Neural Processing Letters (2020) 52:135–152
https://doi.org/10.1007/s11063-018-9926-1

Time Series, Spectral Densities and Robust Functional Clustering

D. Rivera-García[1] · L. A. García-Escudero[2] · A. Mayo-Iscar[2] · J. Ortega[1]

Published online: 1 October 2018

Abstract
In this work, a robust clustering algorithm for stationary time series is proposed. The algorithm is based on the use of estimated spectral densities, which are considered as functional data, as the basic characteristic of stationary time series for clustering purposes. A robust algorithm for functional data is then applied to the set of spectral densities. Trimming techniques and restrictions on the scatter within groups reduce the effect of noise in the data and help to prevent the identification of spurious clusters. The procedure is tested in a simulation study and is also applied to a real data set.

Keywords Time series clustering · Robust clustering · Robust functional data clustering · Spectral analysis

1 Introduction

Interest in the problem of finding clusters of time series has grown in recent years. A variety of methods have been proposed, and applications to several different fields have been studied. However, many of these methods are sensitive to the presence of outliers in the sample. A robust clustering algorithm for stationary time series that considers the possible presence of contamination and is based on the analysis of spectral densities as functional data is proposed in this work. A mixture modeling approach is adopted for the functional clustering approach developed. To achieve robustness, the possibility of trimming certain fraction of the (hopefully) most outlying time series is considered. The introduction of appropriate constraints on the clusters' scatter parameters is another key ingredient of the proposed method. These constraints aim to avoid a well-known drawback in mixture modeling which has to do with local maxima of the likelihood resulting in the detection of non-interesting or "spurious" clusters. The method considered in this work has robust characteristics that reduce the effect of noise in the sample and hinder the detection of spurious clusters.

✉ J. Ortega
jortega@cimat.mx

1 Centro de Investigación en Matemáticas, Guanajuato, Mexico

2 Universidad de Valladolid, Valladolid, Spain

Liao [31], Caiado et al. [9], Aghabozorgi et al. [1] provide extensive revisions of the time series clustering area (see also [20]). A time series clustering package in R, with a wide choice of methods, was developed by [33]. According to [31], there are three main approaches to time series clustering: procedures based on the comparison of the raw data, procedures that depend on the comparison of models fitted to the data and, finally, methods based on characteristics derived from the time series. Our proposal falls within the third group, and the feature used to assess the similarity between time series is the spectral density.

Other authors have considered the spectral viewpoint for time series clustering. For instance, [7,8] considered the use of periodogram and normalized periodogram values. [32] propose a fuzzy clustering algorithm based on the estimated "cepstrum", which is the spectrum of the logarithm of the spectral density function. [3,17] use the total variation distance as a dissimilarity measure between normalized estimates of the spectral densities for time series clustering. A brief review of these two algorithms will be given in Sect. 2. Other distances such as the Kullback–Leibler divergence [34] have been used. A discussion on its use in cluster analysis of time series can be found in [39]

Our approach relies on the use of functional data clustering as a tool for grouping stationary time series, and the functional object considered is the spectral density. The use of Functional Data Analysis in Statistics can be reviewed in the following two monographs: [18,35]. A recently developed algorithm for robust functional data clustering presented in [36] is used on the normalized estimated spectral densities to cluster the corresponding time series. This procedure will be described in Sect 3. Some preliminary results on this approach, for robust time series clustering, were presented in [37]. We extend this work here with special emphasis on how to choose, in practical applications, the several tuning parameters involved. Various clustering procedures for functional data have been proposed in the literature as, for example, [6,28,29] but they are not aimed at dealing with contamination by outlying curves, which is a commonplace in (unsupervised) Data Analysis. Trimming techniques for robust clustering of functional data have been also used in [13,22].

Several works have focused on developing robust algorithms for time series clustering. [42] obtain Independent Components for multivariate time series and develop a clustering algorithm, known as ICLUS to group them with good robustness performance. [15] use autoregressive models and a fuzzy approach to propose a robust clustering model for time series. A partition around medoids scheme is adopted and the robustness of the method comes from the use of a robust metric between time series. [16] present robust fuzzy clustering algorithms for heteroskedastic time series, based on GARCH parametric models, using again a partition around medoids approach. Three different robust models are proposed, following different robustification approaches: metric, noise, and trimming. [5] propose a clustering framework for functional data, known as Functional Subspace Clustering, which is based on subspace clustering [41] and can be applied to time series with warped alignments.

Many of the previously commented methods for robust clustering are based on the partition around medoids (PAM) procedure. Although it is true that PAM entails a noticeable robustification in most cases, it is also known that a single outlying observation placed in a harmful position can still break PAM-based clustering results down (see [21]). In fact, when using PAM, protection against "gross" outliers is not fully guaranteed and "gross" outliers may often be present in Data Analysis applications. Trimming is also useful because it naturally helps to identify potential outliers, as long as trimmed observations are considered as the potentially "most outlying" ones. The proper identification of outliers is an interesting task in itself if the researcher is able to explain why these observations differ from the bulk of the data. The consideration of constraints is also an important feature of the proposed methodology given that they allow us to obtain a wider range of different solutions, accord-

ing to the degree of the constraint imposed, so that one of them can better fit the researcher's clustering aims. Constraints also turn the corresponding likelihood maximizations into a mathematically well-defined problems and they prevent the detection of spurious clusters [24,25].

The paper is organized as follows: Sect. 2 considers time series clustering and describes the idea behind our proposal. Sect. 3 gives a brief description of the robust clustering procedure for functional data that supports the time series clustering algorithm. Sect. 4 provides information about the choice of the several tuning parameters involved. Sect. 5 presents a simulation study designed to compare the performance of the algorithm with existing alternatives and Sect. 6 gives an application to a real data set. The paper ends with some discussion of the results and some ideas about future work.

2 Spectral Densities and Time Series Clustering

In this section, we describe the main ideas behind the proposed clustering method. The general problem is that of clustering a collection of stationary time series. Assume that we have data from n time series: $\mathbf{Y}_1 = \{Y_{1t}\}_{t=1}^{\tau_1}, \ldots, \mathbf{Y}_n = \{Y_{nt}\}_{t=1}^{\tau_n}$, i.e. different series may have different lengths. The spectral densities $\phi_1, \ldots, \phi_n$ for these series are estimated using one of the many methods available, and these functions are used to compare the different time series to obtain homogeneous clusters. As we mentioned, previous methods for time series clustering based on the spectral domain rely on similarity measures for differentiation. In contrast, in this work, the spectral densities are considered as functional data and the robust clustering algorithm presented in [36] is used. Time series belonging to the same cluster will have spectral densities similar in shape and therefore similar oscillatory characteristics. The algorithm is sensitive to the presence of contamination in the sample of spectral densities, which correspond to time series having unusual oscillatory behavior.

In Sect. 5 the method proposed in this paper will be compared with three other methods based on the estimated spectral densities as the characteristic feature of each time series. Two of them were proposed in [3,17] and will be described briefly in what follows. We refer to them as "TVDClust" and "HSMClust", respectively. The third one is based on the use of the well-known Kullback–Leibler divergence.

"TVDClust" and "HSMClust" use the total variation distance (TVD) to measure dissimilarities between spectral densities. This distance is frequently used for probability measures and if the measures P and Q have densities f and g, the TVD between them is given by

$$d_{TV}(f, g) = 1 - \int_{-\infty}^{\infty} \min(f, g)\, dx = \frac{1}{2} \int_{-\infty}^{\infty} |f - g|\, dx. \tag{1}$$

The first equation helps to interpret the meaning of this distance. If $d_{TV}(f, g) = 1 - a$ then the common area below the curves representing f and g is a. In consequence, the bigger the common area, the closer the two densities and the smaller the TVD.

However, spectral densities are not probability densities, since their integral is not necessarily equal to one. Hence, they have to be normalized so that the total area below the curve equals one. These normalized spectral densities will be denoted as $x_1, \ldots, x_n$ in the sequel. This is tantamount to rescaling the time series so that its variance is one. Thus, the oscillatory behavior of the series and not the size of the oscillations, is the basis of these algorithms.

The "TDVClust" method, proposed in [3], uses a classical hierarchical agglomerative algorithm, fed with a dissimilarity matrix that contains the TVD between all pairs of normalized spectral densities. Two linkage functions, average or complete, can be used to obtain a

dendrogram which can be cut to obtain the desired number of groups. The number of clusters is selected using an external criterion, such as the Silhouette or Dunn's index. More details can be found in [3].

The "HSMClust" (Hierarchical Spectral Merger) method is a modification of the previous one. The difference is that every time two clusters are joined, the information in them is used to obtain a new characteristic spectrum for the new group. This can be done by taking the average of all the spectra in the new cluster which is the "average" option or by concatenating the corresponding time series and estimating a new spectral density, which is the "single" option. Under the assumption that the time series in the same cluster come from the same model, either of these procedures will give a better estimation of the common spectral density. Hence, every time two clusters are joined the dissimilarity matrix reduces its size while in the previous algorithm the dissimilarity matrix is the same during the whole procedure, and the distances between clusters are calculated using linear combinations of the entries in this matrix. The linear combination used varies according to the linkage function selected. Euán et al. [17] present two methods for selecting the number of clusters, one method is based on the distance between the closest clusters at each step, and the other is based on bootstrap procedures. More details can be found in the reference. This method, has been implemented in R and is available at https://es.kaust.edu.sa/Pages/CarolinaEuan.aspx.

Finally, the use of the Kullback–Leibler divergence in this context has been reviewed in [3], where its performance is compared with the two methods we have just described. This method will be denoted as 'KLClust" in what follows.

3 Robust Clustering for Functional Data

In this section, we review the methodology introduced in [36] for robust functional clustering. Let us assume that X is a random process taking values in $L^2([0, T])$, the Hilbert space of the square integrable functions in the closed interval $[0, T]$. The inner product in that space is defined as $\langle f, g\rangle = \int f(t)g(t)\,dt$ and $\|f\|^2 = \langle f, f\rangle$.

The well-known Karhunen-Loève expansion allows representing the random process X through the expression:

$$X(t) = \mu(t) + \sum_{j=1}^{\infty} C_j(X)\psi_j(t), \tag{2}$$

where $\mu(t) = E\{X(t)\}$ and ψ_j are the orthonormal eigenfunctions of the eigen-equation:

$$\langle \Gamma(\cdot, t), \psi_j\rangle = \lambda_j\psi_j(t), \tag{3}$$

for the covariance operator $\Gamma(s, t) = \text{cov}(X(s), X(t))$. The random variables $C_j(X)$ are known as the principal components and are obtained as $C_j(X) = \langle X - \mu, \psi_j\rangle$. These principal component variables are uncorrelated with $E(C_j(X)) = 0$ and $\text{Var}(C_j(X)) = \lambda_j$. The λ_j are chosen in decreasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_j \geq \cdots$ and satisfy $\sum_{j=1}^{\infty} \lambda_j < \infty$.

A finite dimensional approximation of the random process X can be obtained by taking only the first p terms in the expansion (2). Moreover, if $f_{C_j}(\cdot)$ denotes the probability density function (p.d.f.) of the random variable $C_j(X)$ and $c_j(x) = \langle x, \psi_j\rangle$, it is shown in [14] that $\log P(||X - x|| \leq h)$ can be approximated by $\sum_{j=1}^{p} \log f_{C_j}(c_j(x))$ for small values of h. This approximation holds for any $x \in L_2([0, T])$ and, thus, the joint distribution of $(C_1(X), \ldots, C_p(X))$ can be used to define a kind of "small-ball pseudo-density" in the context of Functional Data Analysis. More details about this "pseudo-density" approach can

be found in [14]. Furthermore, if we assume that X is a Gaussian random process then the $C_j(X)$ random variables are uncorrelated and normally distributed $C_j(X) \sim N(0, \lambda_j)$ and the joint distribution of $(C_1(X), \ldots, C_p(X))$ simplifies to

$$\prod_{j=1}^{p} \frac{1}{\sqrt{2\pi\lambda_j}} \exp\left(\frac{-c_j(x)^2}{2\lambda_j}\right). \tag{4}$$

From previous ideas, a "model-based" clustering approach for functional data can be derived assuming K different expansions, one for each cluster component, $g = 1, \ldots, K$. This is the approach considered in [6,28]. They both propose considering the same order p in all the cluster expansions, but [6] considers that the first q_g eigenvalues/variances are the most important ones while the remaining $p - q_g$ are smaller and they are assumed to be equal in each cluster. This simplifies this largely parameterized problem and, then, q_g may be seen as a kind of "intrinsic" cluster dimension. We denote the first q_g main variances/eigenvalues in the g-th cluster as a_{1g},..., $a_{q_g g}$ and the common (residual) eigenvalue/variance as b_g.

Although the methodology introduced in [6,28] is notably useful for clustering purposes, these methods are not designed to deal with the presence of outlying functions in the data set. However, even the presence of a small fraction of outlying functions can be extremely harmful. For instance, main clusters can be artificially split or joined together and small clusters consisting of a few outlying observations can be wrongly detected. With the idea of robustifying the procedures in [6,28], [36] introduces a robust functional clustering approach where a fixed proportion α of the most outlying functions are allowed to be trimmed. The proposed approach also imposes appropriate constraints on the eigenvalues/variances to avoid the detection of non-interesting or "spurious" clusters (see [36]).

Given a set of functions $\{x_1, \ldots, x_n\}$ in $L^2([0, T])$, we use a 0–1 indicator function such that $\eta(x_i) = 0$ if the x_i function is trimmed and $\eta(x_i) = 1$ if it is not. In this work, recall that $\{x_1, \ldots, x_n\}$ are going to be the normalized spectral densities for our n time series $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$. As we only trim a proportion α of functions, we consider the constraint $\sum_{i=1}^{n} \eta(x_i) = [n(1-\alpha)]$. The associated trimmed mixture-loglikelihood is then defined as

$$\sum_{i=1}^{n} \eta(x_i) \log(D(x_i; \theta)), \tag{5}$$

with

$$D(x_i; \theta) = \sum_{g=1}^{K} D_g(x_i; \theta), \tag{6}$$

and

$$D_g(x_i; \theta) = \pi_g \prod_{j=1}^{q_g} \frac{1}{\sqrt{2\pi a_{jg}}} \exp\left(\frac{-c_{ijg}^2}{2a_{jg}}\right) \prod_{j=q_g+1}^{p} \frac{1}{\sqrt{2\pi b_g}} \exp\left(\frac{-c_{ijg}^2}{2b_g}\right). \tag{7}$$

All the parameters, which are needed to be estimated for that representation, are encompassed in θ and $c_{ijg} = c_{jg}(x_i)$ is the j-th principal component score of function x_i in group g. As often in mixture modeling, π_g weights, summing up to 1, are considered as mixing proportions which also have to be estimated.

Given two fixed constants $d_1 \geq 1$ and $d_2 \geq 1$, we propose the constrained maximization of the trimmed likelihood (5) under the following two restrictions:

$$\frac{\max_{g=1,\ldots,K; j=1,\ldots,q_j} a_{jg}}{\min_{g=1,\ldots,K; j=1,\ldots,q_j} a_{jg}} \leq d_1 \tag{8}$$

and

$$\frac{\max_{g=1,\ldots,K} b_g}{\min_{g=1,\ldots,K} b_g} \leq d_2. \tag{9}$$

The smaller d_1 and d_2, the more similar the variance/eigenvalues should be across and within clusters but a different treatment can be given for the main and the residual variances.

The highest value obtained for (5) after the constrained optimization is denoted as $L_{d_1,d_2}(\alpha, K)$ and will be used in Sect. 4.

Cluster assignments for the original time series $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$ can be obtained by assigning $\mathbf{Y}_i$ to cluster G if

$$\pi_G D_G\left(x_i; \hat{\theta}\right) = \max_{g=1,\ldots,K} \pi_g D_g\left(x_i; \hat{\theta}\right), \tag{10}$$

where $\widehat{\theta}$ is the optimal θ resulting from the constrained trimmed likelihood maximization.

The proposed constrained maximization is obviously a computationally expensive task. However, a computationally feasible algorithm was given in [36]. The algorithm follows from the adaptation of the classical Expectation-Maximization (EM) algorithm but incorporating a "trimming step". This is similar to the "concentration" steps already applied in [38]. A brief description of the proposed algorithm is given in Algorithm 1 but a more detailed description of all the steps can be seen in [36].

Algorithm 1 Basic algorithm

Data: Time series $\{\mathbf{Y}_i\}_{i=1,\ldots,n}$, trimming level α, dimensions $\{q_g\}_{g=1,\ldots,K}$ and constraining constants d_1 and d_2
Result: Time series that are trimmed and clustering assignments for the non-trimmed ones
Compute the normalized spectral densities $\{x_i\}_{i=1,\ldots,n}$ of $\{\mathbf{Y}_i\}_{i=1,\ldots,n}$
for $b = 1, \ldots, B$ **do**
  Initialize $\theta^{(0)}$
  for $c = 1, \ldots, C$ **do**
    $\tau_{ig} \leftarrow D_g(x_i; \theta^{(c-1)})/D(x_i; \theta^{(c-1)})$
    Sort $\{D(x_i; \theta^{(c-1)})\}_{i=1,\ldots,n}$ into $\{D(x_{(i)}; \theta^{(c-1)})\}_{i=1,\ldots,n}$
    $\mathcal{I}^{(c)} \leftarrow \{i : D(x_i; \theta^{(c-1)}) \leq D(x_{([n(1-\alpha)])}; \theta^{(c-1)})\}$
    $\tau_{ig} \leftarrow 0$ for $i \notin \mathcal{I}^{(c)}$
    $\pi_g^{(c)} \leftarrow \sum_{i=1}^{n} \tau_{ig}/(n(1-\alpha))$
    Perform FPCA with $\{\tau_{ig}\}_{i=1,\ldots,n}$ weights for $g = 1, \ldots, K$
    Apply constraints on $\{a_{jg}\}_{g=1,\ldots,K}^{j=1,\ldots,q_g}$ and $\{b_g\}_{g=1}^{K}$
    Update $\theta^{(c)}$ accordingly
  end for
  Compute (5) with $\eta(x_i) = 1$ if $i \in \mathcal{I}^{(C)}$ and $\eta(x_i) = 0$ if not
end for
Return the optimal $\theta^{(C)}$ and $\mathcal{I}^{(C)}$ yielding the smallest value of (5)
Use the rule as in (10) to obtain the clustering assignments

In the M-step, we have to perform weighed Functional Principal Component Analyses [35]. It may happen that the resulting a_{jg} and b_g parameters do not satisfy the required constraints, for the fixed d_1 and d_2 constants. In that case, these parameters have to be truncated in an optimal way so that the required constraints are satisfied. This is done as in [19] by maximizing two real valued functions (see details in [36]).

As will be seen in Sect. 4, we propose determining the q_g "intrinsic dimensions" by considering a Bayesian Information Criterion (BIC) approach.

The computing cost, although high, is not much higher than the two closely related approaches in [6,28]. Notice that only an extra sorting of the $D(x_i; \theta^{(c-1)})$ values is needed in each iteration, and that the constraints on the scatter parameters only require the maximization of two real valued functions (that can be quickly done by adapting the procedure in [19]). The higher B, the smaller the probability that the algorithm ends up trapped in a local maxima of the likelihood. Of course, higher B and C values yield higher computing times. Computing time can be saved by considering an stopping criterion that avoids fully iterating until $c = C$, in Algorithm1, when convergence has been already reached for $c < C$. In fact, our experience shows that very high C values are rarely needed. Consequently, a sensible strategy is to consider a high B value but a small C value. Then, we only fully iterate the "most promising" initializations, i.e. those with the highest values of the target function after a small number C of iterations.

Given that each initialization $b = 1, \ldots, B$ returns a clustering partition, another possibility to be explored is trying to combine all the information from those B partitions by applying "ensemble clustering" techniques (see, e.g., [27] and references therein). This could serve to obtain interesting initializing partitions which would allow exploring the space of solutions in a more efficient way. Take also into account that the algorithm can be easily parallelized. It is also important to note that the procedures for choosing parameters, that will be described in Sect. 4, are the most computationally demanding part of the procedure because the problem must be solved for several combinations of the tuning parameters. In that case, the optimal values from contiguous parameter configurations have to be taken into account in the initializing steps to alleviate the computational efforts.

4 Choice of Parameters

The proposed methodology is very flexible but needs the specification of several tuning parameters. In this section, we will comment on some tools that can help the user in this task.

For instance, the q_g "intrinsic dimensions" can be determined by considering the Bayesian Information Criterion (BIC). In the BIC approach, models with complexity higher than needed are penalized when comparing their associated log-likelihoods. To be more precise, we propose performing the constrained maximization in (5) for several combinations of the q_g "intrinsic dimensions" and consider as final solution the one with the largest value of the target function in (5) after subtracting $\kappa \log(n)$, where κ is the total number of free parameters that need to be estimated in the target function. As already shown in [36], we can see that the number of free parameters is

$$\kappa = (Kp + K - 1) + \sum_{g=1}^{K} q_g(p - (q_g + 1)/2) + 2K + \sum_{g=1}^{K} q_g.$$

We have considered this BIC approach in the simulation study and in all the illustrating examples throughout this work.

The parameters d_1 and d_2 serve to control the maximum differences in scatter allowed within and across clusters. Large values of these parameters may produce excessive cluster heterogeneity. In fact, having a small value for d_2 (i.e., close to 1) is recommended because this is the parameter that avoids degeneracy of the target function. Our experience is that the number of significantly different solutions when moving d_1 and d_2 is not very large, once large d_1 values are excluded and it is a good idea to explore them to choose the one that better fits the user's clustering purposes. In a recent work [10], a modified BIC proposal is

introduced that takes into account the higher "model complexity" that larger d_1 and d_2 entail. This possibility needs to be explored and it will be considered in future works.

Choosing the correct trimming level α is a key decision. It is particularly important to choose α higher than the true contamination level to avoid the harmful effect of outlying functions in the final clustering results. However, although less problematic, a trimming level higher than needed is also troublesome because good observations can be wrongly trimmed. This conflict may be seen as a kind of tradeoff between robustness and efficiency. The choice of the number of clusters K is also a complex problem, extensively addressed in Cluster Analysis. Note also that the choices for K and α are also dependent on d_1 and d_2. For instance, a set of scattered spectra can be considered as a main cluster (so increasing K and decreasing α is needed) if d_1 and d_2 are chosen large enough to allow the detection of more heterogeneous clusters. Once d_1 and d_2 are fixed, an adaptation of the "ctlcurves" introduced in [23] can be used to obtain sensible choices of K and α, simultaneously. The "ctlcurves" are based on plotting the functionals $(\alpha, K) \mapsto L_{d_1,d_2}(\alpha, K)$, where $L_{d_1,d_2}(\alpha, K)$ was defined in Sect. 3. As an example, let us consider the simulated data set shown in Fig. 1. The modes of two main clusters spectra are located at 0.2 and 0.3 Hz, respectively, with the same energy equal to 1.2. On the other hand, the modes for a 10% proportion of outlying spectra are randomly obtained from a uniform distribution in the $(0.1, 0.4)$ interval and their energies are also randomly obtained from a uniform distribution in the $(1.05, 1.1)$ interval. The right panel of Fig. 1 shows the "ctlcurves" for this simulated data set when $d_1 = d_2 = 1$ and $q_g = 1$, $g = 1, \ldots, K$ (these values are chosen just to have an idea of sensible K and α values for this data set). The examination of these "ctlcurves" suggests that more than $K = 2$ clusters are needed if $\alpha = 0$, as a noticeable increase in the "ctlcurves" when increasing K can be seen, because extra clusters are needed in order to accommodate the outlying spectra. We can also see that increasing K from $K = 2$ is not clearly needed when considering trimming levels $\alpha \geq 0.1$, as the increments in the associated curves are not so obvious. Therefore, these "ctlcurves" suggest $K = 2$ and α around 0.1 as a sensible choice for these two tuning parameters. More details about the use of these "ctlcurves" can be found in [23].

Once the "ctlcurves" give us an idea of the number of clusters to be detected, the trimming proportion also suggested by "ctlcurves" can be further checked and improved if needed. For this purpose, we can use a graphical procedure described in [36], which is based on the evaluation of the rates of increase in the function mapping α onto $D(x_{([n\alpha])}; \widehat{\theta}_{\alpha_0})$, where $\widehat{\theta}_{\alpha_0}$ are the optimal parameter values when α_0 is the trimming level that we want to check. $D(x_{(1)}; \theta) \leq D(x_{(2)}; \theta) \leq \cdots \leq D(x_{(n)}; \theta)$ are the sorted contributions of our n observations to the target function (see Algorithm 1 in Sect. 3). For instance, the resulting curves when $\alpha_0 = 0.05$, $\alpha_0 = 0.2$ and $\alpha_0 = 0.1$ for the same data set as in Fig. 1 are shown in Fig. 2. For this data set, with a "true" 10% contamination level, Fig. 2a shows how the sorted contributions are still increasing sharply at $\alpha = 0.05$ when a trimming level $\alpha_0 = 0.05$ (smaller than needed) is tried. In contrast, the sorted contributions already show a stable behavior at $\alpha = 0.2$ when an $\alpha_0 = 0.2$ (greater than needed) is tried in Fig. 2b. Finally, Fig.2c shows how a value of α around 0.1 is a sensible choice for the trimming level in this problem.

5 Simulation Study

In order to evaluate the performance of the methodology proposed here, a simulation study was carried out. We now describe the different scenarios and contamination types. As in [17], the simulations are based on autoregressive time series of order 2 and mixtures of

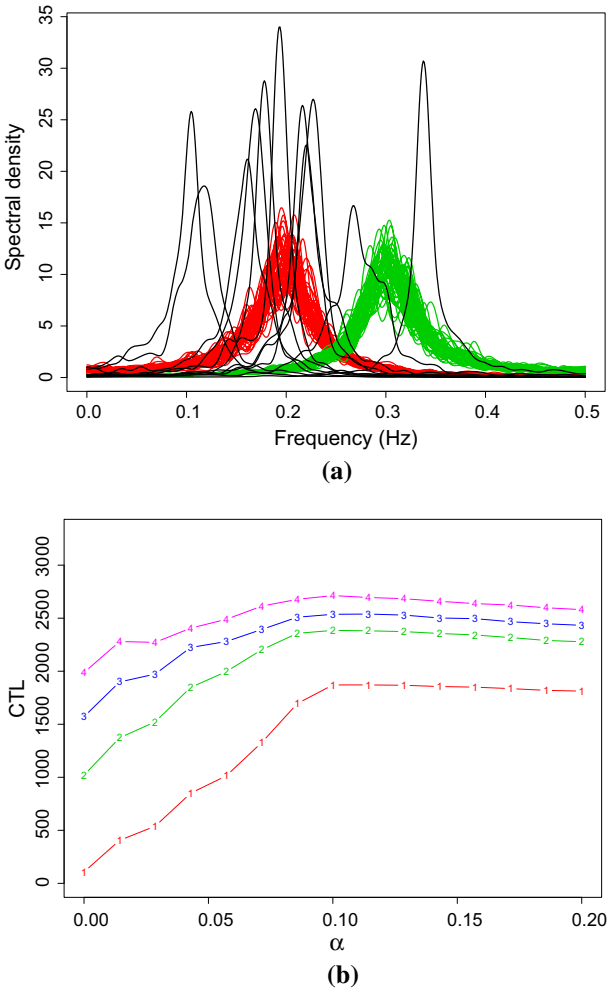

Fig. 1 **a** The simulated dataset with two main clusters and a fraction of 10% outlying spectra. **b** Associated "ctlcurves" curves for this data set in **a** when $d_1 = d_2 = 1$

them. AR(2) processes are defined as $Y_t = u_1 Y_{t-1} + u_2 Y_{t-2} + \epsilon_t$, where ϵ_t is a white noise process. The associated characteristic polynomial is $h(y) = 1 - u_1 y - u_2 y^2$ and its roots, denoted by y_1 and y_2 are related to the oscillatory properties of the time series. If the roots are complex-valued, then they must be conjugate, i.e., $y_1 = \overline{y_2}$ and their polar representation is

$$|y_1| = |y_2| = M \quad \text{and} \quad \arg(y_i) = \frac{2\pi\nu}{w_s} \tag{11}$$

where w_s is the sampling frequency in Hertz; M is the magnitude of the root ($M > 1$ for causality) and ν the frequency index, $\nu \in (0, w_s/2)$. The spectrum of the AR(2) process will have modal frequency at ν, which will be broader as $M \to \infty$ and narrower when $M \to 1^+$. Then, given (ν, M, w_s) and with $\omega_0 = \frac{2\pi\nu}{w_s}$, we have

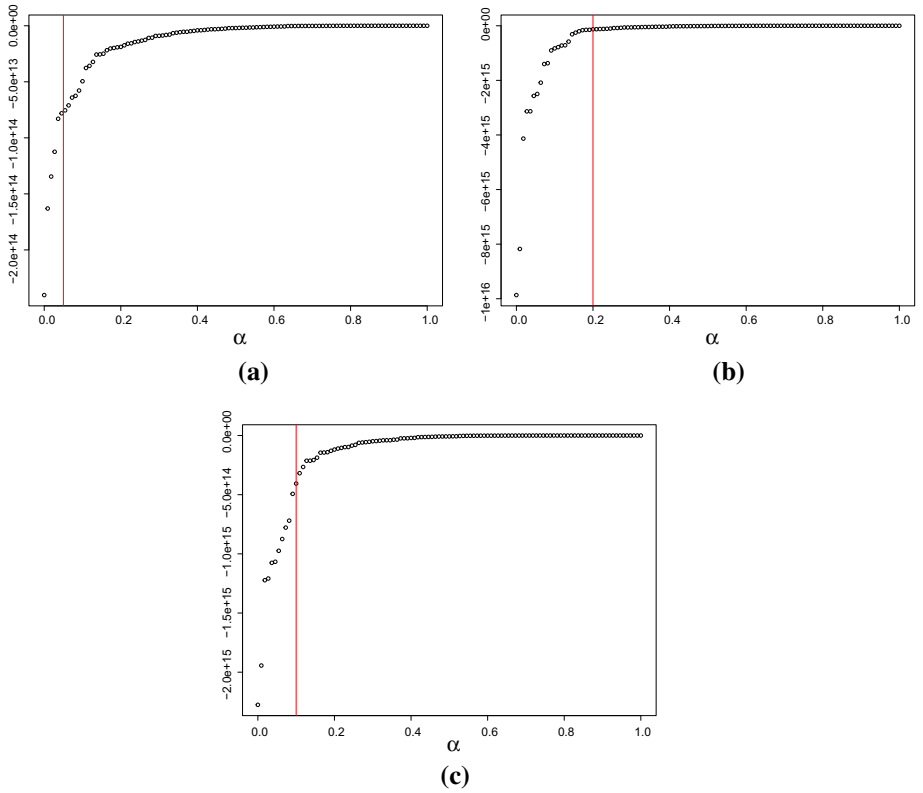

Fig. 2 Graphical procedure for determining α for the data set depicted in Fig. 1: $\alpha_0 = 0.05$ in **a**, $\alpha_0 = 0.2$ in **b** and $\alpha_0 = 0.1$ in **c**

$$u_1 = \frac{2\cos(\omega_0)}{M} \quad \text{and} \quad u_2 = -\frac{1}{M^2}. \tag{12}$$

Two groups of 50 AR(2) time series each were simulated, with parameters $\nu_1 = 0.21$, $\nu_2 = 0.22$, $M_1 = M_2 = 1.15$, $w_s = 1$ and length $\tau = \tau_1 = \tau_2 = 1000$. From the simulated time series, the spectral densities were estimated using a smoothed lag-window estimator with a Parzen window and bandwidth $100/\tau$. The estimated spectral densities for both clusters are shown in Fig. 3a. The functional form of the estimated spectral densities was recovered using a B-spline basis of degree 3 with 14 equispaced nodes and smoothing parameter $\lambda = 0.000003$ (see e.g. [35], Ch. 3) We want to test the performance of the different algorithms in recovering these two groups, even in the presence of contaminating data. In the absence of contamination we have 100 observations divided into two groups.

Before describing the contamination schemes considered, we introduce the mixtures of AR(2) processes that will be used in some cases. Let Y_t^1 and Y_t^2 be two AR(2) processes with parameters M_1 and M_2, and ν_1 and ν_2. Their mixture is given by $Y_t = a_1 Y_t^1 + a_2 Y_t^2 + \epsilon_t$, where a_1 and a_2 are the weights and ϵ_t is a white noise process. This mixture of AR(2) processes creates a signal that combines the oscillatory behavior of the original Y_t^1 and Y_t^2 time series.

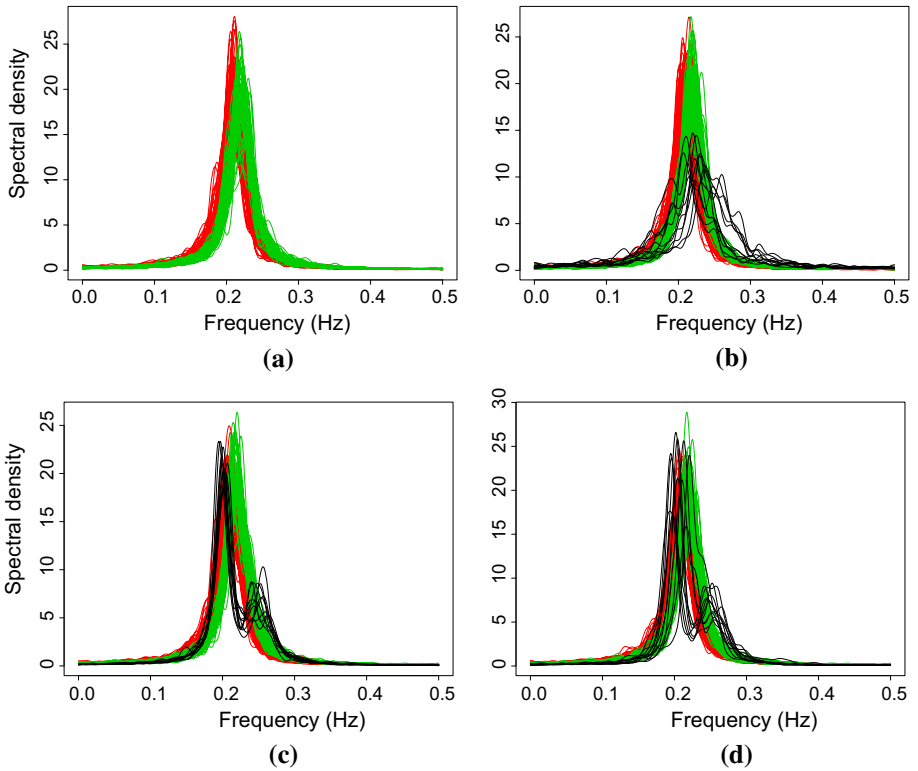

Fig. 3 Spectral density of the simulated time series: **a** No contamination, **b** contamination type (i), **c** contamination type (ii) and **d** contamination type (iii)

Starting from the two groups of 50 AR(2) time series described in the beginning of this section, which are considered as the clean data, we added another 11 time series (around 10% contamination level) using the following schemes:

(i) AR(2) processes with parameters ν_i chosen randomly with uniform distribution in the interval (.20, .25), denoted as $U(.20, .25)$, $i = 1, \ldots, 11$; $M = 1.2$ and $w_s = 1$. This means that the contaminating curves have less energy than the series in the clusters (see Fig. 3b).
(ii) A mixture of two AR(2) processes having parameters $\nu_1 = .20$ and $\nu_2 = .25$; $M_1 = 1.05$ and $M_2 = 1.1$ and $w_s = 1$ (see Fig. 3c).
(iii) A mixture of two $AR(2)$ processes with random parameters $\nu_1 = U(.19, .22)$ and $\nu_2 = U(.24, .26)$; $M_1 = 1.05$ and $M_2 = 1.1$, and $w_s = 1$ (see Fig. 3d).

In order to test the performance of the proposed methodology based on robust functional clustering (RFC), the simulated time series and their estimated spectral densities were used to compare with the results obtained when using the "Funclust" algorithm [28], the hierarchical methods using the total variation distance: "HSMClust" [17], "TVDClust" [3] and the Kullback–Leibler divergence "KLClust" [34].

It is important to recall that we assume the q_g dimensions in the RFC procedure to be unknown parameters and that the BIC criterion is used to estimate them. The results in [36] already show the importance of trimming. Trimming levels $\alpha = 0$ and $\alpha = 0.1$ are used. As

regards the constraints, we are assuming $d_1 = d_2$ to simplify the simulation study. Values of $d_1 = d_2 = 3$, $d_1 = d_2 = 10$ and $d_1 = d_2 = 10^{10}$ (i.e., almost unconstrained in this last case) were used. We always return the best solution in terms of the highest BIC value for each combination of all those fixed values of trimming level and constraints. We use $B = 100$ random initializations with $C = 20$ iterations.

For the "Funclust" method we have used the library `Funclustering` [40] in R where the EM algorithm has been initialized with the best solutions out of 20 "short" EM algorithms with only 20 iterations and threshold values of $\varepsilon = 0.001, 0.05$ and 0.1 in the Cattell test. In case of the agglomerative methods we use the library `HSMClust` in R for "HSMClust" [2], "TVDClust" and "KLClust" by means of the algorithms described in [3,34].

Figure 4 shows the results for the simulation study. This figure is composed of a matrix of graphs, where the rows correspond to the different contamination schemes (uncontaminated in the first row) while the columns correspond to the methodologies tested. The first column corresponds to "Funclust", the second to "HSMClust", the third to "KLClust", the fourth to "TVDClust" and finally, the fifth column shows the results for the procedure based on robust functional clustering (RFC) with trimming levels $\alpha = 0$ (untrimmed), $\alpha = 0.1$ and three constraint levels $d_1 = d_2 = 3$, $d_1 = d_2 = 10$ and $d_1 = d_2 = 10^{10}$. The x-axis

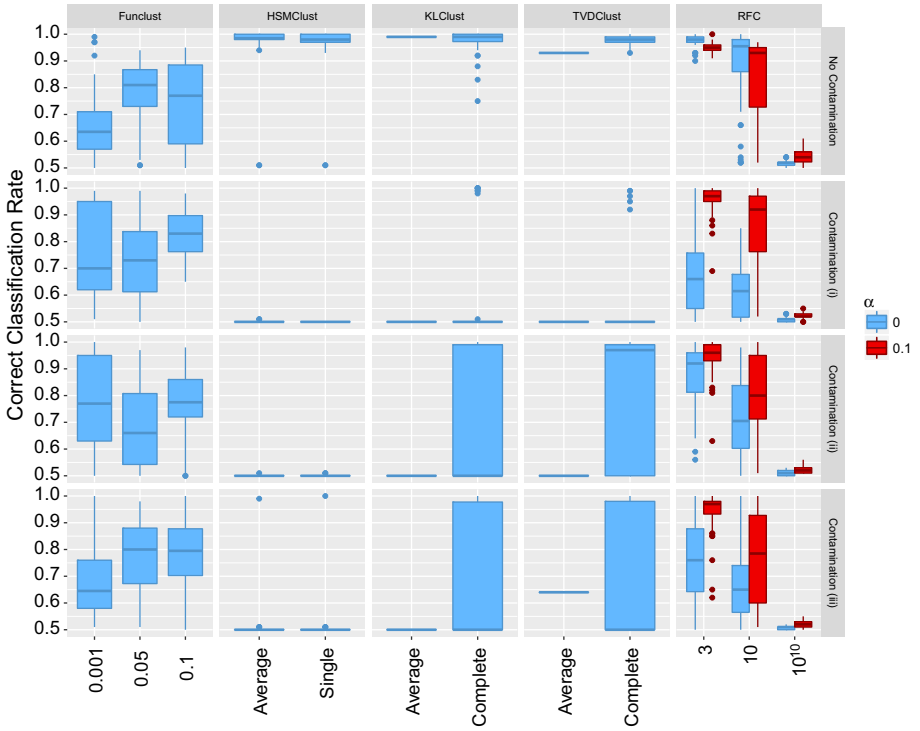

Fig. 4 Correct classification rate (CCR) for the five methods considered, represented in different columns. Rows correspond to the different contamination schemes described previously in this section, starting with no contamination in the first row and following with contamination schemes (i), (ii) and (iii) described in the text. Constraint levels $d_1 = d_2 = 3$, 10 and 10^{10}, trimming levels $\alpha = 0$ and 0.1 were used for the RFC method. Threshold values $\varepsilon = 0.001, 0.05$ and 0.1 are used for the "Cattell" procedure in "Funclust". Single and average linkage aggregation procedures are used for "HSMClust" while average and complete linkage are used for "TVDClust" and "KLClust"

corresponds to the threshold applied in the Cattell test for "Funclust", the linkage function, for the "HSMClust", "TVDClust" and "KLClust" agglomerative methods, and constraints values $d_1 = d_2$ for RFC while the y-axis corresponds to the correct classification rate (CCR).

Results show that the hierarchical methods, "HSMClust", "KLClust" and "TVDClust" are better in the absence of contamination, giving very consistent results. However, their performance degrades sharply in the presence of noise. This is not surprising since these procedures were not designed to handle contamination in the sample. The joint use of trimming and constraints in RFC improve the results (CCR) substantially. Results are very good for small ($d_1 = d_2 = 3$) and moderate ($d_1 = d_2 = 10$) values of the constraint constants, while for high values the results are poor. Very high values for these constants are equivalent to having unconstrained parameters. The use of trimming also turns out to be very useful in all the contaminated cases while the results are not severely affected by trimming in the uncontaminated case.

In the presence of contamination, the results for "Funclust", "HSMClust", "KLClust" and "TVDClust" fall below those of RFC when applying the $\alpha = 0.1$ trimming and small/moderate values d_1 and d_2 for the variance parameters.

6 Analysis of Real Data

In this section, we will consider an application to the analysis of Electrocardiogram (ECG) data. The data sets come from several databases in PhysioBank [26] (https://physionet.org/physiobank/). We consider data from 22 subjects in the malignant ventricular arrhythmia database and 26 normal subjects, 8 from the MIT arrhythmia database and 18 from the normal sinus rhythm database. For each subject the data corresponds to 3 min ECG recorded at frequencies ranging from 128 to 360 Hz. depending on the database. Additionally, 8 fabricated time series were added as noise. These were constructed by mixing amplitude-augmented normal ECG recordings from the MIT arrhythmia database with a sinusoidal signal with random uniform amplitude and random frequency.

Several authors have considered the use of clustering techniques on ECG data. We only mention a few, as examples. [30] compare subjects with malignant ventricular arrhythmia, supraventricular arrhythmia and normal subjects. Their procedure is based on fitting ARIMA models and using the partition around medoids method with various similarity measures. [4] also consider patients with the same conditions and their procedure is based on fitting ARMA models to clipped data. Finally, [12] compare supraventricular arrhythmia patients with healthy subjects. They use the autoregressive distance between ARIMA processes.

In our case, the spectral densities were estimated by a smoothed lag-window estimator with a Parzen window and bandwidth 100 points, using the WAFO toolbox in Matlab. The estimated spectral densities are shown in Fig. 5. The RFC method was applied to this data set in order to obtain a clustering for the time series. The functional form of the data was recovered using B-splines of order 3 with 19 equispaced nodes. We use 100 initializations with 20 iterations each. The constraint levels considered were $d_1 = d_2 = 1, 3, 5, 10^{10}$, and the trimming levels were 5, 10, 15 and 20%.

The results for the correct classification rates (CCR) obtained with RFC are given in Table 1. We can see that the CCR clearly improve when the trimming level α allows to discard the $8/(26 + 22 + 8) = 0.125$ fraction of "gross outliers" added. Moreover, we can see that a trimming level α slightly greater than needed does not deteriorate the method's performance. Also, allowing certain flexibility to the cluster variabilities, by considering

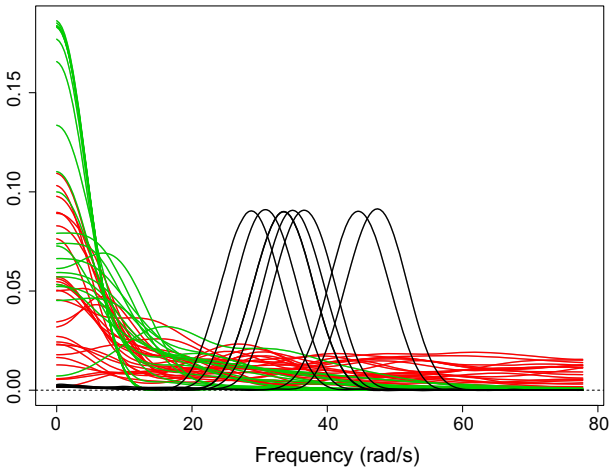

Fig. 5 Original estimated spectral densities. Red is used for those corresponding to the malignant ventricular arrhythmia patients and green for those corresponding to normal subjects. The black curves represent the 8 fabricated time series that were added as noise. (Color figure online)

Table 1 Correct classification rates for different combinations of the α and $d_1 = d_2 = d$ parameters for the proposed methodology

	α				
d	0.00	0.05	0.10	0.15	0.20
1	0.54	0.56	0.56	0.75	0.73
3	0.54	0.56	0.77	0.79	0.83
5	0.54	0.60	0.73	0.79	0.73
10	0.54	0.54	0.79	0.81	0.77
10^{10}	0.60	0.56	0.56	0.69	0.71

Table 2 Correct classification rate for the four clustering procedures, ECG data

Funclust		TVDClust		HSMClust	
0.001	0.708	Average	0.542	Average	0.542
0.05	0.688	Complete	0.646		
0.1	0.729				

$d_1 = d_2 = d > 1$, serves to improve slightly the CCRs but too high a flexibility, by considering $d_1 = d_2 = d = 10^{10}$, seems to reduce the CCRs.

Three other clustering procedures considered previously were applied to this contaminated sample and the results were compared using 2 groups. In the case of the RFC methodology, the assignments based on ' posterior" probabilities were considered for the wrongly trimmed observations. Results are given in Table 2, which should be compared with Table 1. For the "Funclust" procedure three values of the Cattel threshold were considered, 0.001, 0.05 and 0.1.

The best results are obtained with RFC with $\alpha = 0.2$ and $d_1 = d_2 = d = 3$ (Figs. 6, 7). "Funclust" gives reasonable values in the presence of noise. As regards "TVDClust" and "HSMCllust" results are generally poor.

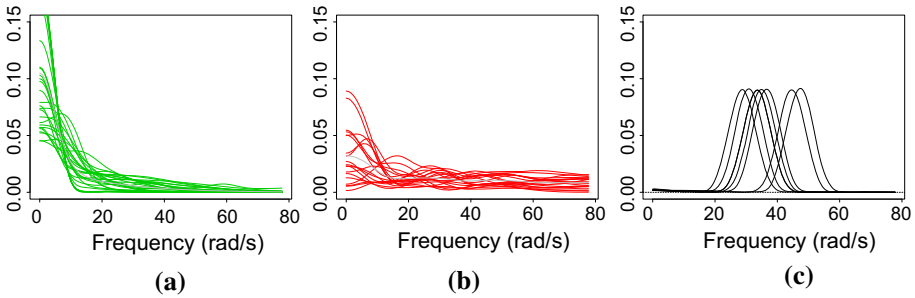

Fig. 6 Results of applying the proposed methodology with $\alpha = 0.15$ and $d_1 = d_2 = 10$

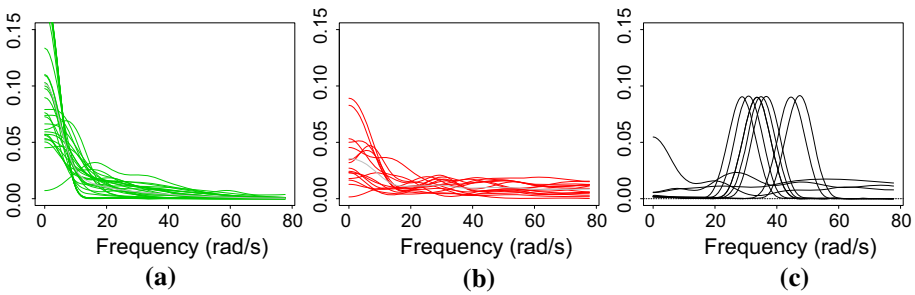

Fig. 7 Results of applying the proposed methodology with $\alpha = 0.2$ and $d_1 = d_2 = 3$

To reinforce these claims, Fig. 8 shows examples of the effects that the presence of noise may have on the clustering methods considered. The first row shows the two original groups, which correspond to normal subjects and malignant ventricular arrhythmia patients. The following rows correspond to "Funclust" with parameters 0.1 and 0.05, and "TVDClust" with the average linkage. The results for "TVDClust" with complete linkage and "HSMClust" with average linkage (not included) coincide exactly with this last graph For all the methods, noise affects the composition of the groups, and, in the case of "TDVClust", some of the noise curves are clustered as a group while the bona-fide data together with the rest of the noise curves constitute the other group, giving a clear example of the type of extreme clustering errors that may be obtained in the presence of noise.

As commented by one of the anonymous reviewers, it may be interesting to extend this approach to "multi-view" clustering, which would allow us to consider multi-channel ECG [11].

7 Conclusions

A feasible methodology of robust clustering for stationary time series has been proposed and illustrated. The key idea behind the algorithm presented is the use of estimated spectral densities of the time series, that are then considered as functional data. A robust model-based algorithm based on approximation of the "density" for functional data, together with the simultaneous use of trimming and constraints is then used to cluster the original time series.

The use of trimming tools protects the estimation of the parameters against effect of outlying curves, while the constraints avoid the presence of spurious clusters in the solution

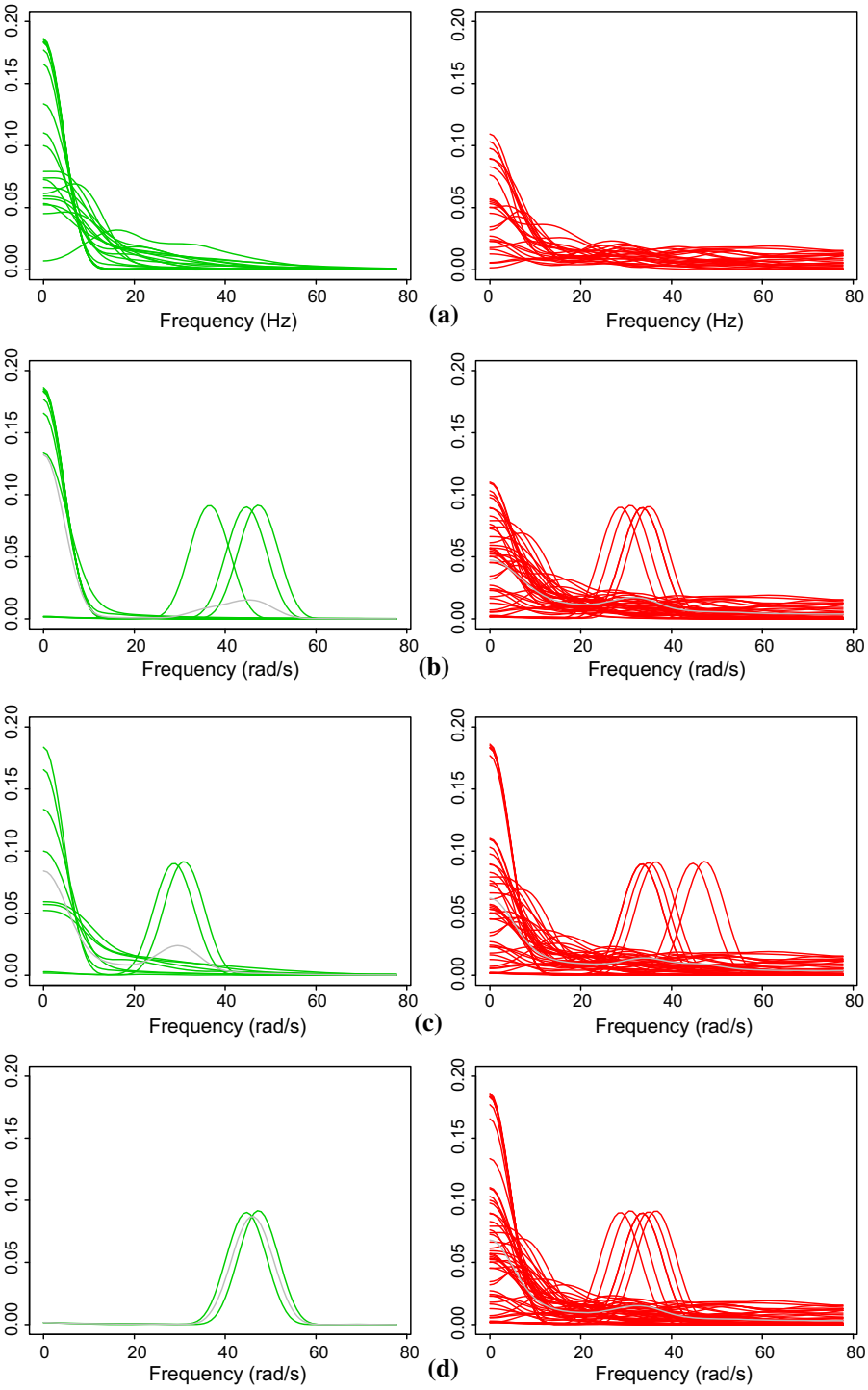

Fig. 8 Groups obtained with the different clustering methods. **a** Original groups, **b** "Funclust" with parameter 0.1, **c** "Funclust" with parameter 0.05, and **d** "TVDClust" with average linkage

and improve the performance of the algorithms. The simulation study shows that the joint use of constraints and trimming tools improve results of the clustering algorithm in the presence of outliers, in comparison to some other procedures for time series and functional data clustering, not designed to work with contamination. The real data example shows that the proposed RFC method for time series clustering has a good performance, with or without the presence of outlying curves. The trimmed curves often correspond to curves with different characteristics to the rest. Moreover, in the presence of contamination, the RFC method is able to detect almost all the outliers in the data. In fact, we conclude that the proposed robust methodology can be a useful tool to detect contamination and groups in a time series data sets simultaneously.

However, this methodology has some limitations. The choice of trimming level α and the choice of the scatter constraints constants d_1 and d_2, can be subjective and sometimes depend on the final purpose of the cluster analysis. For this reason, we always recommend the use of different values of trimming and constraint, monitoring the effect in the clustering partition of these choices. The development of more automated selection procedures for these values may be considered as an open problem for future research.

Acknowledgements Research by DRG and JO was partially supported by Conacyt, Mexico Project 169175 Análisis Estadístico de Olas Marinas, Fase II, Research by LA G-E and A M-I was partially supported by the Spanish Ministerio de Economía y Competitividad, grant MTM2017-86061-C2-1-P, and by Consejería de Educación de la Junta de Castilla y León and FEDER, Grants VA005P17 and VA002G18.

References

1. Aghabozorgi S, Shirkhorshidi AS, Wah TY (2015) Time-series clustering: a decade review. Inf Syst 53:16–38
2. Alvarez-Esteban PC, Euán C, Ortega J (2016) Statistical analysis of stationary intervals for random waves. Proc Int Offshore Polar Eng Conf 3:305–311
3. Alvarez-Esteban PC, Euán C, Ortega J (2016) Time series clustering using the total variation distance with applications in oceanography. Environmetrics 27(6):355–369
4. Bagnall AJ, Janacek GJ (2004) Clustering time series from ARMA models with clipped data. In: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining, KDD '04. ACM, New York, pp 49–58
5. Bahadori MT, Kale DC, Fan Y, Liu Y (2015) Functional subspace clustering with application to time series. In: Proceedings of the 32nd international conference on machine learning, pp 228–237
6. Bouveyron C, Jacques J (2011) Model-based clustering of time series in group-specific functional subspaces. Adv Data Anal Classif 5(4):281–300
7. Caiado J, Crato N, Peña D (2006) A periodogram-based metric for time series classification. Comput Stat Data Anal 50(10):2668–2684
8. Caiado J, Crato N, Peña D (2009) Comparison of times series with unequal length in the frequency domain. Commun Stat Simul Comput 38(3):527–540
9. Caiado J, Maharaj EA, D'Urso P (2015) Time Series Clustering, Chapter 12. CRC Handbooks of Modern Statistical Methods. Chapman & Hall, London, pp 241–263
10. Cerioli A, García-Escudero LA, Mayo-Iscar A, Riani M (2017) Finding the number of normal groups in model-based clustering via constrained likelihoods. J Comput Graph Stat 27:404
11. Chao G, Sun S, Bi J (2017) A survey on multi-view clustering. arXiv preprint arXiv:1712.06246
12. Corduas M, Piccolo D (2008) Time series clustering and classification by the autoregressive metric. Comput Stat Data Anal 52:1860–1872
13. Cuesta-Albertos JA, Fraiman R (2007) Impartial trimmed k-means for functional data. Comput Stat Data Anal 51(10):4864–4877
14. Delaigle A, Hall P (2010) Defining probability density for a distribution of random functions. Ann Stat 38(2):1171–1193
15. D'Urso P, De Giovanni L, Massari R (2015) Time series clustering by a robust autoregressive metric with application to air pollution. Chemom Intell Lab Syst 141:107–124

16. D'Urso P, De Giovanni L, Massari R (2016) Garch-based robust clustering of time series. Fuzzy Sets Syst 305:1–28 (**Theme: Classification, Recognition and Clustering**)
17. Euán C, Ombao H, Ortega J (2018) The hierarchical spectral merger algorithm: a new time series clustering procedure. J Classif 35:71–99. https://doi.org/10.1007/s00357-018-9250-5
18. Ferraty F, Vieu P (2006) Nonparametric functional data analysis. Springer series in statistics. Springer, New York
19. Fritz H, García-Escudero LA, Mayo-Iscar A (2013) A fast algorithm for robust constrained clustering. Comput Stat Data Anal 61:124–136
20. Fu T (2011) A review on time series data mining. Eng Appl Artif Intell 24:164–181
21. García-Escudero LA, Gordaliza A (1999) Robustness properties of k means and trimmed k means. J Am Stat Assoc 94(447):956–969
22. García-Escudero LA, Gordaliza A (2005) A proposal for robust curve clustering. J Classif 22(2):185–201
23. García-Escudero LA, Gordaliza A, Matrán C, Mayo-Iscar A (2011) Exploring the number of groups in robust model-based clustering. Stat Comput 21(4):585–599
24. García-Escudero LA, Gordaliza A, Mayo-Iscar A (2014) A constrained robust proposal for mixture modeling avoiding spurious solutions. Adv Data Anal Classif 8(1):27–43
25. García-Escudero LA, Gordaliza A, Matrán C, Mayo-Iscar A (2015) Avoiding spurious local maximizers in mixture modeling. Stat Comput 25(3):619–633
26. Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ch Ivanov P, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE (2000) PhysioBank, PhysioToolkit, and PhysioNet: somponents of a new research resource for complex physiologic signals. Circulation 101(23):e215–e220. https://doi.org/10.1161/01.CIR.101.23.e215
27. Huang D, Wang CD, Lai JH (2018) Locally weighted ensemble clustering. IEEE Trans Cybernet 48(5):1460–1473
28. Julien J, Cristian P (2013) Funclust: a curves clustering method using functional random variables density approximation. Neurocomputing 112:164–171
29. James GM, Sugar CA (2003) Clustering for sparsely sampled functional data. J Am Stat Assoc 98(462):397–408
30. Kalpakis K, Gada D, Puttagunta V (2001) Distance measures for effective clustering of ARIMA time-series. In: Proceedings 2001 IEEE international conference on data mining, pp 273–280
31. Liao TW (2005) Clustering of time series data: a survey. Pattern Recognit 38:1857–1874
32. Maharaj EA, D'Urso P (2011) Fuzzy clustering of time series in the frequency domain. Inf Sci 181(7):1187–1211
33. Montero P, Vilar J (2014) TSclust: an R package for time series clustering. J Stat Softw 62:1–43
34. Preuss P, Vetter M, Dette H (2013) Testing semiparametric hypotheses in locally stationary processes. Scand J Stat 40(3):417–437
35. Ramsay JO, Silverman BW (2005) Functional data analysis. Springer series in statistics, 2nd edn. Springer, New York
36. Rivera-García D, García-Escudero LA, Mayo-Iscar A, Ortega J (2018) Robust clustering for functional data based on trimming and constraints. Adv Data Anal Classif. https://doi.org/10.1007/s11634-018-0312-7
37. Rivera-García D, García-Escudero LA, Mayo-Iscar A, Ortega J (2017) Robust clustering for time series using spectral densities and functional data analysis. In: International work-conference on artificial neural networks, LNCS 10306. Springer, pp 142–153
38. Rousseeuw PJ, Van Driessen K (1999) A fast algorithm for the minimum covariance determinant estimator. Technometrics 41:212–223
39. Shumway RH, Stoffer DS (2010) Time series analysis and its applications: with R examples, 3rd edn. Springer, Berlin
40. Soueidatt M (2014) Funclustering: a package for functional data clustering. R package version 1.0.1
41. Vidal R (2011) Subspace clustering. IEEE Signal Process Mag 28(2):52–68
42. Wu EHC, Yu PLH (2006) Iclus: a robust and scalable clustering model for time series via independent component analysis. Int J Syst Sci 37(13):987–1001