape outliers are printed in black.

We have now applied the outliergram, in its adjusted version, to each one of these clusters to assess the overall cluster quality and to look for possibly misclassified genes. Since some of the cluster sizes are relatively small, we have set to 400 the number of sampled data sets and to 20 the number of search points for the factor value. The results are presented in Figure 4. Two of the clusters (5 and 8) do not present any outlier, and three of them (3, 4, and 6) only present one. If we now focus on cluster 7, the one with the highest number of outliers, we appreciate that the general pattern of the cluster is a linear decreasing

trend, whereas all the detected outliers present at least one trend change. Some of these outliers, the ones that start with a decreasing trend and then increase around time point 3, seem to be similar to the genes in cluster 3 or 8 (which could be considered as a unique cluster). The outliers that, on the other hand, start with an increasing trend to then decrease after the first time point, do not seem to have similarities with any of the existing clusters. With respect to the outliers detected in the rest of the clusters, note that, except for one gene in cluster 2 and three in cluster 1, all exhibit trend changes difficult to accommodate in cluster 3 or 8 (the only clusters with non-monotonic patterns). Whether they constitute a new cluster or should be considered atypical genes is something an expert should determine. In any case, the outliergram visualization of the clusters provides a helpful tool for cluster inspection.

6. Discussion

This article proposes the outliergram as a tool for representing functional observations in the plane as a function of two depth indices, the MBD and the MEI. It allows one to visually assess sample variability in terms of shape and to detect potential shape outliers. Indeed, the MBD and the MEI are conceptually similar: both are defined through the proportion of time that the curve is inside a band or an epigraph. This allows one to establish a relationship between them. Moreover, since the MBD of a curve is highly dependent on its location (in the sense of vertical position) in the sample, and the MEI provides a measure of this location, the conditional observation of the MBD given the modified band index provides an accurate shape descriptor which allows one to identify shape outliers.

The more similar and smooth the curves in the sample, the closer to the parabola (2.1) are the points in the outliergram. On the other hand, the more noisy the curves and the larger number of crossing points between them, the more dispersed are the points under (2.1) in the outliergram. In both cases, the points with the largest distances to the parabola represent the most outlying curves, in terms of shape, of the sample. However, it will be easier to detect them if the rest of the points in outliergram are concentrated near (2.1). Thus, we suggest applying a smoothing step to noisy data sets before using the outliergram in order to enhance shape differences and similarities among curves. Also, it is important to note that our method considers different from the rest a curve that may have the same shape as the majority of the curves of the sample but is misaligned with respect to them. Then, depending on the particular application, it could be necessary to align the curves before using the outliergram.

In addition to the visualization tool, we propose a general boxplot-based rule on the distances to the parabola (2.1) to classify observations into outlying and non-outlying in terms of shape. While this is a simple rule with no extra computational cost once the distances to the parabola are obtained, it may tend to detect more outliers than there actually are in the sample. A more sophisticated data set-dependent rule is also proposed, where the boundary for the outlying region is chosen to control the false detection rate. According to the simulation results this rule exhibits better specificity and equivalent sensitivity rates than the simple one. However, its computational cost is higher, and sometimes a visual inspection of the outliergram points will be enough to provide good understanding of the nature of the sample and the classifying rule.

For curves that lie above or below the majority of the curves it will be difficult to assess whether they have an atypical shape since they are not surrounded by other curves to which they could be compared. We propose to shift these curves toward the center of the sample to see if, in this new position, they stand out as having a different shape. Although we give here a specific rule on which curves should be shifted and on how much they should be shifted, extensions to different rules that may cope with the particular nature of different data sets are possible.

Finally, we suggest to combine the outliergram with the functional boxplot (Sun and Genton, 2011) to account for both magnitude and shape outliers, as we do in the R functions that we provide as supplementary material available at *Biostatistics* online.

SUPPLEMENTARY MATERIAL

Supplementary material is available at http://biostatistics.oxfordjournals.org.

ACKNOWLEDGMENTS

We are grateful to Marcela Svarc, who provided us with the R code for the implementation of the method described in Fraiman and Svarc (2013). We also thank two anonymous referees and an Associate Editor for their constructive comments. *Conflict of Interest*: None declared.

FUNDING

The authors acknowledge financial support from grant ECO2011-25706, Spain. The first author also acknowledges financial support from grant MTM2010-17323, Spain.

REFERENCES

CUEVAS, A., FEBRERO, M. AND FRAIMAN, R. (2006). On the use of bootstrap for estimating functions with functional data. *Computational Statistics and Data Analysis* **51**, 1063–1074.

FEBRERO, M., GALEANO, P. AND GONZÁLEZ-MANTEIGA, W. (2008). Outlier detection in functional data by depth measures, with application to identify abnormal NOx levels. *Environmetrics* **19**, 331–347.

FRAIMAN, R. AND SVARC, M. (2013). Resistant estimates for high dimensional and functional data based on random projections. *Computational Statistics & Data Analysis* **58**, 326–338.

FUTSCHIK, M. (2013). *Mfuzz: Soft Clustering of Time Series Gene Expression Data*. R package version 2.20.0.

GENTLEMAN, R. C., CAREY, V. J., BATES, D. M., BOLSTAD, B., DETTLING, M., DUDOIT, S., ELLIS, B., GAUTIER, L., GE, Y., GENTRY, J. *and others*. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology* **5**, R80.

GERVINI, D. (2009). Detecting and handling outlying trajectories in irregularly sampled functional datasets. *The Annals of Applied Statistics* **3**, 1758–1775.

GERVINI, D. (2012). Outlier detection and trimmed estimation for general functional data. *Statistica Sinica* **22**, 1639–1660.

GILLESPIE, C. S., LEI, G., BOYS, R. J., GREENALL, A. AND WILKINSON, D. K. (2010). Analysing time course microarray data using bioconductor: a case study using yeast2 affymetrix arrays. *BMC Research Notes* **3**, 81.

HUBERT, M. AND VANDERVIEREN, E. (2008). An adjusted boxplot for skewed distributions. *Computational Statistics and Data Analysis* **52**, 5186–5201.

HYNDMAN, R. J. (1996). Computing and graphing highest density regions. *The American Statistician* **50**, 120–126.

HYNDMAN, R. J. AND SHANG, H.L. (2010). Rainbow plots, bagplots, and boxplots for functional data. *Journal of Computational and Graphical Statistics* **19**, 29–49.

HYNDMAN, R. J. AND ULLAH, M. S. (2007). Robust forecasting of mortality and fertility rates: a functional data approach. *Computational Statistics & Data Analysis* **52**, 4924–4956.

LÓPEZ-PINTADO, S. AND ROMO, J. (2009). On the concept of depth for functional data. *Journal of the American Statistical Association* **104**, 718–734.

LÓPEZ-PINTADO, S. AND ROMO, J. (2011). A half-region depth for functional data. *Computational Statistics & Data Analysis* **55**, 1679–1695.

Maronna, R. A. and Zamar, R. H. (2002). Robust estimates of location and dispersion of high-dimensional datasets. *Technometrics* **44**, 307–317.

Martín-Barragán, B., Lillo, R. and Romo, J. (2012). Functional boxplots based on half-regions. Universidad Carlos III de Madrid, Working paper series.

Ramsay, J. O. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer Series in Statistics. New York: Springer.

Rong, T. and Müller, H. G. (2009). Time-synchronized clustering of gene expression trajectories. *Biostatistics* **10**, 32–45.

Sohn, I., Owzar, K., George, S. L., Kim, S. and Jung, S.-H. (2010). Robust test method for time-course microarray experiments. *BMC Bioinformatics* **11**, 391.

Sun, Y. and Genton, M. G. (2011). Functional boxplots. *Journal of Computational and Graphical Statistics* **20**, 316–334.

Sun, Y. and Genton, M. G. (2012). Adjusted functional boxplots for spatio-temporal data visualization and outlier detection. *Environmetrics* **23**, 54–64.

[*Received September 30, 2013; revised February 3, 2014; accepted for publication February 3, 2014*]