fdANOVA: An R Software Package for Analysis of Variance for Univariate and Multivariate Functional Data

Tomasz Górecki
Adam Mickiewicz University

Łukasz Smaga
Adam Mickiewicz University

Abstract

This introduction to the R package **fdANOVA** is a (slightly) modified version of Górecki T and Smaga Ł (2017). "**fdANOVA**: An R Software Package for Analysis of Variance for Univariate and Multivariate Functional Data." (under review).

Functional data, i.e., observations represented by curves or functions, frequently arise in various fields. The theory and practice of statistical methods for such data is referred to as functional data analysis (FDA) which is one of major research fields in statistics. The practical use of FDA methods is made possible thanks to availability of specialized and usually free software. In particular, a number of R packages is devoted to these methods. In the paper, we introduce a new R package **fdANOVA** which provides an access to a broad range of global analysis of variance methods for univariate and multivariate functional data. The implemented testing procedures are briefly overviewed and illustrated by examples on a well known functional data set. To reduce the computation time, parallel implementation is developed and its efficiency is empirically evaluated. Since some of the implemented tests have not been compared in terms of size control and power yet, appropriate simulations are also conducted. Their results can help in choosing proper testing procedures in practice.

Keywords: analysis of variance, functional data, **fdANOVA**, R.

1. Introduction

In recent years considerable attention has been devoted to analysis of so called functional data. The functional data are represented by functions or curves which are observations of a random variable (or random variables) taken over a continuous interval or in large discretization of it. Sets of functional observations are peculiar examples of the high-dimensional data where the number of variables significantly exceeds the number of observations. Such data are often gathered automatically due to advances in modern technology, including computing environments. The functional data are naturally collected in agricultural sciences, behavioral sciences, chemometrics, economics, medicine, meteorology, spectroscopy, and many others. The main purpose of functional data analysis (FDA) is to provide tools for statistically describing and modelling sets of functions or curves. The monographs by Ramsay and Silverman (2005), Ferraty and Vieu (2006), Horváth and Kokoszka (2012) and Zhang (2013) present a broad perspective of the FDA solutions. The following problems for functional data are commonly studied (see also the review papers of Cuevas 2014; Wang, Chiou, and Müller 2016): analysis of variance (Faraway 1997; Cuevas, Febrero, and Fraiman 2004; Shen and Faraway 2004; Zhang and Chen 2007; Cuesta-Albertos and Febrero-Bande 2010; Zhang 2011, 2013; Zhang and Liang 2014; Górecki and Smaga 2015; Zhang, Cheng, Tseng, and Wu 2016), canonical correlation analysis (Krzyśko and Waszak 2013; Keser and Kocakoç 2015), classification (James and Hastie 2001; Delaigle and Hall 2012, 2013; Chang, Chen, and Ogden 2014), cluster analysis (Giacofci, Lambert-Lacroix, Marot, and Picard 2013; Coffey, Hinde, and Holian 2014; Yamamoto and Terada 2014), outlier detection (Febrero-Bande, Galeano, and González-Manteiga 2008), principal component analysis (Boente, Barrera, and Tyler 2014; Fremdt, Horváth, Kokoszka, and Steinebach 2014), regression analysis (Chen, Hall, and Müller 2011; Hilgert, Mas, and Verzelen 2013; Morris 2015; Robbiano, Saumard, and Curé 2015; Peng, Zhou, and Tang 2016), repeated measures analysis (Martínez-Camblor and Corral 2011; Smaga 2017), simultaneous confidence bands (Degras 2011; Cao, Yang, and Todem 2012; Ma, Yang, and Carroll 2012). The above references concern the univariate functional data. However, some methods have their multivariate counterparts, e.g., analysis of variance (Górecki and Smaga 2017a), canonical correlation analysis (Górecki, Krzyśko, Waszak, and Wołyński 2016a), classification (Górecki, Krzyśko, and Wołyński 2015; Górecki *et al.* 2016a; Górecki, Krzyśko, and Wołyński 2016b), cluster analysis (Tokushige, Yadohisa, and Inada 2007; Jacques and Preda 2014; Park and Ahn 2016), linear regression and prediction (Chiou, Yang, and Chen 2016; Krzyśko and Smaga 2017), principal component analysis (Berrendero, Justel, and Svarc 2011; Chiou, Chen, and Yang 2014), statistical modelling (Chiou and Müller 2014). Some examples of applications of functional data analysis can be found in Ogden, Miller, Takezawa, and Ninomiya (2002), Pfeiffer, Bura, Smith, and Rutter (2002), Rossi, Wang, and Ramsay (2002), Jank and Shmueli (2006), Febrero-Bande *et al.* (2008), Bobelyn, Serban, Nicu, Lammertyn, Nicolai, and Saeys (2010), Tarrío-Saavedra, Naya, Francisco-Fernández, Artiaga, and Lopez-Beceiro (2011) and Long, Li, Wang, and Cheng (2012).

Many methods for functional data analysis have been already implemented in the R software (R Core Team 2017). The packages **fda** (Ramsay, Hooker, and Graves 2009; Ramsay, Wickham, Graves, and Hooker 2014) and **fda.usc** (Febrero-Bande and Oviedo de la Fuente 2012) are the biggest and probably the most commonly used ones. The first package includes the techniques for functional data in the Hilbert space $L_2(I)$ of square integrable functions over an interval $I = [a, b]$. On the other hand, in the second one, many of the methods implemented do not need such assumption and use only the values of functions evaluated on a grid of discretization points (also non-equally spaced). There is also a broad range of R packages containing solutions for more particular functional data problems: Widely used principal components analysis for functional data implemented in the packages **fda** and **fda.usc** is also available in **fdapace** (Dai, Hadjipantelis, Ji, Müller, and Wang 2016), **fpca** (Peng and Paul 2011), **fdasrvf** (Tucker 2017) and **MFPCA** (Happ 2017; Happ and Greven 2016) ones. For classification and cluster analysis, the packages **classiFunc** (Maierhofer 2017), **fdakma** (Parodi, Patriarca, Sangalli, Secchi, Vantini, and Vitelli 2015), **Funclustering** (Soueidatt and *. 2014), **funcy** (Yassouridis 2016) and **GPFDA** (Shi and Cheng 2014) can be used. In the packages **fdaPDE** (Lila, Sangalli, Ramsay, and Formaggia 2016), **FDboost** (Brockhaus, Scheipl, Hothorn, and Greven 2015; Brockhaus and Rügamer 2016), **flars** (Cheng and Shi 2016), **funreg** (Dziak and Shiyko 2016) and **refund** (Goldsmith, Scheipl, Huang, Wrobel, Gellar, Harezlak, McLean, Swihart, Xiao, Crainiceanu, and Reiss 2016), a lot of work was done to develop the regression analysis for functional data. The packages **freqdom** (Hormann and Kidzinski 2015), **ftsa** (Shang 2013;

Hyndman and Shang 2016), **ftsspec** (Tavakoli 2015) and **pcdpca** (Kidzinski, Jouzdani, and Kokoszka 2016) provide implementation of various methods of functional time series analysis. Methods for the robust analysis of univariate and multivariate functional data are provided in the **roahd** package (Tarabelloni, Arribas-Gil, Ieva, Paganoni, and Romo 2017). Interval testing procedures for functional data in different frameworks (i.e., one or two-population frameworks, functional linear models) are implemented in the **fdatest** package (Pini and Vantini 2015). The functional data analysis in mixed model framework is implemented in the package **fdaMixed** (Markussen 2014). The functional observations can be visualized by many of plots implemented in the following packages **GOplot** (Walter, Sanchez-Cabo, and Ricote 2015), **rainbow** (Shang and Hyndman 2016) and **refund.shiny** (Wrobel and Goldsmith 2016). The packages **fds** (Shang and Hyndman 2013) and **mfds** (Górecki and Smaga 2017b) contain some interesting functional data sets.

Despite so many R packages for functional data analysis, a broad range of test for a widely applicable analysis of variance problem for functional data was implemented very recently in the package **fdANOVA**. Earlier, only the testing procedures of Cuevas *et al.* (2004) and Cuesta-Albertos and Febrero-Bande (2010) were available in the package **fda.usc**. The package **fdANOVA** is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=fdANOVA`. It is the aim of this package to provide a few functions implementing most of known analysis of variance tests for univariate and multivariate functional data. Most of them are based on bootstrap, permutations or projections, which may be time-consuming. For this reason, the package also contains parallel implementations which enable to reduce the computation time significantly, which is shown in empirical evaluation. Additionally, in this paper, some tests or their implementation versions are compared in terms of size control and power by simulation studies. Those comparisons were not presented anywhere else.

The rest of the paper is organized in the following manner. In Section 2, the problems of the analysis of variance for univariate and multivariate functional data are introduced. A review of most of the known solutions of these problems is also presented there. Some of the testing procedures are slightly generalized. Since it was not easy task to implement many different tests in a few functions, their usage may also be not easy at first. Thus, Section 3 contains a detailed description of (eventual) preparation of data and package functionality as well as a package demonstration on commonly used real data set. A series of experiments evaluating efficiency of parallel implementation is presented in Section 4. Simulation studies comparing some of the implemented tests in terms of size control and power are depicted in Section 5. Finally, Section 6 concludes the paper and notes on future extensions of the **fdANOVA** package.

2. Analysis of variance for functional data

In this section, we briefly describe most of the known testing procedures for the analysis of variance problem for functional data in the univariate and multivariate cases. All of them are implemented in the package **fdANOVA**.

2.1. Univariate case

We consider the l groups of independent random functions $X_{ij}(t)$, $i = 1, \dots, l$, $j = 1, \dots, n_i$

defined over a closed and bounded interval $I = [a, b]$. Let $n = n_1 + \cdots + n_l$. These groups may differ in mean functions, i.e., we assume that $X_{ij}(t)$, $j = 1, \ldots, n_i$ are stochastic processes with mean function $\mu_i(t)$, $t \in I$ and covariance function $\gamma(s, t)$, $s, t \in I$, for $i = 1, \ldots, l$. Of interest is to test the following null hypothesis

$$H_0 : \mu_1(t) = \cdots = \mu_l(t),\ t \in I. \tag{1}$$

The alternative is the negation of the null hypothesis. The problem of testing this hypothesis is known as the one-way analysis of variance problem for functional data (FANOVA).

Many of the tests for (1) are based on the pointwise F test statistic (Ramsay and Silverman 2005) given by the formula

$$F_n(t) = \frac{\mathrm{SSR}_n(t)/(l-1)}{\mathrm{SSE}_n(t)/(n-l)},$$

where

$$\mathrm{SSR}_n(t) = \sum_{i=1}^{l} n_i(\bar{X}_i(t) - \bar{X}(t))^2, \quad \mathrm{SSE}_n(t) = \sum_{i=1}^{l}\sum_{j=1}^{n_i}(X_{ij}(t) - \bar{X}_i(t))^2$$

are the pointwise between-subject and within-subject variations respectively, and $\bar{X}(t) = (1/n)\sum_{i=1}^{l}\sum_{j=1}^{n_i} X_{ij}(t)$ and $\bar{X}_i(t) = (1/n_i)\sum_{j=1}^{n_i} X_{ij}(t)$, $i = 1, \ldots, l$, are respectively the sample grand mean function and the sample group mean functions.

Faraway (1997) and Zhang and Chen (2007) proposed to use only the pointwise between-subject variation and considered the test statistic $\int_I \mathrm{SSR}_n(t)dt$. Tests based on it are called the L^2-norm-based tests. The distribution of this test statistic can be approximated by that of $\beta\chi_d^2$ and comparing the first two moments of these random variables. The parameters β and d were estimated by the naive and biased-reduced methods (see, for instance, Górecki and Smaga 2015, for more detail). Thus we have the L^2-norm-based tests with the naive and biased-reduced methods of estimation of these parameters (the L^{2}N and L^{2}B tests for short). In case of non-Gaussian samples or small sample sizes, the bootstrap L^2-norm-based test is also considered (the L^{2}b tests for short).

A bit different L^2-norm-based test was proposed by Cuevas *et al.* (2004). Namely, they considered $\sum_{1 \le i < j \le l} n_i \int_I (\bar{X}_i(t) - \bar{X}_j(t))^2 dt$ as a test statistic and approximated its null distribution by a parametric bootstrap method via re-sampling the Gaussian processes involved in the limit random expression of their test statistic under H_0. Cuevas *et al.* (2004) investigated two testing procedures (the CH and CS tests for short) under homoscedastic and heteroscedastic cases. The tests differ in estimating the covariance functions.

Following test, which uses both the pointwise between-subject and within-subject variations, is known as the F-type test. More precisely, the test statistic is of the form

$$\frac{\int_I \mathrm{SSR}_n(t)dt/(l-1)}{\int_I \mathrm{SSE}_n(t)dt/(n-l)}. \tag{2}$$

Tests of this type were considered by Shen and Faraway (2004) and Zhang (2011). The null distribution of the above test statistic is approximated by $F_{(l-1)\kappa,(n-l)\kappa}$ distribution and depending on the method of estimation of parameter κ, the F-type tests based on naive and biased-reduced methods of estimation are considered (the FN and FB tests for short). For the same reasons as for L^2-norm-based test, the Fb test is also investigated, i.e., the bootstrap F-type test.

The following testing procedure, which also uses the test statistic (2), is the slightly modified procedure proposed by Górecki and Smaga (2015). Assume that $X_{ij} \in L_2(I)$, $i = 1, \ldots, l, j = 1, \ldots, n_i$, where $L_2(I)$ is a Hilbert space consisting of square integrable functions on I, equipped with the inner product of the form $\langle f, g\rangle = \int_I f(t)g(t)dt$. We represent the functional observations by a finite number of basis functions $\varphi_m \in L_2(I), m = 1, \ldots,$ i.e.,

$$X_{ij}(t) = \sum_{m=1}^{K} c_{ijm}\varphi_m(t), \quad t \in I, \tag{3}$$

where c_{ijm}, $m = 1, \ldots, K$, are random variables such that $\mathrm{Var}(c_{ijm}) < \infty$ and K is sufficiently large (see Ramsay and Silverman 2005, for more details about basis function representation). By the results of Górecki *et al.* (2015), the commonly used least squares method (see, for example, Krzyśko and Waszak 2013) seems to be one of the best for estimation of coefficients c_{ijm}, so we use only that method for this purpose. The value of K can be selected for each observation $X_{ij}(t)$ using certain information criterion, e.g., BIC, eBIC, AIC or AICc. From the values of K corresponding to all observations a modal, minimum, maximum or mean value is selected as the common value for all processes $X_{ij}(t)$. By using (3) and easy modifications of the results obtained by Górecki and Smaga (2015), we proved that the test statistic (2) is equal to

$$\frac{(a-b)/(l-1)}{(c-a)/(n-l)}, \tag{4}$$

where

$$a = \sum_{i=1}^{l} \frac{1}{n_i}\mathbf{1}_{n_i}^\top \mathbf{C}_i^\top \mathbf{J}_{\boldsymbol{\varphi}} \mathbf{C}_i \mathbf{1}_{n_i}, \; b = \frac{1}{n}\sum_{i=1}^{l}\sum_{j=1}^{l} \mathbf{1}_{n_i}^\top \mathbf{C}_i^\top \mathbf{J}_{\boldsymbol{\varphi}} \mathbf{C}_j \mathbf{1}_{n_j}, \; c = \sum_{i=1}^{l} \mathrm{trace}(\mathbf{C}_i^\top \mathbf{J}_{\boldsymbol{\varphi}} \mathbf{C}_i),$$

$\mathbf{1}_a$ is the $a \times 1$ vector of ones, $\mathbf{C}_i = (c_{ijm})_{j=1,\ldots,n_i; m=1,\ldots,K}$ are $K \times n_i$ matrices of coefficients, $i = 1, \ldots, l$, and $\mathbf{J}_{\boldsymbol{\varphi}} := \int_I \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^\top(t)dt$ is the $K \times K$ cross product matrix corresponding to the vector $\boldsymbol{\varphi}(t) = (\varphi_1(t), \ldots, \varphi_K(t))^\top$. So the statistic (2) can be calculated based only on the coefficients c_{ijm} and the matrix $\mathbf{J}_{\boldsymbol{\varphi}}$, which can be approximated by using the function `inprod` from the R package **fda** (Ramsay *et al.* 2009, 2014) (For orthonormal basis, $\mathbf{J}_{\boldsymbol{\varphi}}$ is the identity matrix.). Moreover, observe that any permutation of the observations leaves the values of the sums b and c unchanged. For this reason, Górecki and Smaga (2015) considered the permutation test based on (4). We refer to this test as the FP test. Simulation studies of Górecki and Smaga (2015) suggest that the FP test has better finite sample properties than the F-type and L^2-norm-based tests. Moreover, for short functional data (i.e., observed on a short grid of design time points) it may also be better than the GPF test described in the following paragraph.

In the above test statistics, $\mathrm{SSR}_n(t)$ and $\mathrm{SSE}_n(t)$ were integrated separately. However, by the simulation results of Górecki and Smaga (2015), it follows that for example integrating the whole quotient $\mathrm{SSR}_n(t)/\mathrm{SSE}_n(t)$ is more powerful in many situations. Such test statistic of the form $\int_I F_n(t)dt$ was considered by Zhang and Liang (2014). They proposed the globalizing pointwise F test (the GPF test) based on this test statistic and critical value approximated similarly as for the L^2-norm-based test. Although integration seems to a natural operation on $F_n(t)$ or its part, in some situations other using of $F_n(t)$ may be better in the sense of power, as was shown by Zhang *et al.* (2016). Instead of integral of $F_n(t)$, they used simply

$\sup_{t\in I} F_n(t)$ as a test statistic and simulated the critical value of the resulting Fmaxb test via bootstrapping. By intensive simulation studies, Zhang *et al.* (2016) found that the Fmaxb (resp. GPF) test generally has higher power than the GPF (resp. Fmaxb) test when the functional data are moderately or highly (resp. less) correlated.

A different approach to test the null hypothesis (1) was proposed by Cuesta-Albertos and Febrero-Bande (2010). Their tests are based on the analysis of randomly chosen projections. Suppose that μ_i, $i = 1, \dots, l$ belong to a separable Hilbert space $\mathcal{H}$ endowed with a scalar product $\langle\cdot,\cdot\rangle$, and ξ is a Gaussian distribution on this space and each of its one-dimensional projections is nondegenerate. Let h be a vector chosen randomly from $\mathcal{H}$ using the distribution ξ. When the null hypothesis H_0 holds, then we can easily observe that for every $h \in \mathcal{H}$, the following null hypothesis

$$H_0^h : \langle\mu_1, h\rangle = \dots = \langle\mu_l, h\rangle \tag{5}$$

also holds. Moreover, Cuesta-Albertos and Febrero-Bande (2010) showed that for ξ-almost every h, H_0^h fails, in case of failing of H_0. Therefore, a testing procedure for H_0^h can be used to test H_0. Assuming that $X_{ij} \in L_2(I)$, $i = 1, \dots, l, j = 1, \dots, n_i$, Cuesta-Albertos and Febrero-Bande (2010) propose the following testing procedure, in which k random projections are used:

1. Choose, with Gaussian distribution, functions $h_r \in L_2(I)$, $r = 1, \dots, k$.

2. Compute the projections $P_{ij}^r = \int_I X_{ij}(t)h_r(t)dt / \left(\int_I h_r^2(t)dt\right)^{1/2}$ for $i = 1, \dots, l$, $j = 1, \dots, n_i$, $r = 1, \dots, k$.

3. For each $r \in \{1, \dots, k\}$, apply the appropriate ANOVA test for P_{ij}^r, $i = 1, \dots, l$, $j = 1, \dots, n_i$. Let $p_1, \dots, p_r$ denote the resulting p values.

4. Compute the final p value for H_0 by the formula $\inf\{kp_{(r)}/r, r = 1, \dots, k\}$, where $p_{(1)} \leq \dots \leq p_{(k)}$ are the ordered p values obtained in step 3.

The tests based on the above procedure are referred to as the test based on random projections. Cuesta-Albertos and Febrero-Bande (2010) suggested to use k near 30, which was confirmed by the results of Górecki and Smaga (2017a). However, in the case of unconvincing results of the test, we should use a higher number of projections. We also have to choose a Gaussian distribution and ANOVA test appearing in steps 1 and 3 of the above procedure, respectively. We can do this in many ways and some of them are implemented in the package **fdANOVA** (see Section 3). In step 4, we can also use other final p values instead of Benjamini and Hochberg procedure (Benjamini and Hochberg 1995; Benjamini and Yekutieli 2001), as for example Bonferroni correction. However, according to our experience, the test with corrected p value given in step 4 behaves best under finite samples, so we use only it.

Most of the above testing procedures were compared via simulations in the papers of Górecki and Smaga (2015) and Zhang *et al.* (2016). As mentioned above, the GPF and Fmaxb tests seem to perform best among the tests considered in these articles. However, they were not compared with the testing procedure of Cuesta-Albertos and Febrero-Bande (2010). Such comparison of the finite sample behavior of these tests is given in Section 5. Different variants of the projection method proposed by Cuesta-Albertos and Febrero-Bande (2010) are also investigated there.

2.2. Multivariate case

Now, we study the multivariate version of the ANOVA problem for functional data as well as extensions of certain methods presented in the last section to this problem. The results of this section were mainly obtained by Górecki and Smaga (2017a).

Instead of single functions, we consider independent vectors of random functions $\mathbf{X}_{ij}(t) = (X_{ij1}(t), \dots, X_{ijp}(t))^\top \in SP_p(\boldsymbol{\mu}_i, \boldsymbol{\Gamma})$, $i = 1, \dots, l$, $j = 1, \dots, n_i$ defined over the interval I, where $SP_p(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ is a set of p-dimensional stochastic processes with mean vector $\boldsymbol{\mu}(t)$, $t \in I$ and covariance function $\boldsymbol{\Gamma}(s,t)$, $s, t \in I$. In the multivariate analysis of variance problem for functional data (FMANOVA), we have to test the null hypothesis as follows:

$$H_0 : \boldsymbol{\mu}_1(t) = \dots = \boldsymbol{\mu}_l(t),\ t \in I. \tag{6}$$

The first (permutation) tests are based on a basis function representation of the components of the vectors $\mathbf{X}_{ij}(t)$, $i = 1, \dots, l$, $j = 1, \dots, n_i$, similarly as in the FP test. For this purpose, we assume that $\mathbf{X}_{ij}$ belong to $L_2^p(I)$ – a Hilbert space of p-dimensional vectors of square integrable functions on the interval I, equipped with the inner product: $\langle \mathbf{x}, \mathbf{y} \rangle = \int_I \mathbf{x}^\top(t)\mathbf{y}(t)dt$. Hence we can represent the components of $\mathbf{X}_{ij}(t)$ in a similar way as in (3). Then the vectors are represented as follows

$$\mathbf{X}_{ij}(t) = \begin{pmatrix} \mathbf{c}_{ij1} \\ \vdots \\ \mathbf{c}_{ijp} \end{pmatrix} \boldsymbol{\varphi}(t) = \mathbf{c}_{ij}\boldsymbol{\varphi}(t), \tag{7}$$

where $\mathbf{c}_{ijm} = (c_{ijm1}, \dots, c_{ijmK_m}, 0, \dots, 0) \in R^{KM}$, $\boldsymbol{\varphi}(t) = (\varphi_1(t), \dots, \varphi_{KM}(t))^\top$, $t \in I$ and $i = 1, \dots, l$, $j = 1, \dots, n_i$, $m = 1, \dots, p$, $KM = \max\{K_1, \dots, K_p\}$. The coefficients in $\mathbf{c}_{ij}$ and values of K_m are estimated separately for each feature by using the same methods as described in Section 2.1. Similarly to MANOVA (see Anderson 2003), the following matrices were used in constructing test statistics for FMANOVA problem:

$$\mathbf{E} = \sum_{i=1}^{l}\sum_{j=1}^{n_i}\int_I \left(\mathbf{X}_{ij}(t) - \bar{\mathbf{X}}_i(t)\right)\left(\mathbf{X}_{ij}(t) - \bar{\mathbf{X}}_i(t)\right)^\top dt,$$

$$\mathbf{H} = \sum_{i=1}^{l} n_i \int_I \left(\bar{\mathbf{X}}_i(t) - \bar{\mathbf{X}}(t)\right)\left(\bar{\mathbf{X}}_i(t) - \bar{\mathbf{X}}(t)\right)^\top dt,$$

where $\bar{\mathbf{X}}_i(t) = (1/n_i)\sum_{j=1}^{n_i}\mathbf{X}_{ij}(t)$, $i = 1, \dots, l$ and $\bar{\mathbf{X}}(t) = (1/n)\sum_{i=1}^{l}\sum_{j=1}^{n_i}\mathbf{X}_{ij}(t)$, $t \in I$. Modifying the results of Górecki and Smaga (2017a), we showed that these matrices can be designated only by the coefficient matrices $\mathbf{c}_{ij}$ and appropriate cross product matrix, i.e., $\mathbf{E} = \mathbf{A} - \mathbf{B}$ and $\mathbf{H} = \mathbf{B} - \mathbf{C}$, where

$$\mathbf{A} = \sum_{i=1}^{l}\sum_{j=1}^{n_i}\mathbf{c}_{ij}\mathbf{J}_{\boldsymbol{\varphi}}\mathbf{c}_{ij}^\top,\ \mathbf{B} = \sum_{i=1}^{l}\frac{1}{n_i}\sum_{j=1}^{n_i}\sum_{m=1}^{n_i}\mathbf{c}_{ij}\mathbf{J}_{\boldsymbol{\varphi}}\mathbf{c}_{im}^\top,\ \mathbf{C} = \frac{1}{n}\sum_{i=1}^{l}\sum_{j=1}^{n_i}\sum_{t=1}^{l}\sum_{u=1}^{n_t}\mathbf{c}_{ij}\mathbf{J}_{\boldsymbol{\varphi}}\mathbf{c}_{tu}^\top,$$

and $\mathbf{J}_{\boldsymbol{\varphi}}$ is the $KM \times KM$ cross product matrix corresponding to $\boldsymbol{\varphi}$. The following test statistics for (6) are constructed based on those appearing in MANOVA (Anderson 2003): the Wilk's lambda $\mathrm{W} = \det(\mathbf{E})/\det(\mathbf{E} + \mathbf{H})$, the Lawley-Hotelling trace $\mathrm{LH} = \mathrm{trace}(\mathbf{H}\mathbf{E}^{-1})$, the Pillai trace $\mathrm{P} = \mathrm{trace}(\mathbf{H}(\mathbf{H} + \mathbf{E})^{-1})$, the Roy's maximum root $\mathrm{R} = \lambda_{\max}(\mathbf{H}\mathbf{E}^{-1})$, where

$\lambda_{\max}(\mathbf{M})$ is the maximum eigenvalue of a matrix $\mathbf{M}$. We consider the permutation testing procedures based on these test statistics and refer to them as the W, LH, P and R tests, respectively. Generally, we refer to them as the permutation tests based on a basis function representation for the FMANOVA problem. A quite fast implementation of these permutation tests was obtained by observing that the matrices $\mathbf{A}$ and $\mathbf{C}$ are not changed by any permutation of the data.

The second group of testing procedures for (6) is based on random projections similarly as in the FANOVA test based on random projections. Let a space $\mathcal{H}$ and a distribution ξ be defined as in Section 2.1. Assume that $\mu_{ij} \in \mathcal{H}$, where μ_{ij} are the components of the vectors $\boldsymbol{\mu}_i$, $i = 1, \dots, l$, $j = 1, \dots, p$. If the null hypothesis in (6) holds, then for every $\mathbf{H} = (h_1, \dots, h_p)^\top \in \mathcal{H} \times \cdots \times \mathcal{H}$,

$$H_0^{\mathbf{H}} : (\langle \mu_{11}, h_1 \rangle, \dots, \langle \mu_{1p}, h_p \rangle)^\top = \cdots = (\langle \mu_{l1}, h_1 \rangle, \dots, \langle \mu_{lp}, h_p \rangle)^\top \tag{8}$$

also holds, and further when H_0 fails, for $(\xi \times \cdots \times \xi)$-almost every $\mathbf{H} \in \mathcal{H} \times \cdots \times \mathcal{H}$, $H_0^{\mathbf{H}}$ also fails (see Theorem 2.2 in Górecki and Smaga 2017a). Thus, a test for the MANOVA problem can be used to test the FMANOVA one by using it to test the null hypothesis (8). For this reason, Górecki and Smaga (2017a) investigated the similar testing procedure based on k random projection as described in Section 2.1, which the first three steps are now as follows:

1. Choose, with Gaussian distribution, functions $h_{mr} \in L_2(I)$, $m = 1, \dots, p$, $r = 1, \dots, k$.
2. Compute the projections $P_{ijm}^r = \int_I X_{ijm}(t) h_{mr}(t) dt / \left(\int_I h_{mr}^2(t) dt \right)^{1/2}$ for $i = 1, \dots, l$, $j = 1, \dots, n_i$, $m = 1, \dots, p$, $r = 1, \dots, k$.
3. For each $r \in \{1, \dots, k\}$, apply the appropriate MANOVA test for $\mathbf{P}_{ij}^r = (P_{ij1}^r, \dots, P_{ijp}^r)^\top$, $i = 1, \dots, l$, $j = 1, \dots, n_i$. Let $p_1, \dots, p_r$ denote the resulting p values.

In step 3 of this procedure, the well-known MANOVA tests were applied, namely Wilk's lambda test (Wp test), the Lawley-Hotelling trace test (LHp test), the Pillai trace test (Pp test) and Roy's maximum root test (Rp test). Their permutation versions are also investigated.

By the extensive Monte Carlo simulation studies of Górecki and Smaga (2017a), the performance of the tests considered except the Rp test is very satisfactory under finite samples. Unfortunately, the Rp test does not control the nominal type-I error level, and hence it can not be recommended. The other testing procedures do not perform equally well, and there is no single method performing best. As a supplement to those simulation studies, the finite sample behavior of new variant of the tests based on random projections implemented in the package **fdANOVA** is investigated in simulations of Section 5.

3. R implementation

In this section, we present the R package **fdANOVA** and illustrate the usage of it step by step using certain real data set. First, however, we mention about the eventual preparation of the functional data in the R program to use the functions of our package properly.

3.1. Preparation of the data

In practice, functional samples are not continuously observed, i.e., each function is usually observed on a grid of design time points. In our implementations of FANOVA and FMANOVA tests in the R programming language, all functions are observed on a common grid of design time points equally spaced in the interval $I = [a, b]$. In the case where the design time points are different for different individual functions or not equally spaced in I, we follow the methodology proposed by Zhang (2013). First, we have to reconstruct the functional samples from the observed discrete functional samples using smoothing technique such as regression splines, smoothing splines, P-splines or local polynomial smoothing (see Zhang 2013, Chapters 2-3). For this purpose, in R we can use the function `smooth.spline` from the **stats** package (R Core Team 2017) or functions given in the packages **splines** (R Core Team 2017), **bigsplines** (Helwig 2016), **pspline** (S original by Jim Ramsey. R port by Brian Ripley 2015) and **locpol** (Ojeda Cabrera 2012). After that we discretize each individual function of the reconstructed functional samples on a common grid of $\mathcal{T}$ time points equally spaced in I, and then the implementations of the tests can be applied to discretized samples.

3.2. Package functionality

Now, we describe the implementation of the tests for analysis of variance problem for univariate and multivariate functional data in the R package **fdANOVA**. As we will see many of the implemented tests may be performed with different values of parameters. However, by simulation and real data examples presented in the present and previous papers (see Sections 2 and 5), satisfactory results are usually obtained by using the default values of these parameters. Nevertheless, when the results are unconvincing (e.g., the p values are close to the significance level), we have the opportunity to use other options provided by the functions of the package.

All tests for FANOVA problem presented in Section 2.1 are implemented in the function `fanova.tests`:

```
R> library("fdANOVA")
R> str(fanova.tests)

function(x = NULL, group.label, test = "ALL", params = NULL,
         parallel = FALSE, nslaves = NULL)
```

which is controlled by the following parameters:

- `x` – a $\mathcal{T} \times n$ matrix of data, whose each column is a discretized version of a function and rows correspond to design time points. Its default values is `NULL`, since if the FP test is only used, we can give a basis representation of the data instead of raw observations (see the list `paramFP` below). For any of the other testing procedures, the raw data are needed.

- `group.label` – a vector containing group labels.

- `test` – a kind of indicator which establishes a choice of FANOVA tests to be performed. Its default value means that all testing procedures of Section 2.1 will be used. When we want to use only some tests, the parameter `test` is an appropriate subvector of the following vector of tests' labels:

```
c("FP", "CH", "CS", "L2N", "L2B", "L2b", "FN", "FB", "Fb", "GPF",
  "Fmaxb", "TRP")
```

- `params` – a list of additional parameters for the FP, CH, CS, L^{2}b, Fb, Fmaxb tests and the test based on random projections. Its possible elements and their default values are described below. The default value of this parameter means that these tests are performed with their default values.
- `parallel` – a logical indicating whether to use parallelization.
- `nslaves` – if `parallel = TRUE`, a number of slaves. Its default value means that it will be equal to a number of logical processes of a computer used.

The list `params` can contain all or a part of the elements `paramFP`, `paramCH`, `paramCS`, `paramL2b`, `paramFb`, `paramFmaxb` and `paramTRP` for passing the parameters for the FP, CH, CS, L^{2}b, Fb, Fmaxb tests and the test based on random projections, respectively, to the function `fanova.tests`. They are described as follows. The list

```
paramFP = list(int, B.FP = 1000, basis = c("Fourier", "b-spline", "own"),
               own.basis, own.cross.prod.mat,
               criterion = c("BIC", "eBIC", "AIC", "AICc", "NO"),
               method = c("mode", "min", "max", "mean"),
               minK = NULL, maxK = NULL, norder = 4, gamma.eBIC = 0.5)
```

contains the parameters of the FP test and their default values, where:

- `int` – a vector of two elements representing the interval $I = [a, b]$. When it is not specified, it is determined by a number of design time points.
- `B.FP` – a number of permutation replicates.
- `basis` – a choice of basis of functions used in the basis function representation of the data.
- `own.basis` – if `basis = "own"`, a $K \times n$ matrix with columns containing the coefficients of the basis function representation of the observations.
- `own.cross.prod.mat` – if `basis = "own"`, a $K \times K$ cross product matrix corresponding to a basis used to obtain the matrix `own.basis`.
- `criterion` – a choice of information criterion for selecting the optimum value of K. `criterion = "NO"` means that K is equal to the parameter `maxK` defined below. By (3), we have `BIC`$(X_{ij}) = \mathcal{T}\log(\mathbf{e}_{ij}^\top\mathbf{e}_{ij}/\mathcal{T}) + K\log\mathcal{T}$, `eBIC`$(X_{ij}) = \mathcal{T}\log(\mathbf{e}_{ij}^\top\mathbf{e}_{ij}/\mathcal{T}) + K[\log\mathcal{T} + 2\gamma\log(K_{\max})]$, `AIC`$(X_{ij}) = \mathcal{T}\log(\mathbf{e}_{ij}^\top\mathbf{e}_{ij}/\mathcal{T}) + 2K$ and `AICc`$(X_{ij}) =$ `AIC`$(X_{ij}) + 2K(K+1)/(n-K-1)$, where $\mathbf{e}_{ij} = (e_{ij1}, \ldots, e_{ij\mathcal{T}})^\top$, $e_{ijr} = X_{ij}(t_r) - \sum_{m=1}^{K}\hat{c}_{ijm}\varphi_m(t_r)$, $t_1, \ldots, t_{\mathcal{T}}$ are the design time points, $\gamma \in [0, 1]$, $K_{\max}$ is a maximum K considered and log denotes the natural logarithm.

- `method` – a choice of method for selecting the common value for all observations from the values of K corresponding to all processes.
- `minK` (resp. `maxK`) – a minimum (resp. maximum) value of K. When `basis = "Fourier"`, they have to be odd numbers. If `minK` $=$ `NULL` and `maxK` $=$ `NULL`, we take `minK` $= 3$ and `maxK` equal to the largest odd number smaller than the number of design time points. If `maxK` is greater than or equal to the number of design time points, `maxK` is taken as above. For `basis = "b-spline"`, `minK` (resp. `maxK`) has to be greater (resp. smaller) than or equal to `norder` defined below (resp. $\mathcal{T}$). If `minK` $=$ `NULL` or `minK` $<$ `norder` and `maxK` $=$ `NULL` or `maxK` $> \mathcal{T}$, then we take `minK` $=$ `norder` and `maxK` $= \mathcal{T}$.
- `norder` – if `basis = "b-spline"`, an integer specifying the order of b-splines.
- `gamma.eBIC` – a $\gamma \in [0, 1]$ parameter in the eBIC.

It should be noted that the AICc may choose the finale model with a number K of coefficients close to a number of observations n, when $K_{\max}$ is greater than n. Such selection usually differs from choices suggested by other criterion, but it seems that this does not have much impact on the results of testing.

For the CH and CS (resp. L^{2}b, Fb and Fmaxb) tests, the parameters `paramCH` and `paramCS` (resp. `paramL2b`, `paramFb` and `paramFmaxb`) denote the numbers of discretized artificial trajectories for certain Gaussian processes (resp. bootstrap samples) used to approximate the null distributions of their test statistics. The default value of each of these parameters is 10,000. The parameters of the test based on random projections and their default values are contained in a list

```
paramTRP = list(k = 30, projection = c("GAUSS", "BM"),
                permutation = FALSE, B.TRP = 10000)
```

where:

- `k` – a vector of numbers of projections.
- `projection` – a method of generating Gaussian processes in step 1 of the testing procedure based on random projections presented in Section 2. If `projection = "GAUSS"`, the Gaussian white noise is generated as in the function `anova.RPm` from the R package **fda.usc** (Febrero-Bande and Oviedo de la Fuente 2012). In the second case, the Brownian motion is generated.
- `permutation` – a logical indicating whether to compute p values by permutation method.
- `B.TRP` – a number of permutation replicates.

To perform step 3 of the procedure based on random projections given in Section 2.1, in the package, we use five testing procedures: the standard (`paramTRP$permutation = FALSE`) and permutation (`paramTRP$permutation = TRUE`) tests based on ANOVA F test statistic and ANOVA-type statistic (ATS) proposed by Brunner, Dette, and Munk (1997), as well as the testing procedure based on Wald-type permutation statistic (WTPS) of Pauly, Brunner, and Konietschke (2015).

The function `fanova.tests` returns a list of the class `fanovatests`. This list contains the values of the test statistics, the p values and the parameters used. The results for a given test are given in a list (being an element of output list) named the same as the indicator of a test in the vector `test`. Additional outputs as the chosen optimal length of basis expansion (the FP test), the values of estimators used in approximations of null distributions of test statistics (the L^2N, L^2B, FN, FB, GPF tests) and projections of the data (the test based on random projections) are contained in appropriate lists. The projections of the data are contained in a list of the length equal to length of vector `k`, whose i-th element is an $n \times$ `k[i]` matrix with columns being projections of the data.

The permutation tests based on a basis function representation for FMANOVA problem, i.e., the W, LH, P and R tests are implemented in the function `fmanova.ptbfr`:

```
R> library("fdANOVA")
R> str(fmanova.ptbfr)

function(x = NULL, group.label, int, B = 1000,
         parallel = FALSE, nslaves = NULL,
         basis = c("Fourier", "b-spline", "own"),
         own.basis, own.cross.prod.mat,
         criterion = c("BIC", "eBIC", "AIC", "AICc", "NO"),
         method = c("mode", "min", "max", "mean"),
         minK = NULL, maxK = NULL, norder = 4, gamma.eBIC = 0.5)
```

The parameters `group.label`, `int`, `B`, `parallel`, `nslaves`, `basis`, `norder` and `gamma.eBIC` are the same as in the function `fanova.tests` (`B` corresponds to `B.FP`). The other arguments of `fmanova.ptbfr` are described as follows:

- `x` – a list of $\mathcal{T} \times n$ matrices of data, whose each column is a discretized version of a function and rows correspond to design time points. The mth element of this list contains the data of mth feature, $m = 1, \ldots, p$. Its default values is `NULL`, because a basis representation of the data can be given instead of raw observations (see the parameter `own.basis` below).

- `own.basis` – if `basis = "own"`, a list of length p, whose elements are $K_m \times n$ matrices ($m = 1, \ldots, p$) with columns containing the coefficients of the basis function representation of the observations.

- `own.cross.prod.mat` – if `basis = "own"`, a $KM \times KM$ cross product matrix corresponding to a basis used to obtain the list `own.basis`.

- `criterion` – a choice of information criterion for selecting the optimum value of K_m, $m = 1, \ldots, p$. `criterion = "NO"` means that K_m are equal to the parameter `maxK` defined below.

- `method` – a choice of method for selecting the common value for all observations from the values of K_m corresponding to all processes.

- `minK` (resp. `maxK`) – a minimum (resp. maximum) value of K_m. Further remarks about these arguments are the same as for the function `fanova.tests`.

The function `fmanova.ptbfr` returns a list of the class `fmanovaptbfr` containing the values of the test statistics (`W`, `LH`, `P`, `R`), the p values (`pvalueW`, `pvalueLH`, `pvalueP`, `pvalueR`), chosen optimal values of K_m and the parameters used. This function uses the R package **fda** (Ramsay *et al.* 2014).

The function `fmanova.trp` performs the testing procedures based on random projections for FMANOVA problem (the Wp, LHp, Pp and Rp tests):

```
R> library("fdANOVA")
R> str(fmanova.trp)

function(x, group.label, k = 30, projection = c("GAUSS", "BM"),
         permutation = FALSE, B = 1000,
         parallel = FALSE, nslaves = NULL)
```

The first two parameters of this function as well as the arguments `parallel`, `nslaves` are the same as in the function `fmanova.ptbfr`. The other ones have the same meaning as in the parameter list `paramTRP` of the function `fanova.tests` (`B` corresponds to `B.TRP`). The function `fmanova.trp` returns a list of class `fmanovatrp` containing the parameters and the following elements (|`k`| denotes the length of vector `k`): `pvalues` – a $4 \times |\texttt{k}|$ matrix of p values of the tests; `data.projections` – a list of length |`k`|, whose elements are lists of $n \times p$ matrices of projections of the observations.

The executions of selecting the optimum length of basis expansion by some information criterion, the bootstrap, permutation and projection loops are the most time consuming steps of the testing procedures under consideration. To reduce the computational cost of the procedures, they are parallelized, when the parameter `parallel` is set to `TRUE`. The parallel execution is handled by **doParallel** package (Revolution Analytics and Weston 2015). Some details of the parallel implementation and its efficiency are discussed in Section 4.

In the package, the number of auxiliary functions are also contained. The p values of the tests based on random projections for FANOVA problem against the number of projections are visualized by the function `plot.fanovatests` using the package **ggplot2** (Wickham 2009), which is controlled by the following parameters: `x` – an `fanovatests` object, more precisely, a result of the function `fanova.tests` for the standard tests based on random projections; `y` – an `fanovatests` object, more precisely, a result of the function `fanova.tests` for the permutation tests based on random projections. Similarly, the p values of the Wp, LHp, Pp and Rp tests are plotted by the function `plot.fmanovatrp`. The arguments of this function are as follows: `x` – an `fmanovatrp` object, more precisely, a result of the function `fmanova.trp` for the standard tests; `y` – an `fmanovatrp` object, more precisely, a result of the function `fmanova.trp` for the permutation tests; `withoutRoy` – a logical indicating whether to plot the p values of the Rp test. We can use only one of the arguments `x` and `y`, or both simultaneously.

Using the package **ggplot2** (Wickham 2009), the function `plotFANOVA`:

```
R> library("fdANOVA")
R> str(plotFANOVA)

function(x, group.label = NULL, int = NULL, separately = FALSE,
         means = FALSE, smooth = FALSE, ...)
```

produces a plot showing univariate functional observations with or without indication of groups as well as mean functions of samples. The following parameters control this function:

- `x` – a $\mathcal{T} \times n$ matrix of data, whose each column is a discretized version of a function and rows correspond to design time points.
- `group.label` – a character vector containing group labels. Its default value means that all functional observations are drawn without division into groups.
- `int` – this parameter is the same as in the function `fanova.tests`.
- `separately` – a logical indicating how groups are drawn. If `separately = FALSE`, groups are drawn on one plot by different colors. When `separately = TRUE`, they are depicted in different panels.
- `means` – a logical indicating whether to plot only group mean functions.
- `smooth` – a logical indicating whether to plot reconstructed data via smoothing splines instead of raw data.

The functions `print.fanovatests`, `print.fmanovaptbfr` and `print.fmanovatrp` print out the p values and values of the test statistics for the implemented testing procedures. Additionally, the functions `summary.fanovatests`, `summary.fmanovaptbfr` and `summary.fmanovatrp` print out information about the data and parameters of the methods.

When calling the functions of the **fdANOVA** package, the software will check for presence of the **doBy**, **doParallel**, **ggplot2**, **fda**, **foreach**, **magic**, **MASS** and **parallel** packages if necessary (Hojsgaard and Halekoh 2016; Revolution Analytics and Weston 2015; Wickham 2009; Ramsay *et al.* 2014; Hankin 2005; Venables and Ripley 2002). If the required packages are not installed, an error message will be displayed.

It is worth to mention that fifteen labeled multivariate functional data sets are available in the **mfds** package (Górecki and Smaga 2017b), which is a kind of supplement to the **fdANOVA** package. Table 1 depicts brief information about these data sets, i.e., numbers of variables, design time points, groups and observations. The data sets were created from multivariate time series data available in Chen, Keogh, Hu, Begum, Bagnall, Mueen, and Batista (2015), Leeb, Lee, Keinrath, Scherer, Bischof, and Pfurtscheller (2007), Lichman (2013) and Olszewski (2001) by extending all variables to the same length as in Rodriguez, Alonso, and Maestro (2005). They originate from different domains, including handwriting recognition, medicine, robotics, etc. The data sets can be used for illustrating and evaluating practical efficiency of classification and statistical inference methods, etc. (see, for example, Górecki and Smaga 2015, 2017a). In Section 4, two of them are considered in experiments performed for evaluating efficiency of parallel implementation in the package.

3.3. Package demonstration on real data example

In this section, we provide examples that illustrate how the functions of the R package **fdANOVA** can be used to analyze real data. For this purpose, we use the popular *gait* data set available in the **fda** package. This data set consists of the angles formed by the hip and knee of each of 39 children over each child's gait cycle. The simultaneous variations of the hip and knee angles for children are observed at 20 equally spaced time points in $[0.025, 0.975]$.

Data sets	#Variables	#Time points	#Groups	#Observations
Arabic digits	13	93	10	8800
Australian language	22	136	95	2565
Character trajectories	3	205	20	2858
ECG	2	152	2	200
Graz	3	1152	2	140
Japanese vowels	12	29	9	640
Libras	2	45	15	360
Pen digits	2	8	10	10992
Robot failure LP1	6	15	4	88
Robot failure LP2	6	15	5	47
Robot failure LP3	6	15	4	47
Robot failure LP4	6	15	3	117
Robot failure LP5	6	15	5	164
uWaveGestureLibrary	3	315	8	4478
Wafer	6	198	2	1194

Table 1: Brief information about data sets available in the **mfds** package (Górecki and Smaga 2017b).

So, in this data set, we have two functional features, which we put in the list `x.gait` of length two, as presented below.

```
R> library("fda")
R> gait.data.frame <- as.data.frame(gait)
R> x.gait <- vector("list", 2)
R> x.gait[[1]] <- as.matrix(gait.data.frame[, 1:39])
R> x.gait[[2]] <- as.matrix(gait.data.frame[, 40:78])
```

Similarly to Górecki and Smaga (2017a), for illustrative purposes, the functional observations are divided into three samples. Namely, the first sample consists of the functions for the first 13 children, the second sample of the functions for the next 13 children, and the third sample of the functions for the remaining children. The sample labels are contained in the vector `group.label.gait`:

```
R> group.label.gait <- rep(1:3, each = 13)
```

We can plot the functional data by using the function `plotFANOVA`. For example, we plot the observations for the first functional feature without (Figure 1 (a)) and with indication of the samples (Figure 1 (b) and (c)) as well as the group mean functions (Figure 1 (d)).

```
R> library("fdANOVA")
R> plotFANOVA(x = x.gait[[1]], int = c(0.025, 0.975))
R> plotFANOVA(x = x.gait[[1]], group.label = as.character(group.label.gait),
+             int = c(0.025, 0.975))
R> plotFANOVA(x = x.gait[[1]], group.label = as.character(group.label.gait),
+             int = c(0.025, 0.975), separately = TRUE)
```

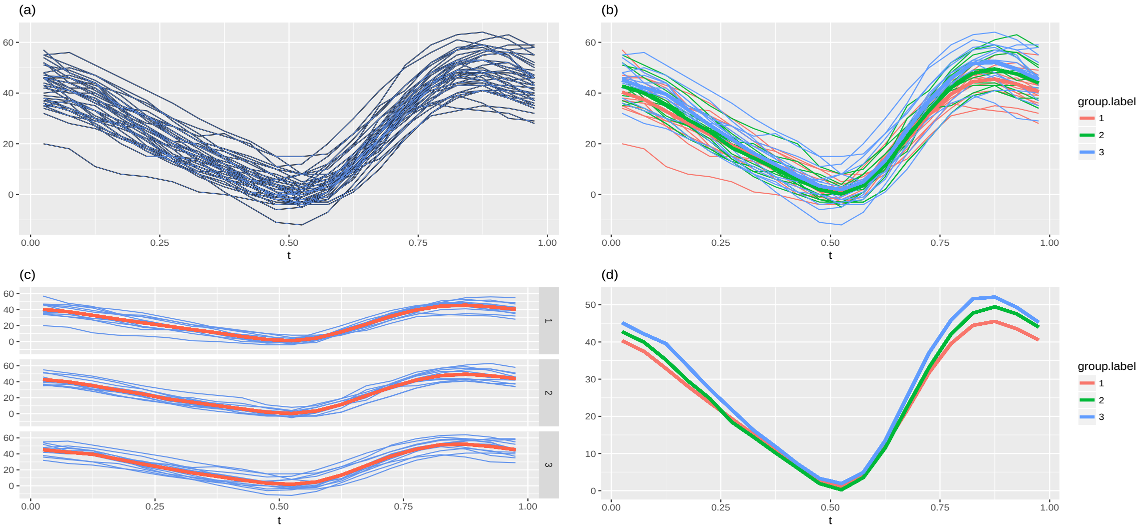

Figure 1: The first functional feature of the *gait* data without (Panel (a)) and with indication of the samples (Panels (b) and (c)). Panel (d) depicts the group mean functions.

```
R> plotFANOVA(x = x.gait[[1]], group.label = as.character(group.label.gait),
+             int = c(0.025, 0.975), means = TRUE)
```

From Figure 1, it seems that the mean functions of the three samples do not differ significantly. To confirm this statistically, we use the FANOVA tests implemented in the `fanova.tests` function. First, we use default values of the parameters of this function:

```
R> set.seed(123)
R> (fanova <- fanova.tests(x = x.gait[[1]], group.label = group.label.gait))

     Analysis of Variance for Functional Data

FP test - permutation test based on a basis function representation
Test statistic = 1.468218  p-value = 0.208

CH test - L2-norm-based parametric bootstrap test for homoscedastic samples
Test statistic = 7911.385  p-value = 0.2282

CS test - L2-norm-based parametric bootstrap test for heteroscedastic samples
Test statistic = 7911.385  p-value = 0.1969

L2N test - L2-norm-based test with naive method of estimation
Test statistic = 2637.128  p-value = 0.2106562

L2B test - L2-norm-based test with bias-reduced method of estimation
Test statistic = 2637.128  p-value = 0.1957646

L2b test - L2-norm-based bootstrap test
```

```
Test statistic = 2637.128  p-value = 0.2212

FN test - F-type test with naive method of estimation
Test statistic = 1.46698  p-value = 0.2226683

FB test - F-type test with bias-reduced method of estimation
Test statistic = 1.46698  p-value = 0.2198691

Fb test - F-type bootstrap test
Test statistic = 1.46698  p-value = 0.2705

GPF test - globalizing the pointwise F-test
Test statistic = 1.363179  p-value = 0.2691363

Fmaxb test - Fmax bootstrap test
Test statistic = 3.752671  p-value = 0.1887

TRP - tests based on k = 30 random projections
p-value ANOVA = 0.4082394
p-value ATS = 0.2994233
p-value WTPS = 0.3465
```

Besides of the p values displayed above, the list of matrices of projections of the data may be of practical interest for the test based on random projections users. The reason for this is that we can check the assumptions of the tests used in step 3 of the procedure based on random projections (see Section 2.1), e.g., the normality assumptions of ANOVA F test. Such inspection may result in choosing the appropriate test used in this step. This is especially important when the tests based on random projections differ in their decisions.

```
R> fanova$TRP$data.projections

[[1]]
          [,1]      [,2]      [,3]       [,4] ...    [,30]
[1,]  37.44057 -26.51193  17.28588  11.323063 ... 1.986230
[2,]  45.50439 -28.20233  21.76841   8.375763 ... 4.213405
[3,]  47.81353 -26.38955  22.49703  10.449332 ... 7.311809
...
[39,] 60.13816 -35.42660  27.27788  14.054984 ... 1.997130
```

As expected, neither FANOVA test rejects the null hypothesis. Now, we show how particular tests can be chosen and how the parameters of these tests can be changed. For the FP test, we use the predefined basis function representation of the data. For this purpose, we expand the data in the b-spline basis by using the functions from the **fda** package. They return the coefficients of expansion as well as the cross product matrix corresponding to the basis functions. For control, we choose the GPF test, which does not need any additional parameters. The Fmaxb test is performed by 1000 bootstrap samples. For the tests based on random projections, 10 and 15 projections are generated by using the Brownian motion, and p values are computed by the permutation method.

```
R> fbasis <- create.bspline.basis(rangeval = c(0.025, 0.975), 19, norder = 4)
R> own.basis <- Data2fd(seq(0.025, 0.975, len = 20), x.gait[[1]], fbasis)$coefs
R> own.cross.prod.mat <- inprod(fbasis, fbasis)
R> set.seed(123)
R> fanova.tests(x.gait[[1]], group.label.gait,
+              test = c("FP", "GPF", "Fmaxb", "TRP"),
+              params = list(paramFP = list(B.FP = 1000, basis = "own",
+                                           own.basis = own.basis,
+                                           own.cross.prod.mat =
+                                             own.cross.prod.mat),
+                            paramFmaxb = 1000,
+                            paramTRP = list(k = c(10, 15),
+                                            projection = "BM",
+                                            permutation = TRUE,
+                                            B.TRP = 1000)))

     Analysis of Variance for Functional Data

FP test - permutation test based on a basis function representation
Test statistic = 1.468105  p-value = 0.199

GPF test - globalizing the pointwise F-test
Test statistic = 1.363179  p-value = 0.2691363

Fmaxb test - Fmax bootstrap test
Test statistic = 3.752671  p-value = 0.17

TRP - tests based on k = 10 random projections
p-value ANOVA = 0.26
p-value ATS = 0.259
p-value WTPS = 0.329

TRP - tests based on k = 15 random projections
p-value ANOVA = 0.252
p-value ATS = 0.249
p-value WTPS = 0.339
```

The above examples concern only the first functional feature of the *gait* data set. Similar analysis can be performed for the second one. However, both features can be simultaneously investigated by using the FMANOVA tests desribed in Section 2.2. First, we consider the permutation tests based on a basis function representation implemented in the function `fmanova.ptbfr`. We apply this function to the whole data set specifying non-default values of most of parameters. Here, we also show how use the function `summary.fmanovaptbfr` to additionally obtain a summary of the data and test parameters. Observe that the results are consistent with these obtained by FANOVA tests.

```
R> set.seed(123)
R> fmanova <- fmanova.ptbfr(x.gait, group.label.gait, int = c(0.025, 0.975),
+                            B = 5000, basis = "b-spline", criterion = "eBIC",
+                            method = "mean", minK = 5, maxK = 20, norder = 4,
+                            gamma.eBIC = 0.7)
R> summary(fmanova)

 FMANOVA - Permutation Tests based on a Basis Function Representation

 Data summary

Number of observations = 39
Number of features = 2
Number of time points = 20
Number of groups = 3
Group labels: 1 2 3
Group sizes: 13 13 13
Range of data = [0.025 , 0.975]

 Testing results

W  = 0.9077424   p-value = 0.5294
LH = 0.1003732   p-value = 0.524
P  = 0.09340229  p-value = 0.5334
R  = 0.08565056  p-value = 0.388

 Parameters of test

Number of permutations = 5000
Basis: b-spline (norder = 4)
Criterion: eBIC (gamma.eBIC = 0.7)
Method: mean
Km = 20 20 KM = 20 minK = 5 maxK = 20
```

Finally, we apply the tests based on random projections for the FMANOVA problem in the *gait* data set. In the following, these tests are performed with $k = 1, 5, 10, 15, 20$ projections as well as standard and permutation methods. The resulting p values are visualized by the `plot.fmanovatrp` function:

```
R> set.seed(123)
R> fmanova1 <- fmanova.trp(x.gait, group.label.gait, k = c(1, 5, 10, 15, 20))
R> fmanova2 <- fmanova.trp(x.gait, group.label.gait, k = c(1, 5, 10, 15, 20),
+                           permutation = TRUE)
R> plot(x = fmanova1, y = fmanova2)
R> plot(x = fmanova1, y = fmanova2, withoutRoy = TRUE)
```

The obtained plots are shown in Figure 2. As we can observe, except the standard Rp test, all testing procedures behave similarly and do not reject the null hypothesis. The standard Rp test does not keep the pre-assigned type-I error rate as Górecki and Smaga (2017a) shown

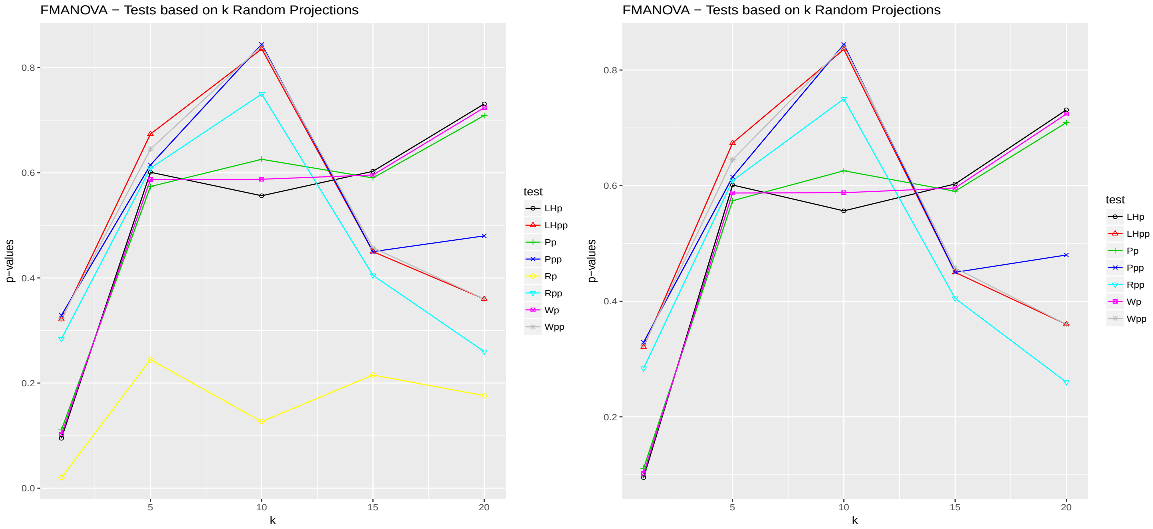

Figure 2: P values of the tests based on random projections for the FMANOVA problem in the *gait* data set. The Wpp, LHpp, Ppp and Rpp tests are the permutation versions of the Wp, LHp, Pp and Rp tests.

in simulations. More precisely, this test is usually too liberal, which explains that its p values are much smaller than these of the other testing procedures. That is why the function has option not to plot the p values of this test.

4. Efficiency of the parallel implementation

As we mentioned in Section 3.2, the functions of the **fdANOVA** package provide the option of parallelization of the execution of the most time consuming steps of performing the testing procedures. By default the parameter `parallel` is set to `FALSE`, which corresponds to sequential version. This option should be used when the data set is not too large or we have a single processor machine with one core, since the parallelization is very inefficient in such cases. When it is possible or needed, we specify `parallel = TRUE` to perform parallel computations using process forking based on the **doParallel** package (Revolution Analytics and Weston 2015). We also have possibility of controlling a number of parallel tasks by setting the parameter `nslaves`. This parameter is set to a number of logical processes of used computer by default, which is an optimal choice. This parallel method can be performed on a single logical machine only. Further information about parallel computing in R and its usage can be found for example in Schmidberger, Morgan, Eddelbuettel, Yu, Tierney, and Mansmann (2009) and Teisseyre, Kłopotek, and Mielniczuk (2016).

In the remainder of this section, we discuss the results of experiments evaluating efficiency of parallel implementation in the **fdANOVA** package. They were performed for one physical computer with 14 cores (processor Intel(R) Xeon(R) E5-2690 v4 @ 2.6 GHz, 3.5 GHz all-core turbo), 64 GB RAM, Windows 7 64-bit, R 3.3.3. We compared the execution time and speedup of the sequential and parallel versions. The number of slaves was set to $2, 4, 8$ and 14. The experiments were performed by using the *Graz* and *uWaveGestureLibrary* data sets available in the **mfds** package (Górecki and Smaga 2017b). We present and discuss the results

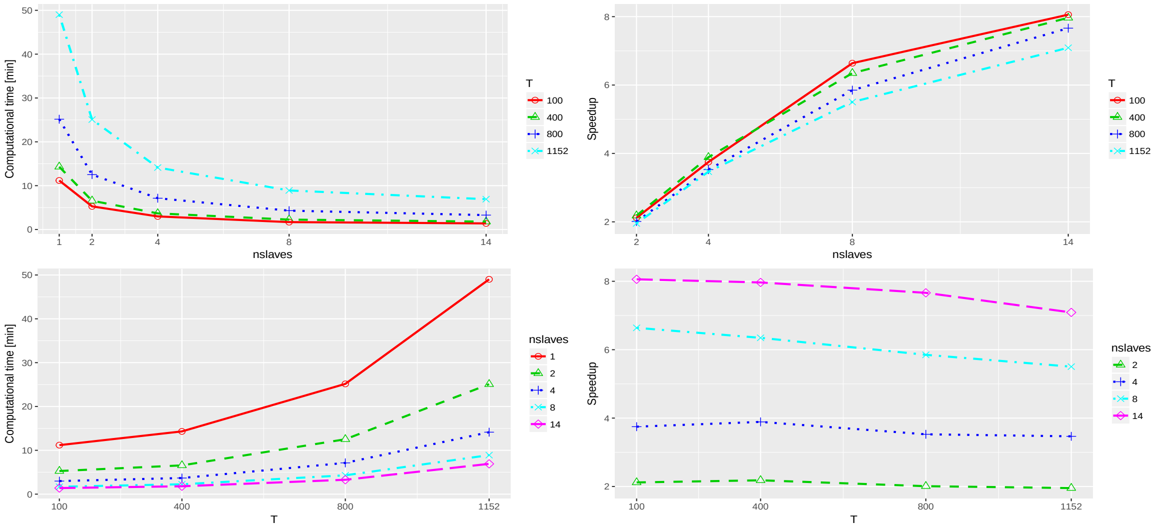

Figure 3: Execution time and speedup versus the number of slaves and the number of design time points $\mathcal{T}$ obtained for the *Graz* data set (One slave refers to the sequential version.).

of the experiments based on these data sets and the FANOVA tests implemented via parallel method in the function `fanova.tests`, i.e., the FP, CH, CS, L^2b, Fb, Fmaxb tests and the permutation test based on random projections. These tests were applied to the first features of these data sets by using default values of parameters, i.e., $1000, 10{,}000$ permutation replicates for the FP test and the tests based on random projections, respectively, 10,000 bootstrap samples and 30 projections. In the other settings, similar results have been obtained, and therefore, they are omitted for space saving.

In the first experiment, the first feature of the *Graz* data set was considered. This data set contains $n = 140$ functional observations, which were measured in $\mathcal{T} = 1152$ design time points and divided into two groups each of 70 observations. In this experiment, we studied how the computational time and speedup depend on the number of design time points, i.e., we applied the tests to the truncated *Graz* data sets with $\mathcal{T} = 100, 400, 800, 1152$ design time points. The second experiment is based on the first feature of the *uWaveGestureLibrary* data set, which contains $n = 4478$ functional observations measured in $\mathcal{T} = 315$ design time points and divided into eight samples of 559 or 560 observations. In this case, we investigated how the computational time and speedup depend on the number of observations. For this purpose, we considered the subsets of the *uWaveGestureLibrary* data set consisting of $n = 100, 400, 700, 1000$ observations. Figures 3 and 4 depict the results of both experiments.

Observe that the parallel implementation results in satisfactory shortening the computational time. The usage of two slaves has already made calculations much faster. The execution time strictly decreases with increase of the number of slaves. The behavior of speedup also seems to be satisfactory, although it is not linear as there are other sequentially executed tasks in the functions. Nevertheless, for two and four slaves, the speedup is almost linear in both experiments, i.e., it is close to the numbers of parallel tasks. From the first experiment, however, it follows that the speedup for greater number of slaves may decrease with increase of the number of design time points (Figure 3). Fortunately, the opposite usually holds when the number of observations increases as in the second experiment (Figure 4). This can perhaps

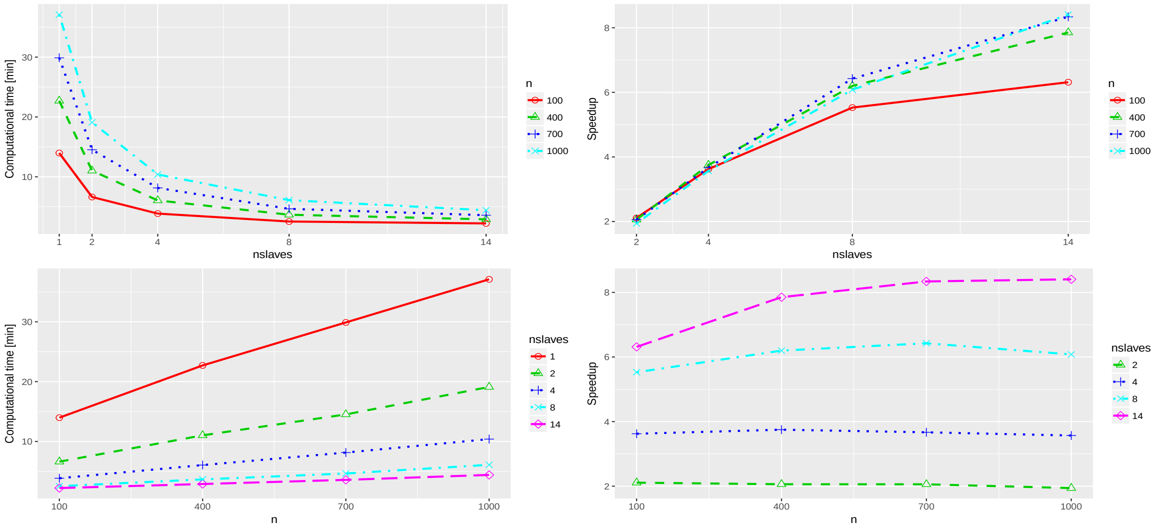

Figure 4: Execution time and speedup versus the number of slaves and the number of observations n obtained for the *uWaveGestureLibrary* data set (One slave refers to the sequential version.).

be explained by that the parallel tasks are more concerned in functional observations than their values in particular design time points. The implementation of the permutation tests based on random projections seems to be the most time-consuming. When one wants to perform this tests, it is recommended to use the parallel method even for not very large data sets. However, when we only choose other testing procedures for analysis, the parallelization should be used for larger data sets. Otherwise it may not make sense. Summarizing, the parallel implementation enables to reduce the computation time significantly, when it is used appropriately.

5. Simulation study

In this section, we consider simulation studies to compare the GPF, Fmaxb tests and the tests based on random projections. For the last one, two methods of generating Gaussian processes implemented in the **fdANOVA** package are considered. As a result, simulations indicate the recommended tests for different scenarios.

The simulations scenarios were inspired by a Monte Carlo study in Zhang and Liang (2014) and Zhang *et al.* (2016).

5.1. Simulation for FMANOVA tests

To compare the FMANOVA tests based on random projections with different methods of generating Gaussian processes, we consider three groups of two-dimensional functional data and two vectors $\mathbf{n} = (n_1, n_2, n_3)$ of sample sizes $\mathbf{n}_1 = (10, 10, 10)$ and $\mathbf{n}_2 = (10, 20, 15)$. Discrete functional samples are generated by using the model $x_{prs}(t) = \eta_{pr}(t) + \upsilon_{prs}(t)$, $t \in [0, 1]$, $p = 1, 2, r = 1, 2, 3$, $s = 1, \ldots, n_r$. The mean group functions are as follows: $\eta_{1r}(t) = \mathbf{c}_1^\top(1, t, t^2, t^3)^\top$ for $r = 1, 2$, $\eta_{13}(t) = (\mathbf{c}_1 + 2\delta\mathbf{u})^\top(1, t, t^2, t^3)^\top$ and $\eta_{2r}(t) = \mathbf{c}_2^\top(1, t, t^2, t^3)^\top$

for $r = 1, 2, 3$, where $\mathbf{c}_1 = (1, 2.3, 3.4, 1.5)^\top$, $\mathbf{c}_2 = (2.2, 1.6, 1, 2.9)^\top$. Thus, the differences between appropriate group mean functions is controlled by δ, whose values are contained in the tables of simulation results. Moreover, the direction of these differences is specified by $\mathbf{u} = (1, 2, 3, 4)^\top/\sqrt{30}$. For i.i.d. $v_{prs}(t) \sim SP(0, \gamma)$, we take $v_{prs}(t) = \mathbf{b}_{prs}^\top \boldsymbol{\Psi}(t)$, $\mathbf{b}_{prs} = (b_{prs,1}, \dots, b_{prs,11})^\top$, $b_{prs,w} \overset{d}{=} \sqrt{\lambda_w} y_{prs,w}, w = 1, \dots, 11$, where $\boldsymbol{\Psi}(t) = (\psi_1(t), \dots, \psi_{11}(t))^\top$ is the orthonormal basis vector such that $\psi_1(t) = 1$, $\psi_{2\omega}(t) = \sqrt{2}\sin(2\pi\omega t)$ and $\psi_{2\omega+1}(t) = \sqrt{2}\cos(2\pi\omega t)$, $\omega = 1, \dots, 5$, $\lambda_w = 1.5\rho^w$, $\rho = 0.1, 0.3, 0.5, 0.7, 0.9$, and $y_{prs,w}$ are independent random variables, $\mathrm{E}(y_{prs,w}) = 0$, $\mathrm{Var}(y_{prs,w}) = 1$. The Gaussian (resp. non-Gaussian) functional observations are obtained by considering $y_{prs,w} \sim N(0, 1)$ (resp. $y_{prs,w} \sim t_4/\sqrt{2}$). Notice that $\gamma(s, t)$ is of the form $\boldsymbol{\Psi}(s)^\top \mathrm{diag}(\lambda_1, \dots, \lambda_{11})\boldsymbol{\Psi}(t)$ and the correlation increases with decreasing ρ. The functions $x_{prs}(t)$ are assumed to be observed at the design time points $q/51$, $q = 1, \dots, 50$. For $\delta = 0$ (resp. $\delta \neq 0$), the null hypothesis is true (resp. false) and we investigate the size control (resp. power) of the tests. This model will be referred to as FMANOVA model, and the simulation results obtained under it are depicted in Tables 2 and 3 in the Appendix A.

All testing procedures based on random projections for FMANOVA except the Rp test control the nominal type-I error rate. They show a tendency of conservativity in most scenarios, but this is natural for tests based on random projection as indicated by Cuesta-Albertos and Febrero-Bande (2010) and Górecki and Smaga (2017a). The Pp (resp. LHp) test has the most conservative (resp. liberal) character. The standard version of the Rp test tends to be highly liberal for both methods of generating Gaussian processes. So, it is not comparable with the other tests. The permutation versions of all tests seem to behave better than the standard ones when the null hypothesis holds true, especially in the case of t distribution.

Since the standard Rp tests do not maintain the pre-assigned type-I error rate, we do not compare their empirical powers with those of the other testing procedures. For completeness, however, they are presented in tables containing the simulation results. With increasing sample sizes, the empirical powers of the tests generally increase. Usually the LHp tests are more powerful than the Wp tests, which outperform the Pp ones. The permutation version of the LHp (resp. Wp or Pp) test has generally slightly smaller (resp. greater) empirical powers than the standard one. In terms of power, the permutation Rp testing procedure behaves similarly to the permutation Wp and LHp tests. Moreover, observe that the tests based on random projections using Brownian motion have higher (resp. lower) empirical powers than those using Gaussian white noise when the functional data are less (resp. highly) correlated, i.e., $\rho = 0.7, 0.9$ (resp. $\rho = 0.1, 0.3$). For moderately correlated functional data ($\rho = 0.5$), both methods of generating Gaussian processes behave very similar.

5.2. Simulation for FANOVA tests

The comparison of the GPF, Fmaxb tests and the tests based on random projections with two methods of generating Gaussian processes is made by generating three discrete functional samples by using the model $x_{rs}(t) = \eta_r(t) + v_{rs}(t)$, $t \in [0, 1]$, $r = 1, 2, 3$, $s = 1, \dots, n_r$, where the sample sizes are as in FMANOVA model and subject-effect functions $v_{rs}(t)$ are defined in the analogous way as in that model. We consider $\eta_r(t) = \mathbf{c}_1^\top(1, t, t^2, t^3)^\top$ for $r = 1, 2$ and $\eta_3(t) = (\mathbf{c}_1 + 2\delta\mathbf{u})^\top(1, t, t^2, t^3)^\top$, where $\mathbf{c}_1$, δ and $\mathbf{u}$ are as in FMANOVA model. This model will be referred to as FANOVA model. The simulation results are given in Table 4 in the Appendix A.

The GPF and Fmaxb tests keep the pre-assigned type-I error rate quite well, but the first one may be slightly liberal. The empirical sizes of the tests based on random projections are generally much lower than the nominal significance level. The tests based on ANOVA F test statistic are usually more conservative than the other testing procedures based on random projections. Observe also that the tests based on random projections and using ATS and WTPS and Gaussian white noise may be slightly liberal for less correlated functional data.

The conclusions about the empirical power of the tests based on random projections under two different methods of generating Gaussian processes are the same as in the multivariate case (see the last paragraph of Section 5.1). For very highly correlated functional data ($\rho = 0.1$), the tests based on random projections using Gaussian white noise are the most powerful. When the functional data are a little less highly or moderately correlated ($\rho = 0.3, 0.5$), in terms of power, the best testing procedure is the Fmaxb method. Finally, for less correlated functional data ($\rho = 0.7, 0.9$), the GPF test and the tests based on random projections using Brownian motion outperform the remaining ones. Thus, there is not one method, which performs best, and the performance of the tests depends on the amount of correlation in functional data. Moreover, the tests based on random projections using different ANOVA methods do not perform equally well. It turns out that the tests based on random projections using ANOVA F test statistic and ATS have the highest power for normally distributed data, perfoming slightly better than those based on WTPS. The reason for this is that these testing procedures are constructed under normality assumption, while the WTPS one is not. The situation changes in case of t distribution. There the tests based on random projections using WTPS are usually the best ones.

6. Conclusions and future work

Functional data analysis offers tools for solving statistical problems for high-dimensional data considered as curves or functions. An R package **fdANOVA** implements a broad range of the analysis of variance testing procedures for univariate and multivariate functional data. The implemented tests are usually very different solutions to the FANOVA and FMANOVA problems. Since their performance may depend on specifics of the functional observations, the package gives the opportunity to choose the most appropriate test for specific real data. The article presents the empirical evaluation of size control and power of some tests that were not compared elsewhere, which, together with previous simulation results (see, for example, Górecki and Smaga 2015, 2017a), may help in choosing appropriate procedures in practice. All the time consuming parts of the package are parallelized, which makes it relatively fast, given that most of the testing procedures are based on permutation, projection and resampling methods.

When the testing procedures implemented in the **fdANOVA** package reject the null hypothesis about equality of group mean functions, it would be of interest to check which mean functions are significantly different and which are not. So, further developments of the package will include the implementation of post hoc and constrast analysis for functional data. Such testing procedures are considered, for instance, by Zhang (2013), who proposed the L^2-norm-based and F-type tests.

Appendix A - Simulation results

The results of simulations described and discussed in Section 5 are depicted in Tables 2–4.

References

Anderson TW (2003). *An Introduction to Multivariate Statistical Analysis.* 3rd edition. John Wiley & Sons, London.

Benjamini Y, Hochberg Y (1995). "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society B*, **57**, 289–300.

Benjamini Y, Yekutieli D (2001). "The Control of the False Discovery Rate in Multiple Testing under Dependency." *The Annals of Statistics*, **29**, 1165–1188.

Berrendero JR, Justel A, Svarc M (2011). "Principal Components for Multivariate Functional Data." *Computational Statistics & Data Analysis*, **55**, 2619–2634.

Bobelyn E, Serban AS, Nicu M, Lammertyn J, Nicolai BM, Saeys W (2010). "Postharvest Quality of Apple Predicted by NIR-spectroscopy: Study of the Effect of Biological Variability on Spectra and Model Performance." *Postharvest Biology and Technology*, **55**, 133–143.

Boente G, Barrera MS, Tyler DE (2014). "A Characterization of Elliptical Distributions and Some Optimality Properties of Principal Components for Functional Data." *Journal of Multivariate Analysis*, **131**, 254–264.

Brockhaus S, Rügamer D (2016). ***FDboost**: Boosting Functional Regression Models.* R package version 0.2-0, URL https://CRAN.R-project.org/package=FDboost.

Brockhaus S, Scheipl F, Hothorn T, Greven S (2015). "The Functional Linear Array Model." *Statistical Modelling*, **15**, 279–300.

Brunner E, Dette H, Munk A (1997). "Box-Type Approximations in Nonparametric Factorial Designs." *Journal of the American Statistical Association*, **92**, 1494–1502.

Cao G, Yang L, Todem D (2012). "Simultaneous Inference for the Mean Function Based on Dense Functional Data." *Journal of Nonparametric Statistics*, **24**, 359–377.

Chang C, Chen Y, Ogden RT (2014). "Functional Data Classification: A Wavelet Approach." *Computational Statistics*, **29**, 1497–1513.

Chen D, Hall P, Müller HG (2011). "Single and Multiple Index Functional Regression Models with Nonparametric Link." *The Annals of Statistics*, **39**, 1720–1747.

Chen Y, Keogh E, Hu B, Begum N, Bagnall A, Mueen A, Batista G (2015). "The UCR Time Series Classification Archive." URL http://www.cs.ucr.edu/~eamonn/time_series_data/.

Cheng Y, Shi JQ (2016). ***flars**: Functional LARS.* R package version 1.0, URL https://CRAN.R-project.org/package=flars.

				GAUSS				BM			
Distribution	**n**	ρ	V	Wp	LHp	Pp	Rp	Wp	LHp	Pp	Rp
$N(0,1)$	$\mathbf{n}_1$	0.1	s	2.6	3.1	1.8	12.0	3.2	3.5	2.7	12.0
			p	3.0	2.5	2.7	2.3	3.1	3.2	3.1	3.4
		0.3	s	2.5	4.0	1.5	13.4	2.8	3.1	2.5	11.4
			p	3.7	3.6	3.8	3.1	2.7	2.6	3.2	2.3
		0.5	s	3.1	5.1	1.7	16.5	3.0	3.2	2.2	11.4
			p	4.3	3.8	4.7	3.7	3.1	2.6	3.3	2.5
		0.7	s	4.1	6.1	2.6	18.5	2.7	4.0	2.1	11.5
			p	4.9	4.3	5.6	4.9	2.8	2.8	3.1	2.7
		0.9	s	4.6	7.2	2.7	21.4	3.5	4.2	2.0	11.3
			p	6.3	5.9	5.9	6.3	3.2	2.7	3.3	2.5
	$\mathbf{n}_2$	0.1	s	1.9	2.2	1.6	9.2	2.4	2.4	2.2	11.2
			p	2.1	2.0	2.5	2.1	2.5	2.4	2.3	2.3
		0.3	s	2.2	2.9	1.7	11.9	2.4	2.6	2.0	11.7
			p	3.5	3.4	3.3	3.9	2.7	2.4	2.8	2.5
		0.5	s	3.4	3.9	2.8	14.4	2.7	3.1	1.9	11.1
			p	4.6	4.5	4.6	4.8	2.9	3.0	3.2	2.7
		0.7	s	3.6	5.3	3.0	16.2	2.9	3.6	1.9	11.4
			p	6.2	6.0	6.0	5.5	3.3	3.4	3.2	2.9
		0.9	s	4.4	5.9	2.8	20.3	2.9	4.1	2.2	13.5
			p	6.1	6.3	6.3	6.9	3.6	3.7	3.9	3.6
$t_4/\sqrt{2}$	$\mathbf{n}_1$	0.1	s	2.9	3.5	2.2	10.3	2.5	3.1	1.9	9.2
			p	3.7	3.9	3.7	3.8	3.3	3.0	3.1	3.3
		0.3	s	3.5	4.2	2.4	12.3	2.4	2.8	1.7	9.9
			p	4.6	4.2	5.6	3.7	3.1	3.2	3.3	3.1
		0.5	s	3.7	4.9	2.5	14.9	2.6	3.1	1.9	9.3
			p	5.8	5.8	5.8	4.5	3.5	3.6	3.4	3.6
		0.7	s	3.8	5.8	1.9	16.4	2.9	3.5	2.1	9.7
			p	6.1	6.6	6.7	6.2	3.6	3.9	4.0	3.8
		0.9	s	3.2	5.9	2.0	18.1	3.0	4.2	2.3	11.0
			p	5.3	5.6	5.8	6.4	4.1	4.6	3.9	5.0
	$\mathbf{n}_2$	0.1	s	3.1	3.6	2.3	11.6	3.2	3.5	2.8	11.6
			p	3.7	3.9	3.9	4.0	3.7	3.8	3.7	3.7
		0.3	s	2.7	4.1	2.0	12.5	3.2	3.8	2.8	10.5
			p	4.0	4.4	4.4	4.4	3.7	4.0	3.9	4.0
		0.5	s	3.3	4.5	2.4	14.8	3.4	4.1	2.8	10.8
			p	4.0	3.7	4.3	4.2	4.1	4.1	4.6	4.1
		0.7	s	4.5	5.6	3.2	16.9	3.4	4.5	2.5	11.9
			p	4.8	4.8	4.7	5.0	4.8	4.7	4.7	4.1
		0.9	s	5.0	6.1	3.3	18.2	4.2	4.8	3.3	13.1
			p	5.3	5.0	4.7	5.7	5.0	5.1	5.0	4.7

Table 2: Empirical sizes (as percentages) of the tests based on $k = 30$ random projections obtained under FMANOVA model versus different methods of generating Gaussian processes (GAUSS and BM). In the column “V”, “s” and “p” refer to the standard and permutation versions of the tests, respectively. Moreover, $\delta = 0$, $M = 50$, $\mathbf{n}_1 = (10, 10, 10)$, $\mathbf{n}_2 = (10, 20, 15)$, $nr = 1000$, $nperm = 1000$ and $\alpha = 5\%$.

					GAUSS				BM			
Distribution	$\mathbf{n}$	ρ	δ	V	Wp	LHp	Pp	Rp	Wp	LHp	Pp	Rp
$N(0,1)$	$\mathbf{n}_1$	0.1	0.12	s	53.2	58.3	43.6	74.4	22.9	27.8	18.5	46.6
				p	54.1	54.8	51.8	55.0	24.4	24.8	23.6	25.6
		0.3	0.30	s	43.2	50.8	33.9	70.0	27.4	32.6	23.2	55.6
				p	45.3	46.2	42.4	45.9	29.7	29.8	29.0	30.1
		0.5	0.45	s	33.5	39.6	22.8	62.2	31.2	34.5	25.7	58.9
				p	35.1	36.7	34.0	36.0	32.6	32.9	32.0	33.2
		0.7	0.80	s	51.9	60.0	39.4	80.2	62.0	67.1	55.6	86.1
				p	53.0	52.8	50.8	52.5	64.6	65.6	63.1	67.1
		0.9	1.20	s	56.4	65.2	41.0	86.1	87.2	89.5	82.2	96.6
				p	55.7	56.7	53.9	57.2	88.5	89.1	86.8	88.8
	$\mathbf{n}_2$	0.1	0.12	s	77.9	80.5	72.0	91.2	42.6	47.7	37.5	65.7
				p	78.6	79.1	77.5	79.1	45.1	45.9	43.9	45.1
		0.3	0.30	s	68.3	72.5	63.6	89.3	49.7	53.2	44.0	71.0
				p	70.4	71.0	68.8	70.8	48.9	49.6	48.4	50.1
		0.5	0.45	s	54.3	58.8	48.4	81.6	51.3	54.5	47.2	75.9
				p	58.5	59.3	57.9	59.1	51.9	52.7	52.1	53.6
		0.7	0.80	s	79.0	83.8	73.5	95.4	88.8	90.7	86.3	97.2
				p	81.2	81.9	80.3	81.2	89.6	89.4	88.9	90.0
		0.9	1.20	s	83.1	87.1	77.4	96.9	98.5	98.8	98.1	99.5
				p	83.7	84.1	82.6	85.5	98.6	98.7	98.4	98.5
$t_4/\sqrt{2}$	$\mathbf{n}_1$	0.1	0.12	s	57.2	62.9	47.9	77.7	26.7	31.4	20.1	48.8
				p	61.4	61.2	58.0	62.0	28.5	29.7	26.9	30.4
		0.3	0.30	s	45.8	51.5	34.2	72.4	30.9	35.7	25.0	54.6
				p	51.7	51.6	48.8	52.0	35.1	35.2	33.9	35.4
		0.5	0.45	s	33.5	40.5	24.9	65.5	34.1	38.0	28.5	59.5
				p	41.6	42.2	39.4	41.3	38.1	38.0	37.6	38.9
		0.7	0.80	s	57.6	64.6	45.1	83.5	68.0	72.3	60.0	86.6
				p	61.4	62.0	58.7	62.4	69.8	70.8	69.0	70.3
		0.9	1.20	s	61.4	68.7	47.5	86.5	88.3	90.5	83.2	96.8
				p	63.8	65.3	61.4	65.4	89.5	89.8	88.3	89.5
	$\mathbf{n}_2$	0.1	0.12	s	79.1	82.3	73.8	90.3	46.8	49.6	41.2	67.9
				p	81.0	81.7	80.2	82.3	48.0	48.8	46.5	49.4
		0.3	0.30	s	72.8	77.4	67.5	88.5	54.0	58.0	49.7	74.9
				p	76.5	76.5	75.6	76.8	56.7	57.2	56.5	56.4
		0.5	0.45	s	59.1	63.6	52.4	82.9	57.8	60.6	54.2	78.2
				p	63.3	64.3	62.3	66.3	61.1	61.0	60.1	59.5
		0.7	0.80	s	80.1	84.6	75.9	94.4	89.1	90.6	86.8	97.1
				p	83.1	83.4	82.5	83.4	90.4	90.4	90.2	90.5
		0.9	1.20	s	84.3	87.8	78.8	96.7	98.7	99.1	98.3	99.5
				p	86.5	86.6	85.7	86.9	98.8	98.8	98.4	98.7

Table 3: Empirical powers (as percentages) of the tests based on $k = 30$ random projections obtained under FMANOVA model versus different methods of generating Gaussian processes (GAUSS and BM). In the column "V", "s" and "p" refer to the standard and permutation versions of the tests, respectively. Moreover, $\delta \neq 0$, $M = 50$, $\mathbf{n}_1 = (10, 10, 10)$, $\mathbf{n}_2 = (10, 20, 15)$, $nr = 1000$, $nperm = 1000$ and $\alpha = 5\%$.

						GAUSS			BM		
Distr.	$\mathbf{n}$	ρ	δ	GPF	Fmaxb	ANOVA	ATS	WTPS	ANOVA	ATS	WTPS
$N(0,1)$	$\mathbf{n}_1$	0.1	0	6.3	5.2	2.7	3.3	3.2	3.4	4.5	3.4
		0.3	0	7.1	5.1	2.4	3.1	3.6	3.0	3.8	3.0
		0.5	0	7.6	5.2	2.9	3.6	4.5	3.2	3.6	3.0
		0.7	0	6.0	5.2	3.7	4.9	4.9	2.7	3.0	3.7
		0.9	0	4.8	5.8	5.1	6.2	5.6	3.0	3.8	3.4
		0.1	0.12	21.6	50.8	67.0	69.9	62.5	35.2	38.0	33.3
		0.3	0.30	42.2	71.3	55.6	60.1	52.2	39.7	43.7	36.6
		0.5	0.45	53.0	64.0	43.3	47.1	39.7	43.1	46.4	40.3
		0.7	0.80	81.6	79.2	64.3	69.0	62.0	77.0	79.9	72.7
		0.9	1.20	89.0	72.7	69.4	73.7	64.6	94.1	95.2	92.4
	$\mathbf{n}_2$	0.1	0	5.8	4.9	3.5	4.6	3.5	3.4	4.5	3.2
		0.3	0	6.3	5.3	4.3	4.7	3.6	3.1	4.4	3.0
		0.5	0	7.1	4.8	5.1	5.3	4.0	3.5	4.3	3.3
		0.7	0	6.5	5.5	4.2	6.8	6.4	2.7	3.4	3.7
		0.9	0	5.9	6.7	4.7	6.9	4.9	4.5	5.2	3.6
		0.1	0.12	31.0	75.0	85.3	82.8	82.3	57.0	53.5	52.1
		0.3	0.30	59.2	93.4	80.8	77.5	75.7	60.4	59.3	58.2
		0.5	0.45	71.3	89.0	69.8	66.5	63.7	65.2	62.8	61.7
		0.7	0.80	95.2	96.0	87.5	84.8	84.0	94.1	93.1	92.3
		0.9	1.20	98.9	93.4	91.9	89.2	88.1	99.2	98.9	98.8
$\frac{t_4}{\sqrt{2}}$	$\mathbf{n}_1$	0.1	0	5.7	3.7	2.1	2.1	2.6	2.6	2.6	3.3
		0.3	0	5.4	3.9	1.8	2.0	3.7	1.9	2.5	3.3
		0.5	0	4.7	3.7	2.0	2.2	3.2	1.7	2.0	2.7
		0.7	0	4.3	3.5	2.3	3.4	4.2	2.0	2.6	2.9
		0.9	0	3.4	3.6	3.5	4.6	5.3	2.0	2.6	2.8
		0.1	0.12	25.6	58.9	72.9	73.5	74.5	40.3	42.1	42.1
		0.3	0.30	45.3	74.6	58.9	61.3	60.9	43.0	44.0	46.1
		0.5	0.45	54.2	68.7	48.9	50.6	51.1	46.0	46.9	49.1
		0.7	0.80	82.6	80.5	69.8	73.2	70.0	80.2	81.2	81.5
		0.9	1.20	87.8	74.5	73.7	74.1	73.2	93.2	93.0	93.8
	$\mathbf{n}_2$	0.1	0	6.5	5.5	2.5	2.6	3.5	3.1	3.4	2.9
		0.3	0	6.2	5.8	2.5	2.9	3.6	2.5	2.7	2.6
		0.5	0	6.5	5.3	3.8	4.3	4.0	2.6	2.7	3.1
		0.7	0	6.3	5.0	3.7	5.8	4.9	2.4	2.9	3.3
		0.9	0	5.9	5.9	4.5	6.1	5.3	2.2	3.1	3.6
		0.1	0.12	34.6	76.0	86.7	83.3	85.7	58.5	52.0	57.9
		0.3	0.30	62.9	92.3	84.4	79.7	83.5	67.2	63.8	65.9
		0.5	0.45	72.9	89.6	73.6	69.9	73.6	69.9	66.8	71.2
		0.7	0.80	94.9	96.4	89.6	86.1	89.0	94.4	92.5	94.6
		0.9	1.20	98.1	94.7	91.6	89.1	90.6	99.2	98.0	99.1

Table 4: Empirical sizes ($\delta = 0$) and powers ($\delta \neq 0$), as percentages, of the GPF and Fmaxb tests and the tests based on $k = 30$ random projections with two methods of generating Gaussian processes (GAUSS and BM) obtained under FANOVA model. Moreover, $M = 50$, $\mathbf{n}_1 = (10, 10, 10)$, $\mathbf{n}_2 = (10, 20, 15)$, $nr = 1000$, $nboot = 10{,}000$, $nperm = 1000$ and $\alpha = 5\%$.

Chiou JM, Chen YT, Yang YF (2014). "Multivariate Functional Principal Component Analysis: A Normalization Approach." *Statistica Sinica*, **24**, 1571–1596.

Chiou JM, Müller HG (2014). "Linear Manifold Modelling of Multivariate Functional Data." *Journal of the Royal Statistical Society B*, **76**, 605–626.

Chiou JM, Yang YF, Chen YT (2016). "Multivariate Functional Linear Regression and Prediction." *Journal of Multivariate Analysis*, **146**, 301–312.

Coffey N, Hinde J, Holian E (2014). "Clustering Longitudinal Profiles Using P-Splines and Mixed Effects Models Applied to Time-Course Gene Expression Data." *Computational Statistics & Data Analysis*, **71**, 14–29.

Cuesta-Albertos JA, Febrero-Bande M (2010). "A Simple Multiway ANOVA for Functional Data." *Test*, **19**, 537–557.

Cuevas A (2014). "A Partial Overview of the Theory of Statistics with Functional Data." *Journal of Statistical Planning and Inference*, **147**, 1–23.

Cuevas A, Febrero M, Fraiman R (2004). "An Anova Test for Functional Data." *Computational Statistics & Data Analysis*, **47**, 111–122.

Dai X, Hadjipantelis PZ, Ji H, Müller HG, Wang JL (2016). ***fdapace**: Functional Data Analysis and Empirical Dynamics.* R package version 0.2.5, URL https://CRAN.R-project.org/package=fdapace.

Degras DA (2011). "Simultaneous Confidence Bands for Nonparametric Regression with Functional Data." *Statistica Sinica*, **21**, 1735–1765.

Delaigle A, Hall P (2012). "Achieving Near Perfect Classification for Functional Data." *Journal of the Royal Statistical Society B*, **74**, 267–286.

Delaigle A, Hall P (2013). "Classification Using Censored Functional Data." *Journal of the American Statistical Association*, **108**, 1269–1283.

Dziak J, Shiyko M (2016). ***funreg**: Functional Regression for Irregularly Timed Data.* R package version 1.2, URL https://CRAN.R-project.org/package=funreg.

Faraway J (1997). "Regression Analysis for a Functional Response." *Technometrics*, **39**, 254–261.

Febrero-Bande M, Galeano P, González-Manteiga W (2008). "Outlier Detection in Functional Data by Depth Measures, with Application to Identify Abnormal NO$_x$ Levels." *Environmetrics*, **19**, 331–345.

Febrero-Bande M, Oviedo de la Fuente M (2012). "Statistical Computing in Functional Data Analysis: The R Package **fda.usc**." *Journal of Statistical Software*, **51**, 1–28.

Ferraty F, Vieu P (2006). *Nonparametric Functional Data Analysis: Theory and Practice.* New York.

Fremdt S, Horváth L, Kokoszka P, Steinebach JG (2014). "Functional Data Analysis with Increasing Number of Projections." *Journal of Multivariate Analysis*, **124**, 313–332.

Giacofci M, Lambert-Lacroix S, Marot G, Picard F (2013). "Wavelet-Based Clustering for Mixed-Effects Functional Models in High Dimension." *Biometrics*, **69**, 31–40.

Goldsmith J, Scheipl F, Huang L, Wrobel J, Gellar J, Harezlak J, McLean MW, Swihart B, Xiao L, Crainiceanu C, Reiss PT (2016). ***refund**: Regression with Functional Data.* R package version 0.1-16, URL https://CRAN.R-project.org/package=refund.

Górecki T, Krzyśko M, Waszak Ł, Wołyński W (2016a). "Selected Statistical Methods of Data Analysis for Multivariate Functional Data." *Statistical Papers* doi:10.1007/s00362-016-0757-8.

Górecki T, Krzyśko M, Wołyński W (2015). "Classification Problem Based on Regression Models for Multidimensional Functional Data." *Statistics in Transition New Series*, **16**, 97–110.

Górecki T, Krzyśko M, Wołyński W (2016b). "Multivariate Functional Regression Analysis with Application to Classification Problems." In *Studies in Classification, Data Analysis, and Knowledge Organization: Analysis of Large and Complex Data*, pp. 173–183.

Górecki T, Smaga Ł (2015). "A Comparison of Tests for the One-Way ANOVA Problem for Functional Data." *Computational Statistics*, **30**, 987–1010.

Górecki T, Smaga Ł (2017a). "Multivariate Analysis of Variance for Functional Data." *Journal of Applied Statistics*, **44**, 2172–2189.

Górecki T, Smaga Ł (2017b). ***mfds**: Multivariate Functional Data Sets.* R package version 0.1.0, URL https://github.com/Halmaris/mfds.

Hankin RKS (2005). "Recreational Mathematics with R: Introducing the '**magic**' Package." *R News*, **5**.

Happ C (2017). ***MFPCA**: Multivariate Functional Principal Component Analysis for Data Observed on Different Dimensional Domains.* R package version 1.1, URL https://CRAN.R-project.org/package=MFPCA.

Happ C, Greven S (2016). "Multivariate Functional Principal Component Analysis for Data Observed on Different (Dimensional) Domains." *Journal of the American Statistical Association*. To appear.

Helwig NE (2016). ***bigsplines**: Smoothing Splines for Large Samples.* R package version 1.0-9, URL http://CRAN.R-project.org/package=bigsplines.

Hilgert N, Mas A, Verzelen N (2013). "Minimax Adaptive Tests for the Functional Linear Model." *The Annals of Statistics*, **41**, 838–869.

Hojsgaard S, Halekoh U (2016). ***doBy**: Groupwise Statistics, LSmeans, Linear Contrasts, Utilities.* R package version 4.5-15, URL https://CRAN.R-project.org/package=doBy.

Hormann S, Kidzinski L (2015). ***freqdom**: Frequency Domain Analysis for Multivariate Time Series.* R package version 1.0.4, URL https://CRAN.R-project.org/package=freqdom.

Horváth L, Kokoszka P (2012). *Inference for Functional Data with Applications.* Springer-Verlag, New York.

Hyndman RJ, Shang HL (2016). ***ftsa**: Functional Time Series Analysis.* R package version 4.7, URL http://cran.r-project.org/package=ftsa.

Jacques J, Preda C (2014). "Model-Based Clustering for Multivariate Functional Data." *Computational Statistics & Data Analysis*, **71**, 92–106.

James GM, Hastie TJ (2001). "Functional Linear Discriminant Analysis for Irregularly Sampled Curves." *Journal of the Royal Statistical Society B*, **63**, 533–550.

Jank W, Shmueli G (2006). "Functional Data Analysis in Electronic Commerce Research." *Statistical Science*, **21**, 155–166.

Keser IK, Kocakoç ID (2015). "Smoothed Functional Canonical Correlation Analysis of Humidity and Temperature Data." *Journal of Applied Statistics*, **42**, 2126–2140.

Kidzinski Ł, Jouzdani N, Kokoszka P (2016). ***pcdpca**: Dynamic Principal Components for Periodically Correlated Functional Time Series.* R package version 0.2.1, URL https://CRAN.R-project.org/package=pcdpca.

Krzyśko M, Smaga Ł (2017). "An Application of Functional Multivariate Regression Model to Multiclass Classification." *Statistics in Transition new series.* To appear.

Krzyśko M, Waszak Ł (2013). "Canonical Correlation Analysis for Functional Data." *Biometrical Letters*, **50**, 95–105.

Leeb R, Lee F, Keinrath C, Scherer R, Bischof H, Pfurtscheller G (2007). "Brain-Computer Communication: Motivation, Aim, and Impact of Exploring a Virtual Apartment." *IEEE Transactions on Neural Systems & Rehabilitation Engineering*, **15**, 473–482.

Lichman M (2013). "UCI Machine Learning Repository." URL http://archive.ics.uci.edu/ml.

Lila E, Sangalli LM, Ramsay J, Formaggia L (2016). ***fdaPDE**: Functional Data Analysis and Partial Differential Equations; Statistical Analysis of Functional and Spatial Data, Based on Regression with Partial Differential Regularizations.* R package version 0.1-4, URL https://CRAN.R-project.org/package=fdaPDE.

Long W, Li N, Wang H, Cheng S (2012). "Impact of US Financial Crisis on Different Countries: Based on the Method of Functional Analysis of Variance." *Procedia Computer Science*, **9**, 1292–1298.

Ma S, Yang L, Carroll RJ (2012). "A Simultaneous Confidence Band for Sparse Longitudinal Regression." *Statistica Sinica*, **22**, 95–122.

Maierhofer T (2017). ***classiFunc**: Classification of Functional Data.* R package version 0.1.0, URL https://CRAN.R-project.org/package=classiFunc.

Markussen B (2014). ***fdaMixed**: Functional Data Analysis in a Mixed Model Framework.* R package version 0.4, URL https://CRAN.R-project.org/package=fdaMixed.

Martínez-Camblor P, Corral N (2011). "Repeated Measures Analysis for Functional Data." *Computational Statistics & Data Analysis*, **55**, 3244–3256.

Morris JS (2015). "Functional Regression." *Annual Review of Statistics and Its Application*, **2**, 321–359.

Ogden RT, Miller CE, Takezawa K, Ninomiya S (2002). "Functional Regression in Crop Lodging Assessment with Digital Images." *Journal of Agricultural, Biological, and Environmental Statistics*, **7**, 389–402.

Ojeda Cabrera JL (2012). ***locpol** - Kernel Local Polynomial Regression.* R package version 0.6-0, URL http://CRAN.R-project.org/package=locpol.

Olszewski RT (2001). "Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data." *Ph.D. dissertation, Carnegie Mellon University, Pittsburgh.*

Park J, Ahn J (2016). "Clustering Multivariate Functional Data with Phase Variation." *Biometrics* doi:10.1111/biom.12546.

Parodi A, Patriarca M, Sangalli L, Secchi P, Vantini S, Vitelli V (2015). ***fdakma**: Functional Data Analysis: K-Mean Alignment.* R package version 1.2.1, URL https://CRAN.R-project.org/package=fdakma.

Pauly M, Brunner E, Konietschke F (2015). "Asymptotic Permutation Tests in General Factorial Designs." *Journal of the Royal Statistical Society B*, **77**, 461–473.

Peng J, Paul D (2011). ***fpca**: Restricted MLE for Functional Principal Components Analysis.* R package version 0.2-1, URL https://CRAN.R-project.org/package=fpca.

Peng QY, Zhou JJ, Tang NS (2016). "Varying Coefficient Partially Functional Linear Regression Models." *Statistical Papers*, **57**, 827–841.

Pfeiffer RM, Bura E, Smith A, Rutter JL (2002). "Two Approaches to Mutation Detection Based on Functional Data." *Statistics in Medicine*, **21**, 3447–3464.

Pini A, Vantini S (2015). ***fdatest**: Interval Testing Procedure for Functional Data.* R package version 2.1, URL https://CRAN.R-project.org/package=fdatest.

R Core Team (2017). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Ramsay JO, Hooker G, Graves S (2009). *Functional Data Analysis with R and MATLAB.* Springer-Verlag, Berlin.

Ramsay JO, Silverman BW (2005). *Functional Data Analysis.* 2nd edition. Springer-Verlag, New York.

Ramsay JO, Wickham H, Graves S, Hooker G (2014). ***fda** - Functional Data Analysis.* R package version 2.4.4, URL http://CRAN.R-project.org/package=fda.

Revolution Analytics, Weston S (2015). ***doParallel** - Foreach Parallel Adaptor for the '**parallel**' Package.* R package version 1.0.10, URL http://CRAN.R-project.org/package=doParallel.

Robbiano S, Saumard M, Curé M (2015). "Improving Prediction Performance of Stellar Parameters Using Functional Models." *Journal of Applied Statistics*, **43**, 1465–1476.

Rodriguez JJ, Alonso CJ, Maestro JA (2005). "Support Vector Machines of Intervalbased Features for Time Series Classification." *Knowledge-Based Systems*, **18**, 171–178.

Rossi N, Wang X, Ramsay JO (2002). "Nonparametric Item Response Function Estimates with the EM Algorithm." *Journal of Educational and Behavioral Statistics*, **27**, 291–317.

Schmidberger M, Morgan M, Eddelbuettel D, Yu H, Tierney L, Mansmann U (2009). "State of the Art in Parallel Computing with R." *Journal of Statistical Software*, **31**, 1–27.

Shang HL (2013). "**ftsa**: An R Package for Analyzing Functional Time Series." *The R Journal*, **5**, 64–72.

Shang HL, Hyndman RJ (2013). ***fds**: Functional Data Sets.* R package version 1.7, URL https://CRAN.R-project.org/package=fds.

Shang HL, Hyndman RJ (2016). ***rainbow**: Rainbow Plots, Bagplots and Boxplots for Functional Data.* R package version 3.4, URL https://CRAN.R-project.org/package=rainbow.

Shen Q, Faraway J (2004). "An F Test for Linear Models with Functional Responses." *Statistica Sinica*, **14**, 1239–1257.

Shi JQ, Cheng Y (2014). ***GPFDA**: Apply Gaussian Process in Functional Data Analysis.* R package version 2.2, URL https://CRAN.R-project.org/package=GPFDA.

Smaga Ł (2017). "Repeated Measures Analysis for Functional Data Using Box-Type Approximation - with Applications." *REVSTAT*. To appear.

S original by Jim Ramsey R port by Brian Ripley (2015). ***pspline** - Penalized Smoothing Splines.* R package version 1.0-17, URL http://CRAN.R-project.org/package=pspline.

Soueidatt M, * (2014). ***Funclustering**: A Package for Functional Data Clustering.* R package version 1.0.1, URL https://CRAN.R-project.org/package=Funclustering.

Tarabelloni N, Arribas-Gil A, Ieva F, Paganoni AM, Romo J (2017). ***roahd**: Robust Analysis of High Dimensional Data.* R package version 1.3, URL https://CRAN.R-project.org/package=roahd.

Tarrío-Saavedra J, Naya S, Francisco-Fernández M, Artiaga R, Lopez-Beceiro J (2011). "Application of Functional ANOVA to the Study of Thermal Stability of Micro-Nano Silica Epoxy Composites." *Chemometrics and Intelligent Laboratory Systems*, **105**, 114–124.

Tavakoli S (2015). ***ftsspec**: Spectral Density Estimation and Comparison for Functional Time Series.* R package version 1.0.0, URL https://CRAN.R-project.org/package=ftsspec.

Teisseyre P, Kłopotek RA, Mielniczuk J (2016). "Random Subspace Method for High-dimensional Regression with the R Package **regRSM**." *Computational Statistics*, **31**, 943–972.

Tokushige S, Yadohisa H, Inada K (2007). "Crisp and Fuzzy k-Means Clustering Algorithms for Multivariate Functional Data." *Computational Statistics*, **22**, 1–16.

Tucker JD (2017). ***fdasrvf**: Elastic Functional Data Analysis.* R package version 1.8.1, URL https://CRAN.R-project.org/package=fdasrvf.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S.* 4th edition. Springer-Verlag, New York. ISBN 0-387-95457-0, URL http://www.stats.ox.ac.uk/pub/MASS4.

Walter W, Sanchez-Cabo F, Ricote M (2015). "**GOplot**: An R Package for Visually Combining Expression Data with Functional Analysis." *Bioinformatics.*

Wang JL, Chiou JM, Müller HG (2016). "Functional Data Analysis." *Annual Review of Statistics and Its Application*, **3**, 257–292.

Wickham H (2009). ***ggplot2** - Elegant Graphics for Data Analysis.* Springer-Verlag, New York.

Wrobel J, Goldsmith J (2016). ***refund.shiny**: Interactive Plotting for Functional Data Analyses.* R package version 0.3.0, URL https://CRAN.R-project.org/package=refund.shiny.

Yamamoto M, Terada Y (2014). "Functional Factorial K-means Analysis." *Computational Statistics & Data Analysis*, **79**, 133–148.

Yassouridis C (2016). ***funcy**: Functional Clustering Algorithms.* R package version 0.8.5, URL https://CRAN.R-project.org/package=funcy.

Zhang JT (2011). "Statistical Inferences for Linear Models with Functional Responses." *Statistica Sinica*, **21**, 1431–1451.

Zhang JT (2013). *Analysis of Variance for Functional Data.* Chapman & Hall, London.

Zhang JT, Chen JW (2007). "Statistical Inferences for Functional Data." *The Annals of Statistics*, **35**, 1052–1079.

Zhang JT, Cheng MY, Tseng CJ, Wu HT (2016). "A New Test for One-Way ANOVA with Functional Data and Application to Ischemic Heart Screening." ArXiv:1309.7376v1.

Zhang JT, Liang X (2014). "One-Way ANOVA for Functional Data via Globalizing the Pointwise F-Test." *Scandinavian Journal of Statistics*, **41**, 51–71.

Affiliation:

Tomasz Górecki, Łukasz Smaga
Faculty of Mathematics and Computer Science
Adam Mickiewicz University
Umultowska 87, 61-614 Poznań, Poland
E-mail: tomasz.gorecki@amu.edu.pl, ls@amu.edu.pl