VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
MODELLING AND DATA ANALYSIS
MASTER'S STUDY PROGRAMME

DETECTING CANCEROUS TISSUE IN MAGNETIC RESONANCE IMAGES BY FUNCTIONAL DATA ANALYSIS: A CASE STUDY

Master's thesis

Author: Sabine Püls
VU email address: sabine.puls@mif.stud.vu.lt
Supervisor: Prof. Jurgita Markevičiūtė

Vilnius

2023

Abstract

In this work we demonstrate how to preprocess MRI data of prostate cancer to get discrete time series. This series are smoothed into functional data by B-splie smoothing and differences are detected between functions for malign tissue and functions for healthy tissue. Classification of the tissues follows biopsy results. By a functional t-test we identify an intervall with statistically significant difference between both groups of functions.

Keywords: Functional Data Analysis, Prostate Cancer, MRI, Functional t-Test, Functional ANOVA

Acknowledgements

I want to thank Dr. Mantas Trakymas and Ieva Naruševičiūtė of the National Cancer Institute of Lithuania for providing the data used as base of this work and Justinas Jucevičius for the help in preparing it. An additional gratitude is for Prof. Jurgita Markevičiūtė and Prof. Alfredas Račkauskas for providing help and advice in the process of writing this work. I thank Roman Surkant for explaining the TVM to me. And I want to thank my husband and son for their patience with me over the last months.

Contents

Notation, Abbrevation

- MRI *Magnetic Resonance Imaging*
- FDA *Functional Data Analysis*
- SLIC *Simple Linear Iterative Clustering*
- TVM *Temporal Variation Matrix*
- MSE *Mean-Square-Error*
- LOOCV *Leave-One-Out Cross Validation*
- BD *Band Depth*
- MBD *Modified Band Depth*
- ANOVA *Analyse of Variance*

1 Introduction

Prostate cancer is the most commonly diagnosed male malignancy worldwide and the fourth leading cause of cancer death in men. This amounted worldwide to 1,414,249 newly diagnosed cases and 375,000 deaths from this disease in 2020 [11, 14, 27, 25, 22].

Biopsy is one of the "hands on" methods, to detect prostate cancer. But computer simulations showed, that the risk to miss a cancer by sextant biopsy is estimated to approximately 25% [3]. Also a repeated sextant biopsy of 118 males, failed to detect cancer in 27 men (23%) [16]. At the opposite magnetic resonance images (MRI) are well known to detect cancer of different kind noninvasively [20, 12, 5]. For MRI the patients get an injection of contrast fluid, which reinforces visible differences between different tissues. The resulting image could be imagined as a three dimensional cube, cutted visually into two dimensional slices. Each slice is a grayscale image, where the brightness represents the tissue's intensity to react to the constrast fluid.

At present time, the processing and interpretation of prostate MRI data in clinical routine is entirely performed by human experts (radiologists) who, while competent, are time-limited, cost-intensive, and cannot be easily scaled to meet increasing imaging demands [13]. Furthermore, human performance is dependent on experience and training, leading to significant variability between observers [19, 7, 8].

A computer based method well used for comparing time series is functional data analysis (FDA) [18]. A review of a large number of FDA related publications across various fields of science has shown, that the majority is related to biomedicine applications (21.4%) [28]. Even if FDA was used already for cancer research [2] and biological image processing [26], the new approach to use the TVM in segmentiation without implied anatomical classification invited by Surkant in 2022 [24], makes the use of FDA as planned in this thesis a fully new approach.

Within this method the MRIs of different timesteps are merged slicewise by calculating pixel by pixel the statistical standard deviation over time, getting a new image with equivalent slices named temporal variation matrix (TVM, see equation (1)). Automated segmentation of the TVM is performed by simple linear iterative clusterin (SLIC [1]). The resulting segments, named superpixels, are projected to the original MRI images, grouping the pixels by intensity change over time. By calculating the intensity's mean, each segment get a key value for each timestep, resulting into a discrete time series of intensities for each segment of each slice of the MR-images.

This work will apply FDA methods onto the discrete time series resulting from the use of TVM and SLIC. By doing so, this work will show how to differ between curves representing healthy tissues and curves representing cancerous tissues. Following methods will be helpful: (i) smoothing via B-spline basis, (ii) discriptiv analysis of the resulting curves, (iii) label the curves by biopsy's result, (iv) performing a functional t-test between the groups labeled "healthy" and "cancerous", (v) calculating the maximum velocity of each curve in an intervall, determined by the t-test's results and (vi) use the analyse of variance (ANOVA) method to set the group of functions classified as "undetermined" in context.

This research will show, that it is possible to differ between the groups of "healthy" and "cancerous" tissues with true positive's percentage of nearly 94% (15 out of 16). Statistical significance is shown by a functional t-test. Also the ANOVA shows that the group of functions classified by the histologists as "undetermined" is less similar with the group of "ill" than with the group of "healthy". But the differences between the groups of "ill" and "healthy" is nearly 4 times larger, which suggests to locate the "undetermined" curves inbetween the "healthy" and "ill" ones.

2 Literature Review

As mentioned above, FDA methods are well used in biomedicine field. Crawford et al [2] used the FDA methods to detect glioblastoma multiforme, an agressive form of human brain cancer. Although they used MRI data too, their preprocessing contained the construction of surfaces and shapes to spot growing areas (probable cancer) in the brain. They developed a statistic, called "smooth Euler characteristic transform", that summarizes shape information of glioblastoma multiforme MRI as a collection of smooth curves. Onto this curves they apply FDA methods by using tumor shape information as a covariate in regression frameworks. In opposite to this work, there were no contrast fluid used and also the MRI data was recorded during separate visits instead at different timesteps within one visit of the patient, as it is the case in this work.

Ferro et al [6] collected a list of 30 studies using MRI techniques within the last 5 years. Around one half of them used MRI, the other half used multiparametrial MRI as base. Only two did the segmentation of the images semiautomatic instead of fully manually. The authors conclude that "to accurately distinguish cancerous versus benign tissue, radiometrics has to benefit from technological improvement in segmentation, feature extraction, statistical analysis, multi-center, prospective RCTs to be integrated in clinical practice and in decision-making protocols" [6]. In opposite to this the present work does the segmentation full automated by the SLIC algorithm.

Sunoqrot et al [23] provide an overview about artificial intelligent (AI) methods to process prostate MRIs. They focus on the availability of data, tools for prostate MRI and challenges to measure the quality of results. For AI the existence of large, well curated and diverse datasets is crucial. Sunoqrot et al collected 17 public datasets including a total of 3.369 prostate MRI cases. Actual there are approaches to ease the access for institutions making their data available (Pro-Cancer-I platform [30]) aiming national and international medical data sharing regulations. Implemented AI is found in the products of 11 vendors, five do automated anatomy segmentation, two generate heatmaps to help spot tumors and 3 provide automated tumor detection. Problems occure, if developers want to benchmark against this models. Either the access is not possible or they need to use the source code, install libraries and make changes to fit the model. Possible solution are six mentioned platforms with access to pre-trained models allowing benchmarking easily.

Tian et al [26] highlights the importance of FDA in dimension reduction and feature extraction, spatial classification in MRI studies, and the inverse problem in magneto-encephalography studies. During a brain imaging experiment images are made every one to two seconds for a total time of one to two minutes. This results in 200 to 1000 images. They judge FDA as a considerably effective approach to handle the enormous amount of data, leading to better results in general, than commonly used multivariate methods.

3 Research

3.1 Data preparation - How to get prostate cancer into bits and bytes

By using the effect of magnetic fields onto protons, present in any tissue of the human or animal body, magnetic resonance imaging (MRI) produces noninvasively images of the inside of the body. In difference to computer tomography (CT) there are not even x-rays needed. This makes MRI a visual imaging method, suiting for the need of repetition, as it is given for observations of developments over time. The resulting image represents a three dimensional cube of tissues, organs and similar. By visually "cutting" this cube into slices a set of two dimensional images is formed, one image for each slice. This results into a three dimensional matrix, where each value i represents a pixel at certain coordinates x, y in a certain slice s. Because the MRI is repeated at certain timesteps, in total there will be a 3D matrix for each timestep t.

For this research the dataset "P015" of one patient was used. Each slice of it's MRI has a dimension of 512 to 512 pixels. Each matrix includes 34 slices and there is one matrix for each of the 31 timesteps.

First step in preprocessing the data is the application of the method provided by Surkant et al [24]. It calculates the variance σ^2 for each pixel i at coordinates x, y in the j-th slice s_j over all timesteps $[t_1, t_2, ..., t_T]$ (see equation (1)) resulting in a three dimensional matrix named temporal variation matrix (TVM).

$$i_{TVM}^{(x,y),s_j} = \sigma^2(i^{(x,y),s_j,t_1}, i^{(x,y),s_j,t_2}, ..., i^{(x,y),s_j,t_T}) \tag{1}$$

Next the SLIC algorithm [1] is used to segment the TVM automated. Herefor each slice was handled separately, choosing 50 superpixel cluster centers C_k for each slice, with $k = [1, 50]$ and that followed by calculating distances between pixels in a 5 dimensional space. Three dimensions are caused by the euclidean distance d_{lab} (see equation (2)) in the color space, called CILAB. The remaining two dimensions are caused by the euclidean distance d_{xy} (see equation (3)) of the pixels by coordinates x, y.

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \tag{2}$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \tag{3}$$

The overall distance D_S between center C_k and neighbouring pixel i is the sum of the *lab* distance and the *xy* distance *normalized* by the grid interval $S = \sqrt{N/K}$, where N is the number of pixels in the whole image and $K = 50$ is the number of superpixel cluster centers [1]. By this distance D_S the neighbouring pixels will be assigned to the lowest distant cluster center C_k. Afterwards the segments are projected onto all 34 matrices for 34 timesteps. To get one keyvalue for each segment at each timestep, this research uses the mean as aggregation function for each segment. The end of this preprocessing results in a discrete time series for each segment of each slice (see figure(1)).

After finishing preprocessing the data the methods of functional data analysis are used. First the discrete time series data will be smoothed into continuous curves. Because the "P015" data is non-periodic, this work chose the B-spline basis, following the recommendation of Ramsey and Silverman [18].

A spline function $S_k(t)$ is defined by (i) the order of polynomial segments it consists of and (ii) the

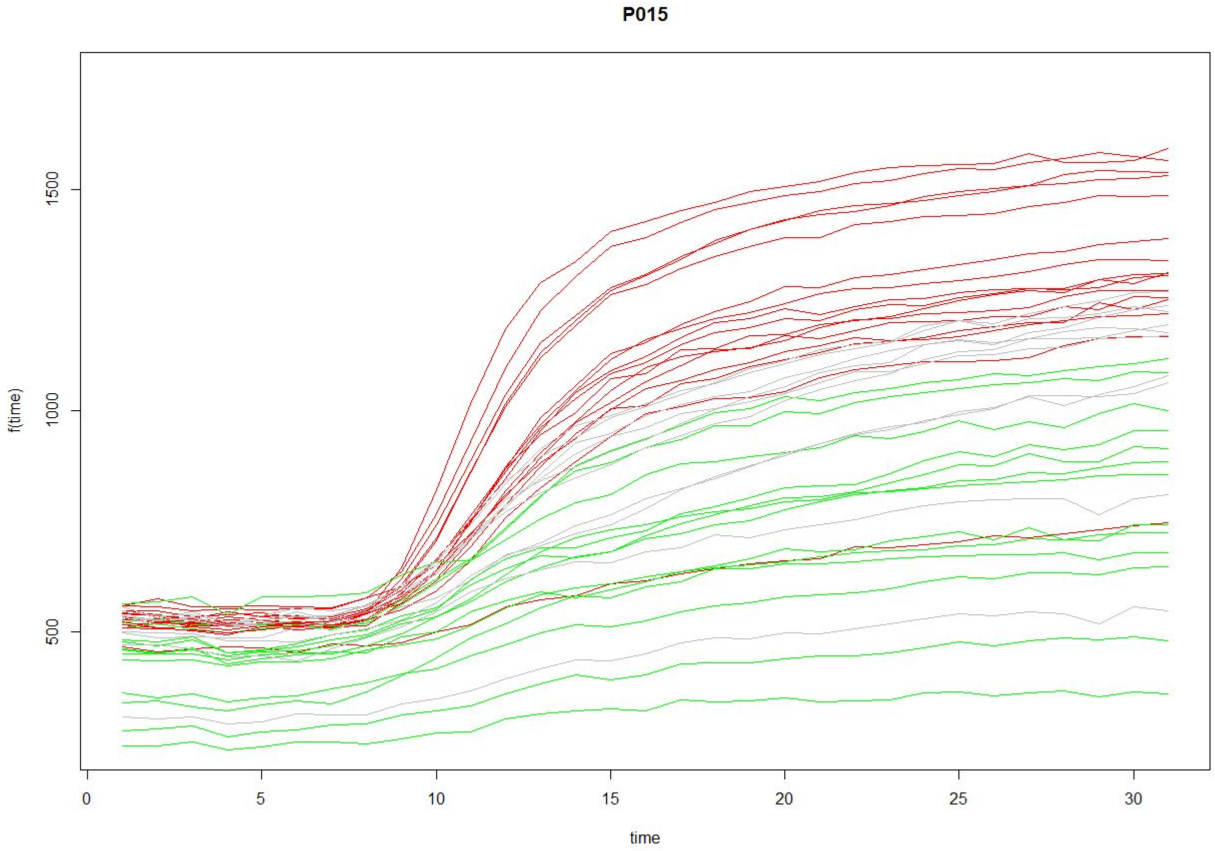

Figure 1: Discrete time series of each classified segment, patient data "P015".

sequence of knots τ, which are the points of matching between discrete time series and continuous curves. The intervall over which a function is to approximated is divided into L subintervalls by τ_l, where $l = 1, ..., L-1$. Each intervall consists of a polynomial of order m, named spline. The B-spline basis system was introduced in 2001 by de Boor (see equation (4),[4]).

$$S(t) = \sum_{k=1}^{m+L-1} c_k B_k(t, \tau), \tag{4}$$

where $B_k(t, \tau)$ is the value of the B-spline basis function defined by the sequence of knots τ at time t, as well as c_k is the coefficient value. For this work we chose order $m = 4$ to get access also to the first and second derivatives of the smoothed curves. Because the number of knots $K = L - 1$ defines how close the smoothed curves will approximate the discrete time series, we need to choose K in an attentive way. The aim is that the smoothing will not cause loss of information (if K is too small) or overfitting by fitting for example noise (if K is too big). To choose the optimal number of knots, we will execute the leave-one-out cross validation (LOOCV, [15]) for $K = [4, 30]$.

LOOCV performs the smoothing with all but one of the discrete time series, called the training set, and the test set (the one time series left out previously). In a second step the smoothed modell is compared with the left out test set by mean-square-error (MSE). This two steps will be repeated until all of the discrete time series happened to be the test set once. By calculating the mean of all MSE a key value is accessable to compare this smoothed model with others, using another value for K. The best fitting smoothing will get the smallest mean of MSE (shown in figure (3), left side). The smoothing with the best fit, in accordance with the LOOCV, provides $K = 10$. The result of smoothing with this parameters is shown in figure (2).

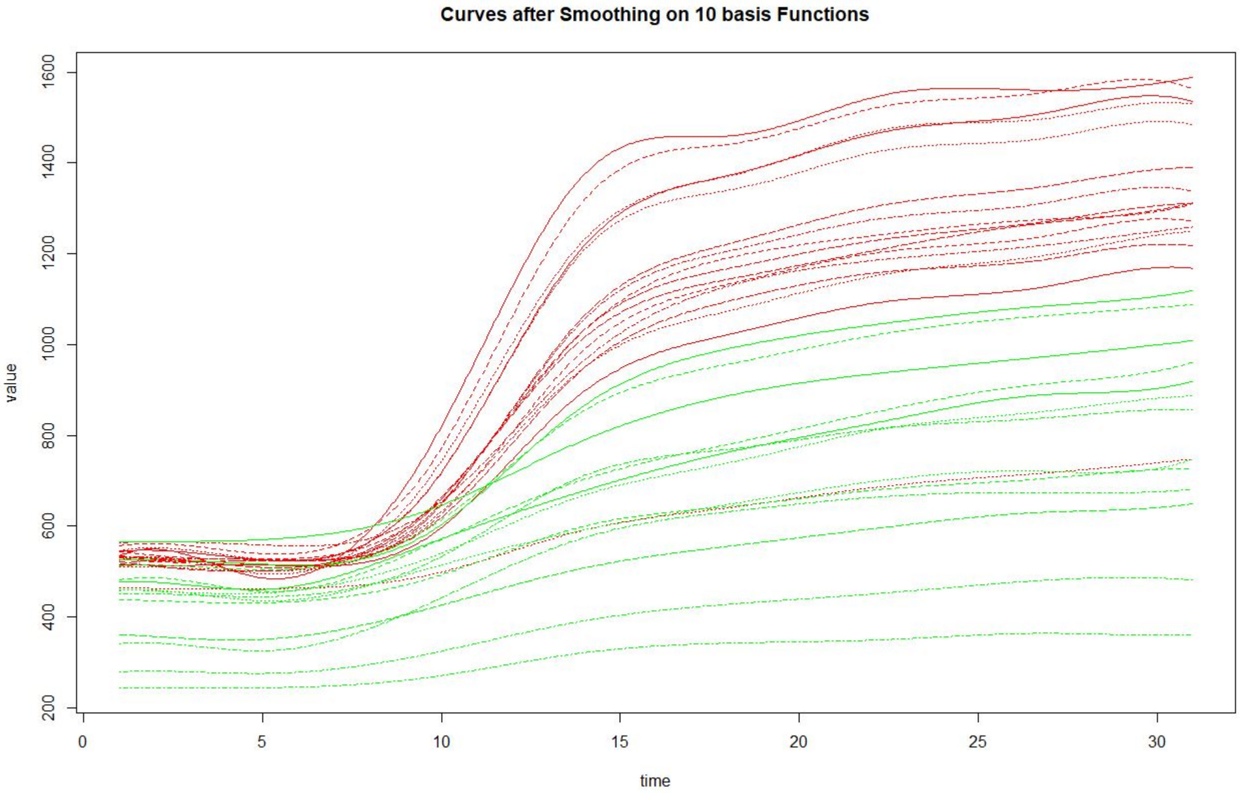

Figure 2: Curves after B-spline smoothing with $K = 10$ basis functions

The last step of data preparation is the classification of smoothed curves by biopsy data. Biopsy means that material is extracted from a region of interest. Because of the shape of needles for doing so, the taken probe has a cylindrical shape with a small diameter (see picture (3), right side). This probe will be classified by a histologist into "healthy", if no cancerous cells were found, or "ill", if cancerous cells were found. A third group is present, classified by histologists as "undetermined". Because biopsy will possible touch more than one slice, it will have effect to the classification of more than one segment and so to more than one curve. The whole probe, and so all touched segments will be classified as "ill", even if cancerous and not cancerous cells were present in the probe at the same time. Contrary to this, a probe will be classified as "healthy" just in absence of cancerous cells. Because of that, we decided to choose the class "healthy" over "ill", plus "healthy" over "undetermined", if one segment was touched by two biopsies with different result. In the data used for this work, this was the case for 10 probes, where "healthy" was chosen over "undetermined". This results into 39 curves, from which 10 are classified as "undetermined" and not handled further, 16 are classified as "ill" and 13 are classified as "healthy".

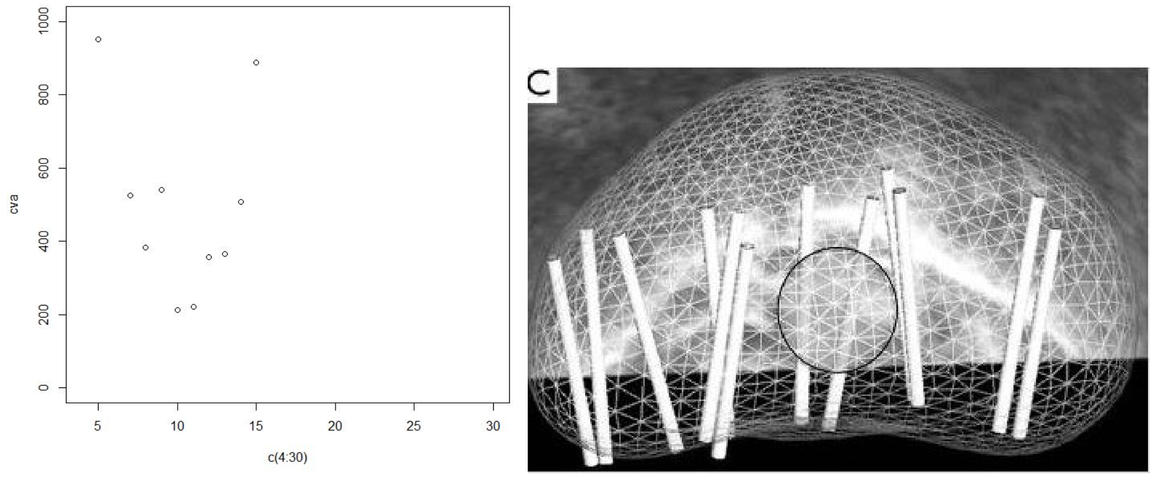

Figure 3:
Left: Results for LOOCV of smoothing on $K = [4, 30]$ basis functions: the minimum value appears for $K = 10$ basis functions.
Right: An example of collected biopsy cores (sagittal view) [29].

3.2 Examination - Characteristics of classified Curves

To examine the curves and get a first impression, the summary statistics mean and standard deviation were calculated. For functional data the mean $\overline{x}$ (see equation(5)) is the average of the functions point-wise across replications [18], where N is the total number of functions; $i = 1, ..., N$ is the index of each single function and $x_i(t)$ is the value of the function with index i at timestep t.

$$\overline{x}(t) = N^{-1} \sum_{i=1}^{N} x_i(t) \tag{5}$$

Similarly is the standard deviation `std` the square root of the point-wise calculated variance function (see equation (6)), where terms are defined in the same way as in equation (5).

$$\texttt{std}_X(t) = \sqrt{(N-1)^{-1} \sum_{i=1}^{N} [x_i(t) - \overline{x}(t)]^2} \tag{6}$$

The graphs are shown in figure (4).
Next step in examining the functions is calculating the first and second derivative. The first derivative gives some information about the velocity, the second one about the accumulation in the original data [18]. The graphs are shown in figure (5).

3.3 Differences - Methods to differ between the groups of functions

At the field of FDA the band depth (BD [21]) gives the option to order all functions resulting from smoothing of the discrete time series. With this order it is possible to detect outliers. By defining a band of functions BD can give a rank of depth of single functions in this band. The smaller the value of BD, the deeper the function is placed inside the band. Let J be the number of functions determining the band and n be the total number of functions with $2 \leq J \leq n$ and let $B(Y_1, \ldots, Y_j)$ be a band defined by j random functions. Additional let $G(y)$ be the subset of a plane, defined by the graph of the function $y(t)$ and let P be the probability measure. Then the band depth BD of function y in a band of size J is defined as the sum of probability P, that the graph $G(y)$ is a subset of each possible band defined by each possible combination of functions Y_1 to Y_j with $j = [2, \ldots, J]$.

$$BD_J(y,P) = \sum_{j=2}^{J} BD^{(j)}(y,P) = \sum_{j=2}^{J} P[G(y) \subset B(Y_1, \ldots, Y_j)] \tag{7}$$

To calculate each possible combination of functions makes this method expensive in matters of computational complexity. In contrast the modified band depth (MBD) is less expensive. It measures for a function $y(t)$ the proportion of time it lays inside the band. If $y(t)$ is part of the band all the time, then MBD degenerates to BD [21]. By use of the MBD measure a boxplot for functions is possible (see figure (9)).

The functional t-test is a method to provide "a sense of the relative separation of two groups of functions" [17]. For this work the absolute value of a t-statistic at each point t was calculated by equation (8), where $\overline{x}_1(t)$ is the mean of group 1 and similar $\overline{x}_2(t)$ is the mean of group 2. By using equation (8) we use the maximum value of the multivariate T-test as test statistic. We use a permutation test to find a critical value of this statistic.

$$T(t) = \frac{\mid \overline{x}_1(t) - \overline{x}_2(t) \mid}{\sqrt{\frac{1}{n_1} Var[x_1(t)] + \frac{1}{n_2} Var[x_2(t)]}} \tag{8}$$

To construct a null distribution, the following steps are repeated 200 times: (i) the labels of the curves are randomly shuffled and (ii) the maximum of $T(t)$ recalculated with the new labels. This provides a reference to evaluate the maximum $T(t)$ of the observed data.

The analyse of variance (ANOVA) is a common method to make a statement about the similarity between groups of data. For functional data Ramsay and Silverman [18] provide a pointwise F test statistic for the one-way analysis of variance (FANOVA). The null hypothesis for FANOVA (see equation (9)) states, that the means $\mu_i(t)$ of l groups of independent random functions $X_{ij}(t)$, defined over a closed and bounded intervall $I = [a, b]$ are equal. The independent random functions $X_{ij}(t)$ are defined with the index of the group $i = 1, \ldots, l$ and the index of the function in one group $j = 1, \ldots, n_i$, while the number of functions $n = n_1 + \cdots + n_l$ and the timestep $t \in I$.

$$H_0 : \mu_1(t) = \cdots = \mu_l(t), t \in I \tag{9}$$

The F-statistic calculates the ratio between the normalized pointwise between-subject variation SSR_n and the normalized pointwise within-subject variation SSE_n (see equation(10)). While l is the number of groups, n is the number of all functions within all groups.

$$F_n(t) = \frac{SSR_n(t)/(l-1)}{SSE_n(t)/(n-l)} \tag{10}$$

Gorecki and Smaga [9],[10] proved, that the F-statistic (see equation (10)) can be calculated approximately equal based on the matrix of the inner product of a functional data object, as described by Ramsay and Silverman [18] and a coefficient calculated from the number of basis functions K, number of functions in total n and number of groups l. Based on this facts, Gorecki and Smaga evolved the permutation test (FP test) provided in the `fanova.tests()` function in the `fdANOVA` package of `R` programming language. Their simulations suggest "that the FP test has better finite sample properties than the F-type and L^2-norm based tests" [10]. For functional data containing few time steps the FP test may also be better than the globalized pointwise F test (GFB test, resp. Fmaxb test) [10].

4 Results

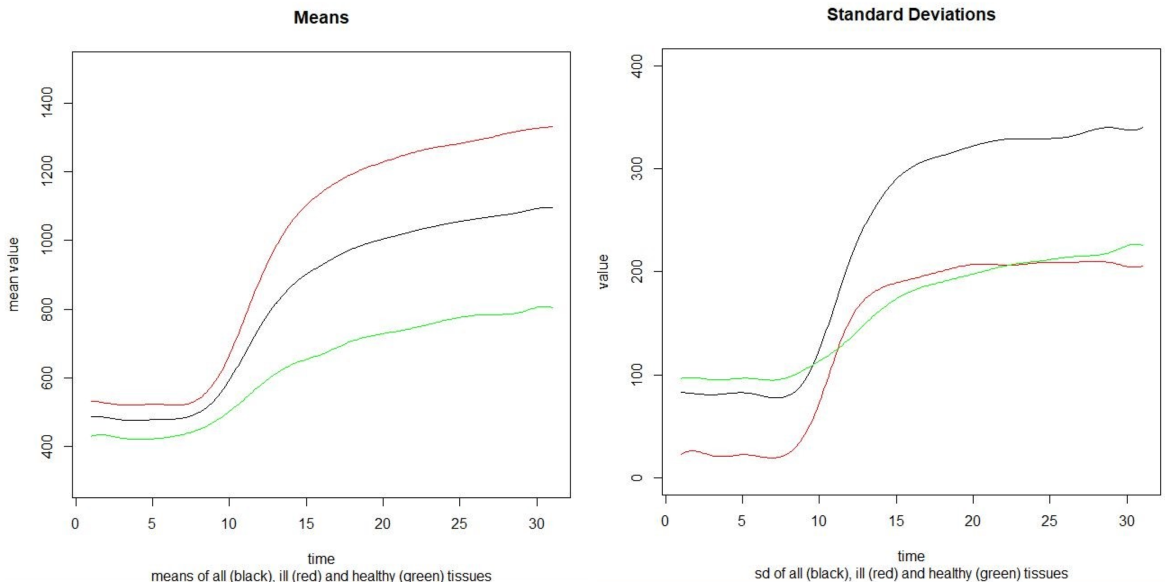

Figure 4: Mean (left side) and standard deviation (right side) of both groups of curves ("ill" in red; "healthy" in green colour) and allover all curves (in black colour).

The mean of functions classified as "healthy" constantly has a smaller functional average than the allover mean. Similar the mean of the functions classified as "ill" stays constantly above the allover mean. With further timesteps the means' difference increases. This is also visible in the rapid increase of the standard deviation from around timestep 10 to around timestep 15 for the "ill" and allover group. After this increase the allover standard deviation is nearly three times the value of the beginning.
As one can see in figure (5), both groups of functions in both derivatives are visibly sorted except of one curve classified as "ill", which is placed between curves classified as "healthy". The first derivative shows a clear difference in the maximum value of green ("healthy") and red ("ill") curves. By calculating the maxima of each curve (see figure (6)), it results in a threshold of value 68. Curves with a maximum above 68 are the ones classified as "ill", curves below 68 are the ones classified as "healthy" (with one exception as mentioned before).

To fortify the previous observation, that the groups differ in the approximated intervall $t[10, 15]$ a funtional t-test (see figure (8)) was performed. In the intervall $t[8, 17]$ the p-value at each timestep is smaller than 0.001. This proofs the hypothesis, that the group classified as "healthy" is significantly different from the group classified as "ill" in that intervall.
The boxplot (see figure (9)), based on the modified band depth (MBD), does provide members of the class "ill" and of the class "healthy" as outliers. Changing the threshold do not change this result, because the boxplot provides symetrical more or less curves of both groups.
To get a perspective about the group of "undetermined" functions, this work decided to calculate an analyse of variance (ANOVA) for different combinations of the groups (see the results in table (1)). While the F-statistic reaches a maximum in comparision of group "healthy" and "ill", it indicates

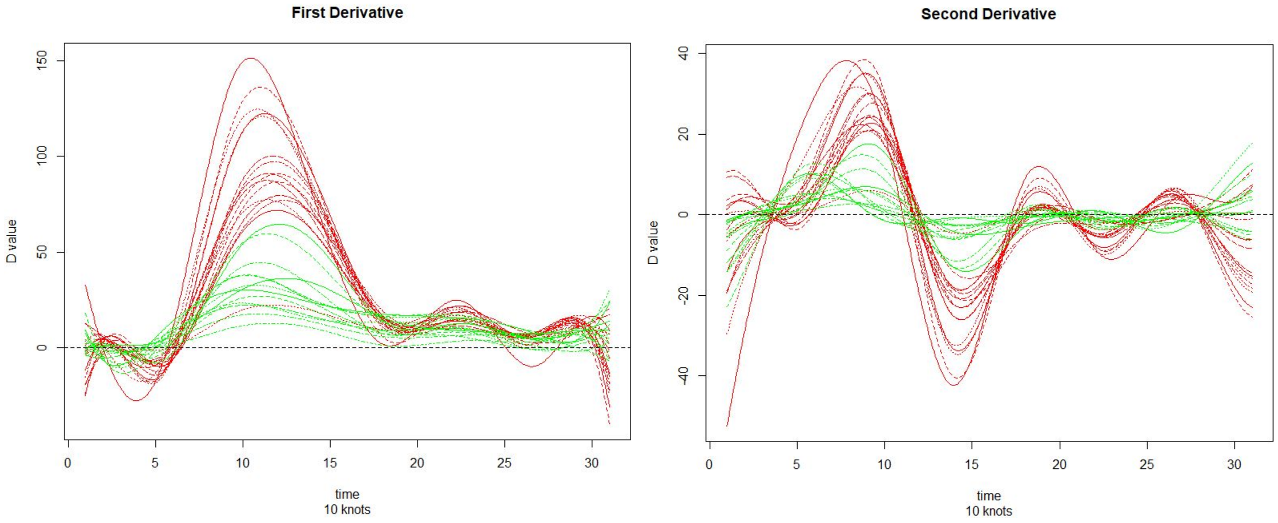

Figure 5:
Left: First derivative - velocity ("healthy" in green, "ill" in red colour);
Right: Second derivative - accumulation ("healthy" in green, "ill" in red colour);

that this two groups are most different from each other. The p-value is for all of the done tests smaller than 2%, which is sufficciently significant. The F-statistics in whole indicate, that the group of "undetermined" functions is more similar to the group of "healthy" functions (F-statistic of 6.84), than to the group of "ill" functions (F-statistic of 10.47). But the difference is not as large, as between "healthy" and "ill", what suggests that the "undetermined" functions lay somewhere inbetween. This supports also the functional ANOVA plot in figure (10).

ANOVA for groups of...	undet., ill, healthy	ill, healthy	undet., ill	undet., healthy
p-value	0	0	0.001	0.015
F-statistic	20.70	40.88	10.47	6.84

Table 1: Results of the functional ANOVA test for different combinations of groups of functions "undetermined" (abbreviated as "undeter."), "ill" and "healthy".

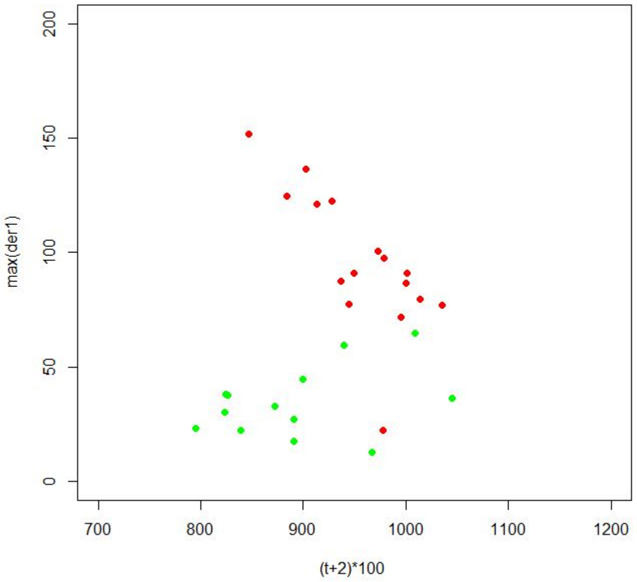

Figure 6: Maximum values of the first derivative in the intervall $t[9,14]$.

5 Conclusions

First hypothesis, that curves for cancerous tissue behaves as outliers in comparision to healthy tissue did not prove true. But in researching the velocity, which is shown in the first derivative of the curves, differences in healthy and cancerous tissue were noticeable with the naked eye. The group of curves representing the healthy tissues reach a maximum velocity beneath the value of 68. Nearly all curves (15 out of 16) in the group representing the cancerous tissues, reach a maximum velocity above 68. The one curve, not fitting into this scheme, could be misclassified. A possible reason for this is the previous discussed nature of biopsis, to classify the whole probe as cancerous, even if there was a mix of cancerous and healthy tissue included. That this special curve represents a tissue segment in the outer region of the prostate area seems to confirm this assumtion.

By performing a pointwise t-test, it is clearly shown that both groups of curves are detectable different. Especially in the intervall t[9,14], where velocity reachs it's maxima, the groups of curves showed a significant difference (more than double of the maximum critical value).

This results could be biological explained by the behaviour of cancerous tissues, to have a more active metabolism than their environment. In this way the contrast fluid may reach cancerous tissues earlier and also in a higher amount than the surrounding tissues. This would lead to an earlier and higher rise of intensity (in comparision to healthy tissues) in the MRI and in the end to the curves we observed in this research. To substanciate this hypothesis, a broader base of data would be needed in minimum two ways. First would be to include more patients into the analyse, so differences between individuals can be researched. The second way would include more classified segements in general per patient to enlarge the ground truth of the data. This may be difficult, because the biopsy is a method the most patients feel uncomfortable with.

The ANOVA test showed, that the group of functions classified by histologists as "undetermined" is in the intensitiy's behavior more similar with the group classified as "healthy", than with the one classified

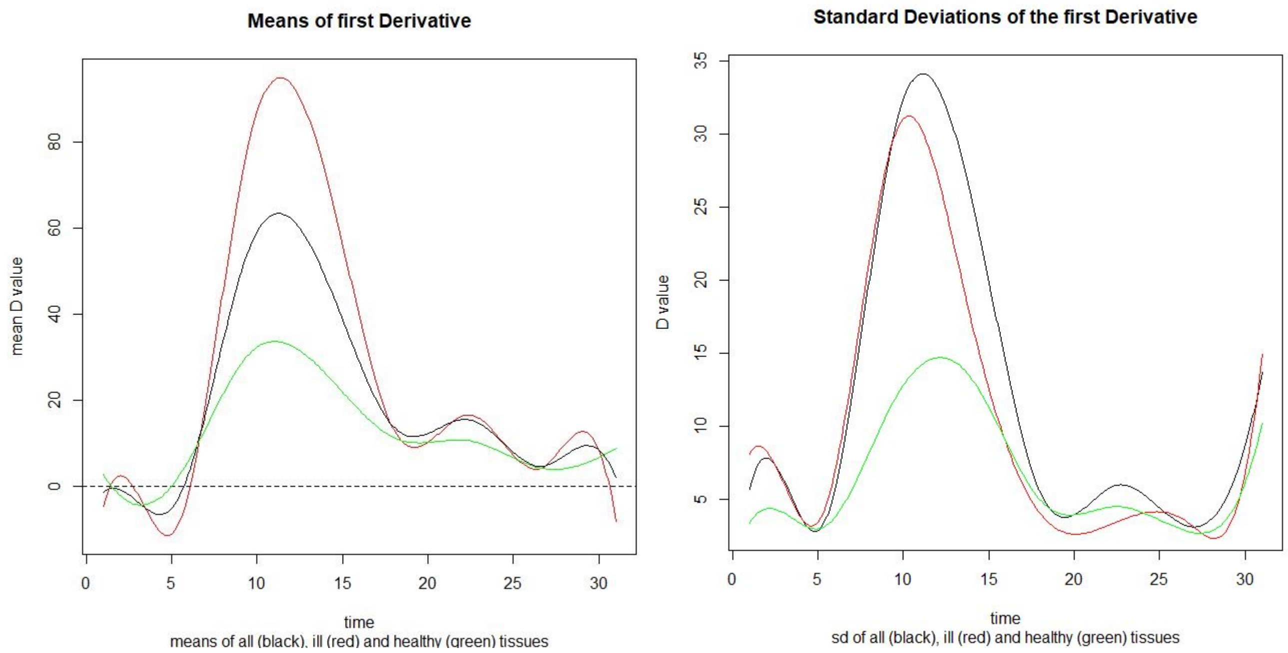

Figure 7:
Left: Mean of the first derivatives;
Right: Standard deviation of the first derivatives;
(Curve of group "healthy" in green, "ill" in red, of all functions in black colour)

as "ill". In the evolvement of cancer the affected cells pass through a step by step process. During this process they switch off the check points one after another, which were originally developed by nature to prevent unregulated growth. Because of this mechanism the threshold between "ill" and "healthy" is not a clear border. The results imply, that the class of "undetermined" biopsies lays in this region of tissue, not as healthy as it should be, but also not as ill as a clear diagnosis of cancer would need.
Besides the need to enlarge the base of data, there is a second possible improvement, which could be worth further research. The SLIC algorithm can process pictures with 3 channels of colour (for example RGB images, with red, green and blue channel). The TVM only allocates one of this channels because it is in grayscale colour, so it is possible to add two more objects with information in the shape of the TVM.
With this work we made a step to automate the process of MRI evaluation. When used on a regular basis this can disburden the specialists who need to spend hours and hours of focussed work to interpret the MRI of dozens of patients. On the long run it could reduce or even replace invasive methods as biopsy, which carry a risk of unwelcome side effects and discomfort for the patients.

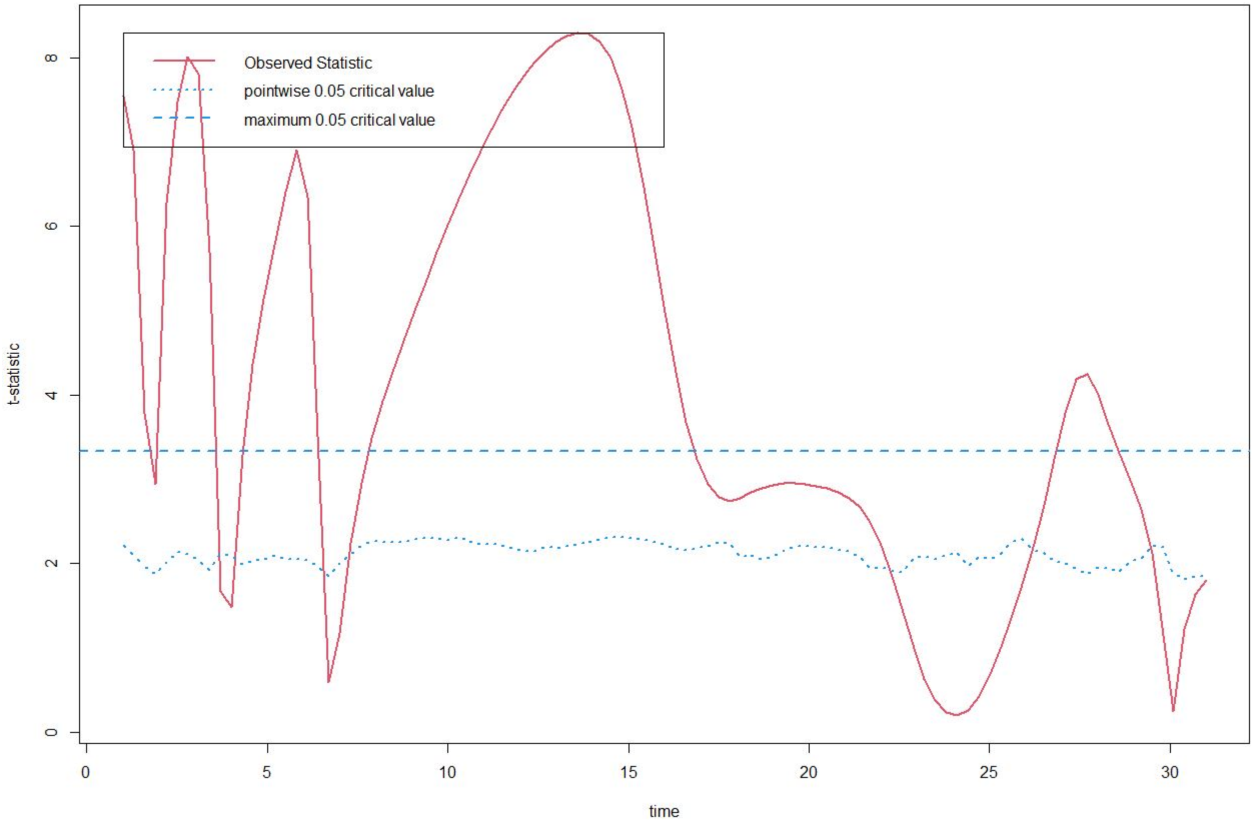

Figure 8: Results of the functional t-test between the group of curves classified as "cancerous" and the group of curves classified as "healthy".

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk. SLIC superpixels, *Technical report, EPFL*, 2010, **06**.

[2] L. Crawford, A. Monod, A. X. Chen, S. Mukherjee, R. Rabadan. Predicting Clinical Outcomes in Glioblastoma: An Application of Topological and Functional Data Analysis, *Journal of the American Statistical Association*, 2019, **115**.

[3] F. Daneshgari, G.D. Taylor, G.J. Miller, E.D. Crawford. Computer simulation of the probability of detecting low volume carcinoma of the prostate with six random systematic core biopsies, *Urology.* 1995, **45**, p.p. 604-609.

[4] C. de Boor. *A Practical Guide to Splines*, New York: Springer, 2001

[5] M. Eklund, F. Jaderling, A. Discacciati. MRI-targeted or standard biopsy in prostate cancer screening, *The New England journal of medicine.* 2021, **385**(10), p.p. 908–920.

[6] M. Ferro, O. de Cobelli, G. Musi, F. Del Giudice, G. Carrieri, G.M. Busetto, U.G. Falagario, A. Sciarra, M. Maggi, F. Crocetto, B. Barone, V.F. Caputo, M. Marchioni, G. Lucarelli, C. Imbimbo, F.A. Mistretta, S. Luzzago, M.D. Vartolomei, L. Cormio, R. Autorino, O.S. Tătaru. Radiomics in prostate cancer: an up-to-date review, *Therapeutic advances in urology*, 2022, **14**.

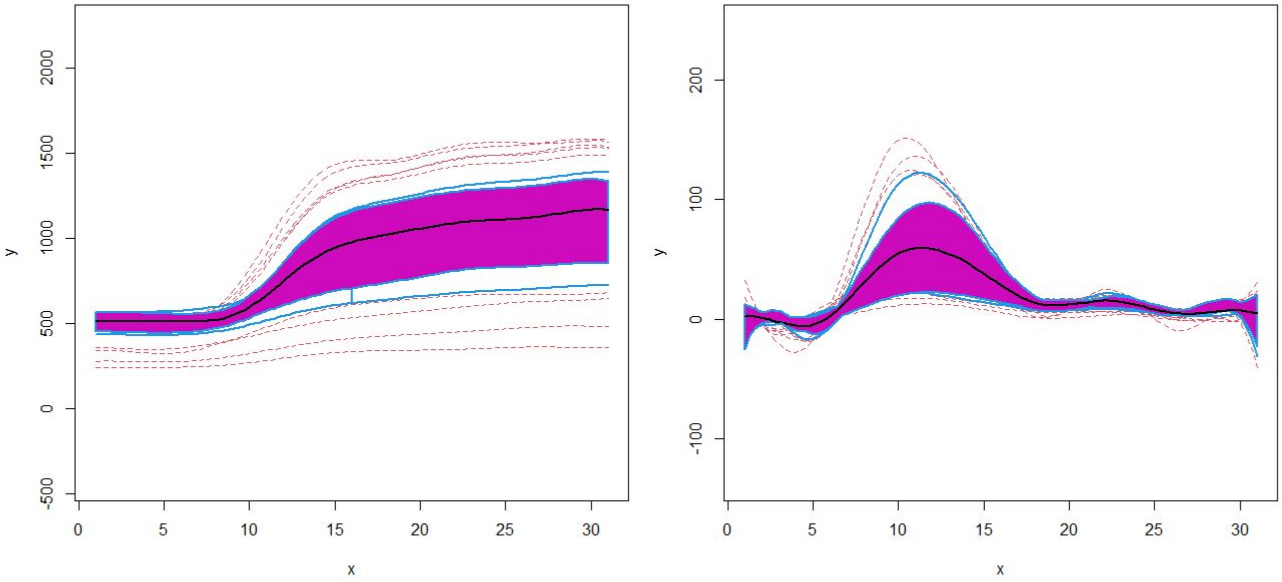

Figure 9: The functional boxplots of the initial functions (left side) and the first derivatives (right side) at a threshold of 0.5

[7] M. Gatti, R. Faletti, G. Calleris, J. Giglio, C. Berzovini, F. Gentile, G. Marra, F. Misischi, L. Molinaro, L. Bergamasco, P. Gontero, M. Papotti, P. Fonio. Prostate cancer detection with biparametric magnetic resonance imaging (bpMRI) by readers with different experience: performance and comparison with multiparametric (mpMRI), *Abdominal radiology*, 2019, **44**(5), p.p. 1883–1893.

[8] M.D. Greer, J.H. Shih, N. Lay, T. Barrett, L. Bittencourt, S. Borofsky, I. Kabakus, Y.M. Law, J. Marko, H. Shebel, M.J. Merino, B.J. Wood, P.A. Pinto, R.M. Summers, P.L. Choyke, B. Turkbey. Interreader Variability of Prostate Imaging Reporting and Data System Version 2 in Detecting and Assessing Prostate Cancer Lesions at Prostate MRI, *AJR. American journal of roentgenology*, 2019, p.p. 1–8.

[9] T. Górecki, Ł. Smaga. A Comparison of Tests for the One-Way ANOVA Problem fo Functional Data, *Computational Statistics*, 2015, **30**, p.p. 987–1010.

[10] T. Górecki, Ł. Smaga. fdANOVA: an R software package for analysis of variance for univariate and multivariate functional data, *Computational Statistics*, 2019, **34**, p.p. 571–597.

[11] A. Jemal, M.M. Center, C. DeSantis, E.M. Ward. Global patterns of cancer incidence and mortality rates and trends, *Cancer Epidemiol Biomarkers Prev*, 2010, **19**(8), p.p. 1893-907.

[12] V. Kasivisvanathan, A.S. Rannikko, M. Borghi, V. Panebianco, L.A. Mynderse, M.H. Vaarala, A. Briganti, L. Budäus, G. Hellawell, R.G. Hindley, M.J. Roobol, S. Eggener, M. Ghei, A. Villers, F. Bladou, G.M. Villeirs, J. Virdi, S. Boxler, G. Robert, P.B. Singh, ... PRECISION Study Group Collaborators. MRI-Targeted or Standard Biopsy for Prostate-Cancer Diagnosis, *The New England journal of medicine* 2018, **378**(19), p.p. 1767–1777.

[13] G. Litjens, O. Debats, J. Barentsz, N. Karssemeijer, H. Huisman. Computer-aided detection of prostate cancer in MRI, *IEEE Trans Med Imaging*, 2014, **33**, p.p. 1083–1092.

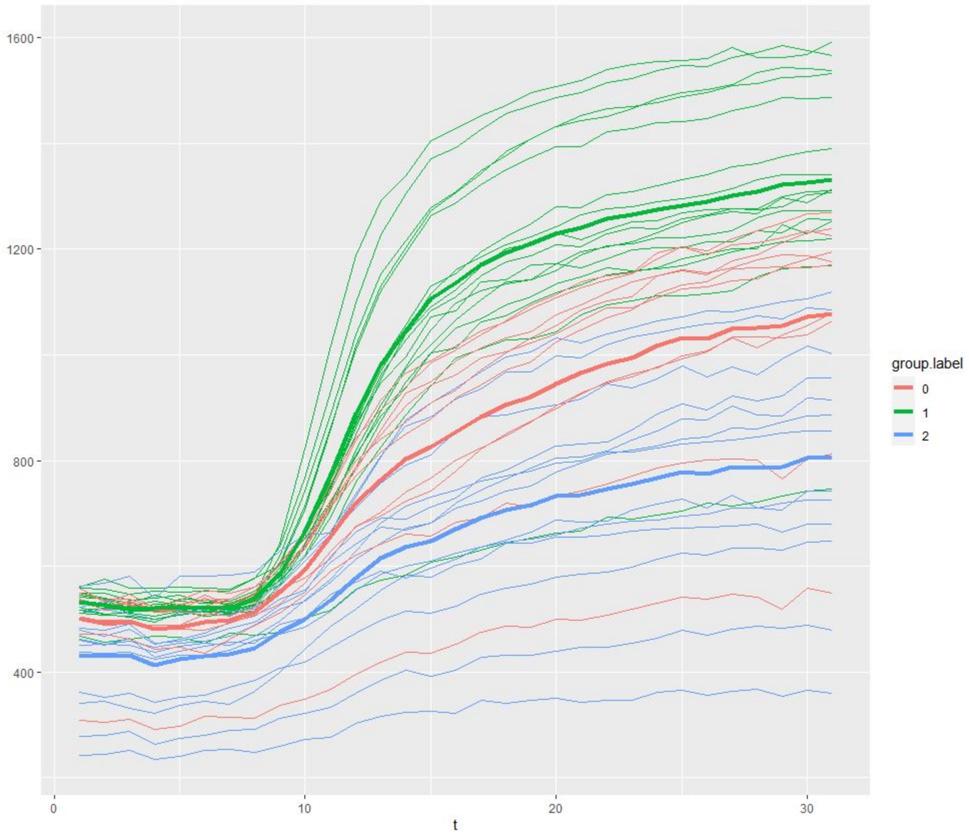

Figure 10: Result of the functional ANOVA plot; Group label 0 (red) stands for "undetermined", 1 (green) for "ill" and 2 (blue) for the group of functions classified as "healthy".

[14] C. Mattiuzzi, G. Lippi. Current Cancer Epidemiology. *Journal of epidemiology and global health*, 2019, **9**(4), p.p. 217-222.

[15] A.M. Molinaro, R. Simon, R.M. Pfeiffer. Prediction error estimation: a comparison of resampling methods, *Bioinformatics*, 2005, **21**(15), p.p. 3301–3307

[16] F. Rabbani, N. Stroumbakis, B.R. Kava, M.S. Cookson, W.R. Fair. Incidence and clinical significance of false-negative sextant prostate biopsies, *The Journal of urology*, 1998, **159**(4), p.p. 1247–1250.

[17] J.o. Ramsay, G. Hooker, S. Graves. *Functional data analysis with R and MATLAB*, New York: Springer,2009.

[18] J.O. Ramsay, B.W. Silverman. *Functional Data Analysis*, New York: Springer, 2005.

[19] A.B. Rosenkrantz, A. Ayoola, D. Hoffman, A. Khasgiwala, V. Prabhu, P. Smereka, M. Somberg, S.S. Taneja. The Learning Curve in Prostate MRI Interpretation: Self-Directed Learning Versus Continual Reader Feedback, *AJR. American journal of roentgenology*, 2017, **208**(3), p.p. W92–W100.

[20] O. Rouvière, P. Puech, R. Renard-Penna, M. Claudon, C. Roy, F. Mège-Lechevallier, M. Decaussin-Petrucci, M. Dubreuil-Chambardel, L. Magaud, L. Remontet, A. Ruffion, M. Colombel, S. Crouzet,

A.M. Schott, L. Lemaitre, M. Rabilloud, N. Grenier, MRI-FIRST Investigators. Use of prostate systematic and targeted biopsy on the basis of multiparametric MRI in biopsy-naive patients (MRI-FIRST): a prospective, multicentre, paired diagnostic study, *The Lancet. Oncology*, 2019, **20**(1), p.p. 100–109.

[21] Y. Sun, M. Genton. Functional Boxplot, *Journal of Computational and Graphical Statistics*, 2010, **20**(10).

[22] H. Sung, J. Ferlay, R.L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, F. Bray. Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries, *CA: a cancer journal for clinicians*, 2021, **71**(3), p.p. 209–249.

[23] M.R.S. Sunoqrot, K.M. Selnæs, E. Sandsmark, G.A. Nketiah, O. Zavala-Romero, R. Stoyanova, T.F. Bathen, M. Elschot. A Quality Control System for Automated Prostate Segmentation on T2-Weighted MRI, *Diagnostics (Basel, Switzerland)*, 2020, **10**(9), p.p. 714.

[24] R. Surkant. Evaluation of Dynamic Contrast in Prostate MRI for Cancerous Tissue Identification, *Master's Thesis at Vilnius University, FMI, Modelling and Data Analysis*, 2022.

[25] U. Testa, G. Castelli, E. Pelosi. Cellular and Molecular Mechanisms Underlying Prostate Cancer Development: Therapeutic Implications, *Medicines (Basel, Switzerland)*, 2019, **6**(3), p.p. 82.

[26] T.S. Tian. Functional data analysis in brain imaging studies, *Frontiers in psychology* 2010, **1**, p.p. 35.

[27] L.A. Torre, F. Bray, R.L. Siegel, J. Ferlay, J. Lortet-Tieulent, A. Jemal. Global cancer statistics, 2012, *CA: a cancer journal for clinicians*, 2015, **65**(2), p.p. 87–108.

[28] S. Ullah, C.F. Finch. Applications of functional data analysis: A systematic review, *BMC Med Res Methodol*, 2013, **13**, 43.

[29] H. Uno, T. Taniguchi, K. Seike, D. Kato, M. Takai, K. Iinuma, K. Horie, K. Nakane, T. Koie. The accuracy of prostate cancer diagnosis in biopsy-naive patients using combined magnetic resonance imaging and transrectal ultrasound fusion-targeted prostate biopsy, *Transl Androl Urol*, 2021, **10**(7), p.p. 2982-2989.

[30] ProCAncer-I: An AI Platform integrating imaging data and models, supporting precision care through prostate cancer's continuum. Available via https://www.procancer-i.eu. Accessed 30 May 2022.

6 Appendix A

```r
# please see end of code for details about
# version of R and the packages used
library(oro.dicom);
library(misc3d);
library(abind);
library(OpenImageR);
library(Rcpp);
library(rgl);
library(supercells);
library(sf);
library(terra);
library(fda);
library(dplyr);
library(ggplot2);
library(abind);
library(funFEM);
library(funHDDC);
library(plotly);
library(fda.usc);
library(fdANOVA);

#################################################
# START PREPROCESSING
#################################################


##########
# Function to read DICOM files

prosReadDICOM <- function(path){
  fil <- readDICOMFile(path)
  #names(fil)
  #head(fil$hdr)
  #attributes(fil$img)

  # patient ID
  meta <- {}
  meta$patID <- fil$hdr$value[28]
  # cycle ID

  # slice ID (for now number of slices)
  meta$slices <- attributes(fil$img)[3]
  # resolution high and width
  meta$resH <- attributes(fil$img)[1]
  meta$resW <- attributes(fil$img)[2]
  # pixel intensity min, max and mean
  meta$intMax <- max(fil$img)
  meta$intMin <- min(fil$img)
  meta$intMean <- mean(fil$img)
  fil$meta <- meta
  return(fil)
}

###############################################
# Please save raw P015 data plus biopsy.csv data at location specified in "path"
# also add an empty folder named "P015" there (used for "pathOUT")
###############################################

path <- ’C:/Users/Sabine/Studium/MA-2022/Daten/’;

patfol <- list.files(path, pattern=’^[0-9]’);
patfol <- patfol[order(nchar(patfol), patfol)];
for(pat in patfol){
print(’#################################’);
print(pat);
print(’#################################’);

  pathIN <- paste0(path,pat,’/’);
  pathOUT <- paste0(path,’/P’,pat,’/’);

# -------------- LOOP OVER PATIENTS -----------------
#for(pat in patfol){
  #####################################
  # READ IN FILES
  # PreProcessing for TVM
  # read two files of "contrast", substract
  timfol <- list.files(pathIN, pattern=’contrast’);
  timfol <- timfol[order(nchar(timfol), timfol)];

  #############
  # TVM BY VARIANCE: sum((each value - mean)^2) / number of values
  fin <- prosReadDICOM(paste0(pathIN, ’/’, timfol[1]));
  summe <- fin$img;

  for(tistep in timfol[2:length(timfol)]){
    fil <- prosReadDICOM(paste0(pathIN, ’/’, tistep));
    summe <- abs(summe + fil$img);
  }

  mittel <- abs(summe/length(timfol));
  quad <- abs(abs(fin$img - mittel)^2);

  for(tistep in timfol[2:length(timfol)]){
    fil <- prosReadDICOM(paste0(pathIN, ’/’, tistep));
    quad <- abs(quad + abs( abs( fil$img - mittel)^2));
  }

  tvm3d <- abs(quad / length(timfol));

  # have a look into slice 22 and 2 of the TVM
```

```
image(tvm3d[,,22]);
image(tvm3d[,,2]);
saveRDS(tvm3d, paste0(pathOUT,'TVM3D.rds'));

###################################
# SLIC
# for each single Slice in TVM
# as agreed with Roman: each single slice done with SLIC, no whole 3D data in SLIC, "combine" afterwards
# make spatial raster from matrix
# calculate SLIC
# combine slices
# save as "Cells.rds"

tvm3d <- readRDS(paste0(pathOUT,'TVM3D.rds'));
tvmCells <- array(NA,dim=dim(tvm3d));

for(i in c(1:dim(tvm3d)[3])){

  raster <- rast(tvm3d[,,i]);
  rasterSLIC <- supercells(raster, k = 500, compactness = 3, dist_fun = "euclidean", avg_fun = "mean");
  #plot(raster);
  #plot(st_geometry(rasterSLIC), add = TRUE, lwd = 0.2);
  meta <- rasterize(rasterSLIC, raster, "supercells");
  pol <- as.matrix(meta, wide=TRUE);
  #image(pol);
  tvmCells[,,i] <- pol;
}

saveRDS(tvmCells, paste0(pathOUT,'Cells.rds'));

# # first maximize and minimize the graphic window
# # do not change size afterwards -.- nearly freeze
# layout(matrix(1:49,nr=7,byr=T));
# for(i in c(1:44)){
  # image(tvmCells[,,i]);
# }

###################################################
# Loop Over Timesteps after Adding Contrast Fluid

# AGGREGATION by mean
# result: discrete values over time for each supercell in each slice
# matrix[supercell,time]

timfol <- list.files(pathIN, pattern='contrast');
timfol <- timfol[order(nchar(timfol), timfol)];

# matrix[supercellID,timestep]=aggregationvalue
fdalist <- array();
```

```
print("# of cells");
for(j in c(1:dim(tvmCells)[3])){ #init one matrix for each slice in TVM
  print(max(tvmCells[,,j],na.rm=TRUE));
  slice <- array(NA, dim=c(max(tvmCells[,,j],na.rm=TRUE),length(timfol)));
  # name the slices by numbers from 1 to # of slices
  assign(paste0("rawFDA_slice_", j), slice);
  # store the names in a list
  fdalist <- c(fdalist, paste0("rawFDA_slice_", j));
}
# delete first empty element from the list
fdalist <- fdalist[2:length(fdalist)];

# read the contrast files and aggregate
print("time");
for(ti in c(1:(length(timfol)))){ # for each timestep
  #print("time");
  print(ti);
  fil1 <- prosReadDICOM(paste0(pathIN, timfol[ti]));
  # fil1$img is the matrix of intensities
  for(i in 1:dim(fil1$img)[3]){ # for each slice
    #print("slice");
    #print(i);
    #print(max(tvmCells[,,i]));
    for(supi in c(1:max(tvmCells[,,i],na.rm=TRUE))){ # for each supercell
      #print("cell");
      #print(supi);
      alle <- which(tvmCells[,,i] == supi); # extract all pixels contained in supercell
      slice <- fil1$img[,,i];
      agg <- mean(slice[alle]); ###### HERE choose aggregation function ######
      #print(agg);
      temp <- get(fdalist[i]); # get variable's name of matrix for corresponding slice
      temp[supi,ti] <- agg; # assign value (mean) for supercell at timestep
      assign(fdalist[i], temp); # store new matrix under variable's name
    } #end supercells
  } # end slices
} # end timesteps

for(mat in fdalist){ # for each matrix in the list of matrices per slice
  saveRDS(get(mat),paste0(pathOUT,mat,'.rds')) # save matrix under variable's name
}
image(get(fdalist[22])); # have a look into discrete values over time per segment of slice 22

#BIOPSY MASK
#classify supercells by using the biopsy mask
mask <- prosReadDICOM(paste0(pathIN,'biopsyMask.dcm'));
cells <- readRDS(paste0(pathOUT,'Cells.rds'));
maske <- mask$img;
biplog <- c(NA,NA); # initiate storage for supercell/biopsy-result pairs
```

```
  for(slice in c(1:dim(maske)[3])){
    for(biop in c(1:max(maske))){
      bio <- which(maske[,,slice] == biop);
      temp <- cells[,,slice];
      if(length(bio) != 0){
        supi <- temp[bio];
        supi <- unique(supi);
        #print(supi);
        for(i in supi){
          te <- slice*100+i; # i.e. supercell 14 of slice 22 becomes "2214"
          #print(te);
          biplog <- rbind(biplog, c(te,biop));
        }
      }
    }
  }
  biplog <- biplog[2:dim(biplog)[1],]; # delete first empty value
  biplog <- biplog[order(biplog[,1]),]; # order by number of biopsy
  saveRDS(biplog, paste0(pathOUT,'/biopsyCells.rds'));


#}

alarm();

##########################################
# END PREPROCESSING
##########################################

##########################################
# START FDA ANALYSIS
##########################################


  pat <- "P015";
  #################################################
  # please adjust "pathOUT" to your needs         #
  #################################################
  pathOUT <- paste0('C:/Users/Sabine/Studium/MA-2022/Daten/FDAdata/',pat,'/');

  # read in discrete time series
  rawFDAfol <- list.files(pathOUT, pattern='rawFDA_slice');
  rawFDAfol <- rawFDAfol[order(nchar(rawFDAfol), rawFDAfol)];

  # initiate an allover matrix and parallel a list with ordered matrices of all "rawFDA_slice" files
  # matrix[supercellID,timestep]=aggregationvalue
  rawFDA <- readRDS(paste0(pathOUT,rawFDAfol[1]));
  rawFDAs <- list();
  rawFDAs[[1]] <- readRDS(paste0(pathOUT,rawFDAfol[1]));
  i <- 1;
```

```
for(rfda in rawFDAfol[2:length(rawFDAfol)]){
  i <- i+1;
  test <- readRDS(paste0(pathOUT,rfda));
  rawFDA <- rbind(rawFDA, test);
  rawFDAs[[i]] <- test;
}

# CLASSIFY
# find classes
print("biopsy");
cate <- readRDS(paste0(pathOUT,'/biopsyCells.rds'));

# read biopsys csv
pat2 <- gsub('P','',pat);
print("csv");
rawBio <- read.csv(file = paste0('C:/Users/Sabine/Studium/MA-2022/Daten/',pat2,'.csv'), header=FALSE);
biop <- c();
for(b in rawBio$V2){
  b <- gsub(" ", "", b, fixed = TRUE);
  b2 <- switch(b, "Malignant" = 1, "Benign" = 2, "0"); # "Undetermined" = 0
  biop <- c(biop, b2);
}
biop <- as.integer(biop);


# find class for supercells with more than one classification
# classes: 0 undefined, 2 healthy, 1 ill
unicate <- array(NA, 2);
for(cell in unique(cate[,1])){
  a <- cate[which(cate == cell),2];
  b <- biop[a];
  print(cell);
  print(b);
  cl <- max(b); # chooses healthy over ill
  unicate <- rbind(unicate, c(cell,cl));
}
unicate <- unicate[2:dim(unicate)[1],] # delete NA row


# get discrete timeseries for classified supercells
cFDA <- array(NA,length(rawFDA[1,])+1); # store classification and discrete time series
ancest <- array(NA); # store slice-supercell / class pairs for labeling plots


for(thing in c(1:dim(unicate)[1])){
  sc <- unicate[thing,1]%%100;    # number of supercell
  sl <- (unicate[thing,1]-sc)/100;  # number of slice
  cl <- unicate[thing,2];       # classification

```

```
    ancest <- c(ancest, paste0(sl,"-",sc));
    cFDA <- rbind(cFDA, c(cl, rawFDAs[[sl]][sc,]));
  }
  cFDA <- cFDA[2:dim(cFDA)[1],] # delete NA row
  ancest <- ancest[2:length(ancest)]

  # cFDA: first column is class, other values are discrete time series
  # classes: 0 undefined, 2 healthy, 1 ill

  # PLOT
  dev.new();
  colo <- switch(toString(cFDA[1,1]), "0" = "grey", "2" = "green", "1" = "red");

  plot(c(1:(dim(cFDA)[2]-1)), cFDA[1,2:(dim(cFDA)[2])], type = "l", xlab = "time", ylab = "f(time)", main = pat, col = colo, xlim=c(1,dim(rawFDA)[2]),ylim=c(250,1750));
  for(i in c(2:dim(cFDA)[1])){
    colo <- "black";
    colo <- switch(toString(cFDA[i,1]), "0" = "grey", "2" = "green", "1" = "red");
    lines(cFDA[i,2:(dim(cFDA)[2])], lwd = 1, col = colo);
  }


###########################################
# FILTER
# only defined values (ill 1, or healthy 2)
fFDA <- cFDA[which(cFDA[,1] %in% c(1,2)),];
fancest <- ancest[which(cFDA[,1] %in% c(1,2))];

###########################################
# VALIDATE SMOOTHING
mik <- 4; #number of minimal knots
mak <- 30; # number of maximal knots

a1 <- 2; #where starts time series (1 is class, 2 is first value of time series)
a2 <- dim(fFDA)[2]; #where ends time series
time_span <- a2-1;
times_basis = seq(0,time_span,1);

cva <- c(); # store values of cross validation
# smooth discrete time series for each number of knots, cross validate and store result
for(knot in c(mik:mak)){
  basisX <- create.bspline.basis(rangeval=c(min(times_basis)+1,max(times_basis)), norder=4, nbasis=knot);
  smoothX <- S.basis(c(min(times_basis)+1:max(times_basis)), basisX);
  cv <- CV.S(t(fFDA[,a1:a2]), smoothX);
  cva <- c(cva, cv);
}
# dev.new();
# plot(c(4:30),cva, ylim=c(0,1000)); # plot results of cross validations
```

```

m <- min(cva);
i <- which(cva %in% m);
print("best number of knots by cross validation:");
print(i+mik-1);

#############################################
# Smoothing Bspline

knot <- 10;

subtitle <- paste0(knot," basis functions");
basis <- create.bspline.basis(rangeval=c(min(times_basis)+1,max(times_basis)), norder=4, nbasis=knot);
Supi_obj <- smooth.basis(argvals = c(1:time_span), y = t(fFDA[,a1:a2]), fdParobj = basis);

# classes: 0 undefined, 2 healthy, 1 ill
dev.new();
plot(Supi_obj$fd, col=c("red", "green", "gray")[fFDA[,1]]);
title(main="Smoothed,original classes", sub = subtitle);
legend("topleft", fancest, fill=c("red", "green")[fFDA[,1]]);


#############################################
# DERIVATIVES
# D1 => Velocity
# D2 => Acceleration

# first derivative allover
der1 <- deriv.fd(Supi_obj$fd,1);
dev.new();
plot(der1,col=c("red", "green", "grey")[fFDA[,1]]);
#legend("topleft", fancest, fill=c("red", "green")[fFDA[,1]]);
title(main="First Derivative", sub = subtitle);

# first derivative of class "ill"
derIll1 <- deriv.fd(Supi_obj$fd[which(fFDA[,1]%in% c(1))],1)
dev.new();
plot(derIll1, col="red");

# first derivative of class "helathy"
derHeal1 <- deriv.fd(Supi_obj$fd[which(fFDA[,1]%in% c(2))],1)
dev.new();
plot(derHeal1, col="green");

# second derivative allover
der2 <- deriv.fd(Supi_obj$fd,2);
dev.new();
plot(der2,col=c("red", "green", "grey")[fFDA[,1]]);
#legend("topleft", fancest, fill=c("red", "green")[fFDA[,1]]);
```

```
title(main="Second Derivative", sub = subtitle);

# second derivative of class "ill"
derIll2 <- deriv.fd(Supi_obj$fd[which(fFDA[,1]%in% c(1))],2)
dev.new();
plot(derIll2, col="red");

# second derivative of class "healthy"
derHeal2 <- deriv.fd(Supi_obj$fd[which(fFDA[,1]%in% c(2))],2)
dev.new();
plot(derHeal2, col="green");


#########################################
# T-TEST
#
# pointwise t-test to see if groups of functions differ

# initial functions
dev.new();
tperm.fd(Supi_obj$fd[which(fFDA[,1]%in% c(1))], Supi_obj$fd[which(fFDA[,1]%in% c(2))
    ]);
# first derivative
dev.new();
tperm.fd(derIll1, derHeal1);
# second derivative
dev.new();
tperm.fd(derIll2, derHeal2);


############################################
# BOXPLOT
dev.new();
boxplot(Supi_obj$fd, method="MBD", factor=0.5);

dev.new();
boxplot(der1, method="MBD", factor=0.5);

#############################################
# MEAN, STANDARD DEVIATION

# MEAN
mAll <- mean.fd(Supi_obj$fd); # allover mean
mIll <- mean.fd(Supi_obj$fd[which(fFDA[,1]%in% c(1))]); # mean of class "ill"
mHeal <- mean.fd(Supi_obj$fd[which(fFDA[,1]%in% c(2))]); # mean of class "healthy"

# plot all three means
dev.new();
plot(mAll, ylim=c(300,1500));
lines(mIll, col="red");
lines(mHeal, col="green");
```

```
title(main="Means", sub="means of all (black), ill (red) and healthy (green) tissues");


dev.new();
plot(mIll$coefs, mHeal$coefs, type="l");
title(main="Compared means", sub="mean of ill (x) against mean of healthy (y) tissue");

# plot all functions plus the allover mean
dev.new();
colo <- switch(toString(fFDA[1,1]), "0" = "grey", "2" = "green", "1" = "red");
plot(c(1:(dim(fFDA)[2]-1)), fFDA[1,2:(dim(fFDA)[2])], type = "l", xlab = "time", ylab = "f(time)", col = colo, xlim=c(1,dim(rawFDA)[2]),ylim=c(250,1750));
for(i in c(2:dim(fFDA)[1])){
  colo <- "black";
  colo <- switch(toString(fFDA[i,1]), "0" = "grey", "2" = "green", "1" = "red");
  lines(fFDA[i,2:(dim(fFDA)[2])], lwd = 1, col = colo);
}
lines(mAll, col="black");
title(main="Original plus Mean", sub="all curves of ill (red) and healthy (green) tissue, plus the mean (black) of all curves");

# plot means of first derivatives
mAllD1 <- mean.fd(der1);
mIllD1 <- mean.fd(derIll1);
mHealD1 <- mean.fd(derHeal1);
dev.new();
plot(mAllD1, ylim=c(300,1500));
lines(mIllD1, col="red");
lines(mHealD1, col="green");
title(main="Means of first Derivative", sub="means of all (black), ill (red) and healthy (green) tissues");

# STANDARD DEVIATION
sdAll <- sd.fd(Supi_obj$fd); #sd allover
sdIll <- sd.fd(Supi_obj$fd[which(fFDA[,1]%in% c(1))]); # sd for class "ill"
sdHeal <- sd.fd(Supi_obj$fd[which(fFDA[,1]%in% c(2))]); # sd for class "healthy"

dev.new();
plot(sdAll, ylim=c(0,400));
lines(sdIll, col="red");
lines(sdHeal, col="green");
title(main="Standard Deviations", sub="sd of all (black), ill (red) and healthy (green) tissues");

# sd for the first derivatives
sdAllD1 <- sd.fd(der1);
sdIllD1 <- sd.fd(derIll1);
sdHealD1 <- sd.fd(derHeal1);

```

```
dev.new();
plot(sdAllD1, ylim=c(0,400));
lines(sdIllD1, col="red");
lines(sdHealD1, col="green");
title(main="Standard Deviations of the first Derivative", sub="sd of all (black), ill (red) and healthy (green) tissues");

#############################################
# # PCA

# pcaAll <- pca.fd(Supi_obj$fd);
# plot.pca.fd(pcaAll);

# # strong first component, second at 0.7%
# # -> nearly all informations stored in the first component
# # if it would be points instead of functions, one dimension would be enough to show differences

# # pcaIll <- pca.fd(Supi_obj$fd[which(fFDA[,1]%in% c(1))]);
# # plot.pca.fd(pcaIll);

# # pcaHeal <- pca.fd(Supi_obj$fd[which(fFDA[,1]%in% c(2))]);
# # plot.pca.fd(pcaHeal);

#####################################################
# MAXIMUM
# eval.fd(sequence,fda-object);

se <- seq(2,30, length=2801); # 28 times x plus 1, x=100
fineFDA <- eval.fd(se,derAllover1); # 2 to 30 to cut infinite ends
maxs <- array(NA,2);
for(c in c(1:dim(fineFDA)[2])){
  cur <- fineFDA[,c];
  m <- max(cur); # gives max
  i <- which(cur %in% m); # gives coord for max
  maxs <- rbind(maxs, c(i,m));
}
maxs <- maxs[2:dim(maxs)[1],];
plot(maxs);


# classified curves
#se <- seq(2,30, length=281); # 28*x+1, x=100
fineIll <- eval.fd(se,derIll1); # 2 to 30 to cut infinite ends
maxIll <- array(NA,2);
for(c in c(1:dim(fineIll)[2])){
  cur <- fineIll[,c];
  m <- max(cur); # gives max
  i <- which(cur %in% m); # gives coord for max
  maxIll <- rbind(maxIll, c(i,m));
}
```

```r
maxIll <- maxIll[2:dim(maxIll)[1],];
points(maxIll, col="red");

fineHeal <- eval.fd(se,derHeal1); # 2 to 30 to cut infinite ends
maxHeal <- array(NA,2);
for(c in c(1:dim(fineHeal)[2])){
  cur <- fineHeal[,c];
  m <- max(cur); # gives max
  i <- which(cur %in% m); # gives coord for max
  maxHeal <- rbind(maxHeal, c(i,m));
}
maxHeal <- maxHeal[2:dim(maxHeal)[1],];
points(maxHeal, col="green");

# plot maximum values of classified functions only
dev.new();
plot(maxIll, col="red", ylim=c(0,200), xlim=c(700, 1200), pch=19);
points(maxHeal, col="green", pch=19);

###############################################
# ANOVA

bas <- create.bspline.basis(rangeval=c(min(times_basis)+1,max(times_basis)), norder=4, nbasis=31);
crosspro <- inprod(bas, bas);
# select groups (0=undetermined, 1=ill, 2=healthy)
gr <- c(0,1,2); #c(1,2) or c(0,1) or ...
ob <- t(cFDA[,2:dim(cFDA)[2]])[,which(cFDA[,1]%in% gr)];
lab <- cFDA[which(cFDA[,1]%in% gr)];
fanova3 <- fanova.tests(group.label = lab, test = "FP",
           params = list(paramFP = list(B.FP = 1000, basis = "own",
                                        own.basis = ob,
                                        own.cross.prod.mat = crosspro
       )));
summary(fanova3);

###############################################
# details about used R and packages           #
###############################################
sessionInfo()
# R version 4.2.1 (2022-06-23 ucrt)
# Platform: x86_64-w64-mingw32/x64 (64-bit)
# Running under: Windows 10 x64 (build 19044)

# Matrix products: default

# locale:
# [1] LC_COLLATE=German_Germany.utf8  LC_CTYPE=German_Germany.utf8
# [3] LC_MONETARY=German_Germany.utf8 LC_NUMERIC=C
# [5] LC_TIME=German_Germany.utf8

# attached base packages:
# [1] splines   stats     graphics  grDevices utils     datasets  methods
# [8] base

# other attached packages:
#  [1] fdANOVA_0.1.2    fda.usc_2.1.0    mgcv_1.8-40      nlme_3.1-157
#  [5] plotly_4.10.1    funHDDC_2.3.1    funFEM_1.2       elasticnet_1.3
#  [9] lars_1.3         ggplot2_3.3.6    dplyr_1.0.10     fda_6.0.5
# [13] deSolve_1.33     fds_1.8          RCurl_1.98-1.9   rainbow_3.7
# [17] pcaPP_2.0-2      MASS_7.3-57      terra_1.6-7      sf_1.0-8
# [21] supercells_0.9.1 rgl_0.109.6      Rcpp_1.0.9       OpenImageR_1.2.5
# [25] abind_1.4-5      misc3d_0.9-1     oro.dicom_0.5.3

# loaded via a namespace (and not attached):
#  [1] bitops_1.0-7         doParallel_1.0.17    httr_1.4.4
#  [4] backports_1.4.1      Deriv_4.1.3          tools_4.2.1
#  [7] utf8_1.2.2           R6_2.5.1             KernSmooth_2.23-20
# [10] DBI_1.1.3            lazyeval_0.2.2       colorspace_2.0-3
# [13] withr_2.5.0          tidyselect_1.2.0     compiler_4.2.1
# [16] cli_3.4.1            microbenchmark_1.4.9 scales_1.2.1
# [19] classInt_0.4-7       mvtnorm_1.1-3        proxy_0.4-27
# [22] digest_0.6.29        tiff_0.1-11          base64enc_0.1-3
# [25] jpeg_0.1-9           pkgconfig_2.0.3      htmltools_0.5.3
# [28] fastmap_1.1.0        htmlwidgets_1.5.4    rlang_1.0.6
# [31] SuppDists_1.1-9.7    shiny_1.7.2          generics_0.1.3
# [34] jsonlite_1.8.0       mclust_5.4.10        magrittr_2.0.3
# [37] Matrix_1.4-1         munsell_0.5.0        fansi_1.0.3
# [40] lifecycle_1.0.3      grid_4.2.1           parallel_4.2.1
# [43] promises_1.2.0.1     lattice_0.20-45      knitr_1.40
# [46] pillar_1.8.1         tcltk_4.2.1          codetools_0.2-18
# [49] kSamples_1.2-9       magic_1.6-0          glue_1.6.2
# [52] doBy_4.6.15          data.table_1.14.6    png_0.1-7
# [55] vctrs_0.4.2          httpuv_1.6.5         foreach_1.5.2
# [58] gtable_0.3.1         purrr_0.3.5          tidyr_1.2.1
# [61] ks_1.13.5            xfun_0.33            mime_0.12
# [64] broom_1.0.2          xtable_1.8-4         pracma_2.4.2
# [67] e1071_1.7-11         later_1.3.0          class_7.3-20
# [70] viridisLite_0.4.1    tibble_3.1.8         iterators_1.0.14
# [73] units_0.8-0          cluster_2.1.3        ellipsis_0.3.2
# [76] hdrcde_3.4
```