Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE

A random forest approach for interval selection in functional regression

Rémi Servien (ORCID: 0000-0003-1270-1843)[1] | Nathalie Vialaneix (ORCID: 0000-0003-1156-0639)[2]

[1]INRAE, Univ Montpellier, LBE, 102 Avenue des Etangs, 11100, Narbonne, France

[2]Université de Toulouse, INRAE, UR MIAT, Castanet-Tolosan, France

Correspondence
Nathalie Vialaneix, INRAE Toulouse, 0875 MIAT, CS 52627, 31326 Castanet Tolosan cedex, France.
Email: nathalie.vialaneix@inrae.fr

Abstract

In this paper, we focus on the problem of variable selection in a functional regression framework. This question is motivated by practical applications in the field of agronomy where identifying temporal periods during which weather measurements impact the most the obtained yield is critical to guide agriculture practices in a changing environment. From a methodological point of view, our goal is to identify consecutive measurement points in the definition domain of the functional predictors, which correspond to the most important intervals for the prediction of a numeric output from the functional variables. We propose an approach based on the versatile random forest method that benefits from its good performances for variable selection and prediction. Our method builds in three steps (interval creation, summary, and selection). Different variants for each of the steps are proposed and compared on both simulated and real-life datasets. The performances of our method compared to alternative approaches highlight its usefulness to select relevant intervals while maintaining good prediction capabilities. All variants of our method are available in the R package **SISIR**.

KEYWORDS
functional data analysis, interval selection, random forest, agronomy

1 | INTRODUCTION

1.1 | A variable selection problem for non-parametric functional regression

The present article focuses on functional regression problems, in which a real random variable Y is predicted from a functional predictor X that takes values in a functional space, $\mathcal{H}$ (*e.g.*, $L^2([0,1])$). The range of X is supposed to be a bounded interval $[a,b]$ of $\mathbb{R}$ and we are given n i.i.d. observations of the random pair (X,Y), $(X_i,y_i)_{i=1,\dots,n}$. This settings have been widely studied over the past years under the name of Functional Data Analysis (FDA), especially since the seminal works of [51, 52, 20]. It covers a wide range of applications including spectrometric data [20, 57], handwriting recognition [6, 64], or weather data [47, 28].

Here, we focus on the non parametric setting

$$Y = F(X) + \epsilon, \tag{1}$$

where F is an unknown function and ϵ is an error term. Non-parametric estimation in FDA has been addressed by a variety of methods including nonparametric kernel estimate [20], k-nearest neighbors [12], SVM [57, 29], multi-layer perceptron [56, 21], or random forest [45, 27]. They have often been found to have a higher prediction accuracy and are more suited to model complex phenomena. However, the form of F is often not easily interpretable and, contrary to more simple linear models, does not straightforwardly identify which specific part of the functional predictor bears most of the variability of the output. Even in the case of the functional linear model

$$Y = \langle X, \beta \rangle_{\mathcal{H}} + \epsilon,$$

the authors of [32] comment on the fact that coefficient curves β that have a "structure" are often easier to interpret. In their case, they propose to structure β under the form of regions in the range of X where $\beta(t) = 0$ and regions where $\beta(t)$ is exactly linear. From a practical point of view, the latter regions are those where the values of X have an impact on the values of Y and can help the user interpret the predictive model. Similarly, taking an example from agronomy (see Section 3), if X is a

weather time-series and Y is the field yield at the end of the year, these regions are the period of the year during which the weather has a strong impact on the yield: This information is thus critical to inform decision-making in agronomy.

From a formal point of view, the current article thus aims at simultaneously finding an accurate solution for the nonparametric model of Equation (1) while selecting a subset $\mathcal{S} \subset [a,b]$ of important observation points in the range of X such that if X^a and X^b are two observations of X with $X^a(t) = X^b(t)$ $\forall\, t \in \mathcal{S}$ and $X^a(t^*) \neq X^b(t^*)\ \forall\, t^* \notin \mathcal{S}$, we have $F(X^a) = F(X^b)$. In other words, the observation points in $\mathcal{S}$ are the only points that change the prediction $F(X)$. To ease interpretation and as discussed before, it is additionally desirable that $\mathcal{S}$ is searched under the form of unions of disjoint intervals

$$\mathcal{S} = \cup_{k=1}^{K^*} I_k^*,$$

with $I_k^* = [a_k, b_k] \subset [a,b]$, $b_k > a_k$, and $\forall\, k \neq k'$, $I_k^* \cap I_{k'}^* = \emptyset$. In the sequel, such intervals will be termed "impact intervals".

1.2 | Related work

Some previous works have already addressed variable selection in functional prediction models. Some of these proposals focus on selecting a finite number of isolated observation points in $[a,b]$ (*e.g.*, $\mathcal{S} = \{\tau_j\}_{j=1,\dots,K}$ that are called "sensitive points" or "points of impact"): In the linear setting, [3] use a ℓ_1-penalty, [43] focus on selecting a unique "sensitive" time point assuming fractal behavior for X, and [35] propose a method based on a local selection procedure performed separately from the regression model estimation. In the nonparametric setting, [19] design an influential measure for nonparametric regression models based on cross validation.

A limited number of articles have addressed this question in a setting that allows to select more interpretable intervals as we seek: [55] proposes a group-Lasso approach in a linear setting but her method requires that the range of X, $[a,b]$, is *a priori* partitioned into intervals, which is a nontrivial task with a potentially high impact on the relevance of the results. Also in the linear setting, [46] propose a linear classification method based on a linear discriminant analysis that has a double penalty: A ℓ_1-penalty is used to impose sparsity on the functional regression coefficient and a ℓ_2-penalty is used on its derivative to impose regularity. However, this approach is restricted to identify very smooth regression coefficients and to the linear setting. Also restricted to the linear setting and building on the smoothness of the regression coefficient, [32] propose the FLiRTI method that combines basis representation to variable selection with the Lasso. Finally, [28] propose a very different course of action based on a sparse Bayesian functional model where an *a priori* distribution is defined for the intervals. However, this approach, which uses a Gibbs sampler, is computationally very demanding and not adapted to the case of large datasets.

As far as we can tell, only two previous articles have addressed this question in a non-linear functional regression setting: [47] developed a semi-parametric model, the penalized multidimensional sliced inverse regression. In this approach, the intervals are defined automatically using a sequential and greedy approach, without any *a priori* knowledge, and they are then selected thanks to the inclusion of a group-Lasso-like penalty in the model. However, while efficient, this approach requires tuning several hyper-parameters (dimension of the projection space, regularization parameters, and parameters of the greedy aggregation) which makes the results sensitive to their choices and the method potentially intensive to run when a meticulous tuning is performed. Also, [13] proposed an optimization framework to learn relevant intervals in a so-called "optimal tree" approach, which uses Lasso selection. However, the computational needs of their approach seem to make it suited to fit only one tree, which is well-known to lack robustness as compared to RF [14]. In addition, it requires the tuning of several sparsity and regularity hyper-parameters (in addition to the number of intervals), which potentially makes its results also sensitive to their choices.

Our approach builds on this latter work, proposing a data driven approach for functional data regression that uses random forest and is able to simultaneously estimate a prediction model (in a nonparametric way) and detect impact intervals in the range of X.

1.3 | Description of our contribution

Our contribution is the proposition of an extension of RF to functional regression that is interpretable in terms of important intervals. More precisely,

1. we propose a new RF-based functional regression method. It has the simplicity of RF models, requiring no or only a few hyper-parameters tuning;
2. we embed the automatic definition and selection of impact intervals in this method;
3. we empirically demonstrate on simulations and real-life datasets that our approach achieves a better trade-off between prediction accuracy and relevance of the selected intervals than alternative approaches.

In addition, our method, termed SFCB (Selection Forest for funCtion Based predictions), is implemented in the R package **SISIR** (version $\geq$ 0.2.0), released on CRAN.

Note that, in previously cited works of the literature, the authors either suppose that the observations of the functional

variable, $(X_i)_{i=1,\dots,n}$, are perfectly known or that they are themselves perfectly or imperfectly observed at different points in $[a,b]$. In the present contribution, we place ourselves in the realistic setting where X_i are all measured at points $t_1,\dots,t_p$ in $[a,b]$ (measurement points are common to all observations but might be irregularly distributed in $[a,b]$; X_i is observed without error at $(t_j)_{j=1,\dots,p}$). We will denote $\mathbf{x}_i$ the p dimensional vector $(X_i(t_1),\dots,X_i(t_p))^\top$.

The article is organized as follows: Section 2 presents the method and its different variations. Section 3 describes the datasets used for the evaluation, both simulated and from real-world problems. Finally, results and comparisons with alternative approaches are provided in Section 4.

2 | METHOD

2.1 | A new random forest for functional data

Random forest for functional data in the literature

In the last decade, with the increase of real-time measurements, RF have been adapted to handle functional predictors, and especially to longitudinal predictors (*e.g.*, time series). A first direction is to extend RF to similarity data embedding the resemblance between time series, such as the work on Fréchet forest [16] or on proximity forest [40] (restricted to classification). In these approaches, the direct relation between the functional predictors and the prediction model is lost, because the prediction model is trained from similarities: The method is thus impossible to interpret in terms of relevant intervals in the predictor range. Other methods are based on dictionaries or symbolic representations of the time series such as the work of [58] (BOSS) then extended in TS-CHIEF [59] (which combines different types of splits including dictionary-based splits), or the work of [9] that uses symbolic representation of longitudinal data. Again, due to the symbolic intermediate representation of the data, these methods are hard to interpret in terms of relevant intervals in the predictor range.

Finally, other longitudinal data RF-based classifiers exist that are built on interval techniques. This is the case, for instance, of Time Series Forest [18] and its extension [44]. In these methods, intervals must be defined *a priori* and they are summarized using simple statistics (such as means and standard deviations of the predictors on this interval) and then used as new inputs for the RF. These approaches have appealing properties such as their interpretability in terms of intervals or their simplicity but they can not handle the selection of all possible intervals and are thus restricted to certain *a priori* intervals for which the user must have certain knowledge. Another interval procedure that does not require the *a priori* definition of intervals is RISE [39]: In this method, an interval is randomly sampled (uniform sampling of the start and end positions of the interval over the predictor range) and a tree is then built only on this interval. However, in this approach, the notion of the importance of an interval is lost because, by definition of the aggregation step in RF, all randomly sampled intervals contribute similarly to the prediction. Similarly, [45] use an interval-based random forest by randomly sampling a different set of intervals for each tree. Again, importance is computed for each observation points.

Estimating F with random forest

Our functional random forest method builds on the proposal of [18, 44] but provides a data-driven method to define intervals, which is thus well adapted to the selection goal, a a more relevant definition of summary computation, better adapted to the regression purpose. More precisely, given observations $(\mathbf{x}_i, y_i)_{i=1,\dots,n}$,

1. **Step 1**: $[a,b]$ is partitioned into K data-driven disjoint intervals $\mathcal{I} = \{I_k\}_{k=1,\dots,K}$, which reduces to a partition of $\{t_1,\dots,t_p\}$ under contiguity constraint over the $(t_j)_j$. Different solutions to find relevant partitions are discussed in Section 2.2;
2. **Step 2**: X_i is summarized in I_k by L numeric summaries, $\left(\tilde{x}_{i,k}^{(l)}\right)_{l=1,\dots,L}$ (with $L = 1$ or 2 in our propositions) for every i and every interval k. Different solutions to summarize functional data in a given interval are discussed in Section 2.3 and, given $\mathcal{I}$, we will denote by $\mathcal{T}_\mathcal{I}$ the transformation that maps a given (X_i, y_i) (or sometimes just a given X_i; see Section 2.3 for details) to the $K \times L$ dimensional vector $\left(\tilde{x}_{i,k}^{(l)}\right)_{k=1,\dots,K,\ l=1,\dots,L}$.

Similarly to [18, 44], F is then estimated by

$$\hat{F} = \mathrm{RF}^* \circ \mathcal{T}_\mathcal{I},$$

where RF^* is the random forest trained with training data $(\mathcal{T}_\mathcal{I}(X_i, y_i), y_i)_i$.

Note that $\mathrm{RF}^* \circ \mathcal{T}_\mathcal{I}$ could be seen as an under-efficient random forest compared to the one that could have been trained directly using $(\mathbf{x}_i, y_i)$ as the training dataset. However, it has two main advantages:

- in functional regression models, it is frequent that the value of the functional predictors at a given observation point are not directly comparable and that the functional predictor should allow some local translations. This is illustrated, for instance, by the very recent work of [41], in the field of agronomy, which showed that measurements of weather data at on a fine temporal grid are less relevant for the prediction task than summarized data at interval levels. Using

information on autocorrelation of X along its range is thus a way to summarize the information in a way that is more relevant for prediction purpose;
- predictors of RF^* are values directly associated to interval I_k. Thus, selecting variables in this setting is equivalent to looking for the impact interval set $\mathcal{S}$.

Searching for $\mathcal{S}$ to improve interpretability

As discussed at the end of the previous section, a simple way to define $\mathcal{S}$ would be performing a variable selection method using $(\tilde{\mathbf{x}}_{ik}^{(l)}, y_i)_{i=1,\dots,n}$ to select a few number of predictors among $(\tilde{\mathbf{x}}_{.k}^{(l)})_{k=1,\dots,K,\, l=1\dots,L}$. Denoting $\mathcal{VS}$ the variable selection procedure built from $(\tilde{\mathbf{x}}_{ik}^{(l)}, y_i)_{i=1,\dots,n}$, $\mathcal{VS}$ maps $(\tilde{\mathbf{x}}_{ik}^{(l)})_{k=1,\dots,K,\, l=1\dots,L}$ to the subset $(\tilde{\mathbf{x}}_{ik}^{(l)})_{(k,l)\in S}$, where S does not depend on i. We can then derive $\mathcal{S}$ as:

$$I_k \in \mathcal{S} \qquad \Leftrightarrow \qquad (k,l) \in S \text{ for at least one } l \in \{1,\dots,L\}.$$

Various choices for $\mathcal{VS}$ are discussed in Section 2.4.

Using variable selection, another estimation of F can be obtained based on the new restricted set of predictors, $(\mathcal{VS}(\mathcal{T}_{\mathcal{I}}(X_i, y_i))_i$:

$$\hat{\tilde{F}} = \mathrm{RF}^{**} \circ \mathcal{VS} \circ \mathcal{T}_{\mathcal{I}},$$

where RF^{**} is the random forest trained with training data $(\mathcal{VS}(\mathcal{T}_{\mathcal{I}}(X_i, y_i)), y_i)_i$.

Overview of the method and implementation

Figure 1 illustrates the different steps of the estimation procedure described in this section. The first box corresponds to the definition of $\mathcal{I}$, the second to $\mathcal{T}_{\mathcal{I}}$ and the third to $\mathcal{VS}$. Our method is implemented in the R package **SISIR** (in versions $\geq$ 0.2.0 of the package) in the function `sfcb`. The package also includes diagnostic quality criteria (including the ones used in the present article; see Section 3) and plots.

Overall, the whole procedure only requires the choice of the different options for the three steps of the methods (relevant choices for different goals are discussed in the next sections). Contrary to the work of [13], which targets a similar goal, it is thus simpler and faster to use (in addition to providing flexibility and robustness of random forest compared to regression trees) but at the cost of having the different steps of the approach fitted independently instead of as a whole.

2.2 | Building a relevant hierarchy of partitions

The first step of our method consists in partitioning $[a, b]$ into a relevant set of K intervals, $\mathcal{I} = \{I_k\}_{k=1,\dots,K}$. This is equivalent to partitioning $\{t_1, \dots, t_p\}$ into K groups constrained to be contiguous. To achieve this, we used clustering methods based on values of $(X(t_j))_{j=1,\dots,p}$ so as to group together time points that have strongly associated values for the functional predictor X. This type of approach is usually named "clustering of variables" and we used versions constrained by contiguity of the time points $\{t_1, \dots, t_p\}$. Hence, in the following, we do not differentiate the partition of $\{t_1, \dots, t_p\}$ (which is the ultimate goal) from the clustering of $(X(t_j))_{j=1,\dots,p}$, (which is the method used to achieve this goal).

In addition, to allow more flexibility in the choice of K in a simple way, we investigated hierarchical methods, which provide a hierarchy of partitions, $\{\mathcal{I}_l\}_{l=1,\dots,p}$, where $\mathcal{I}_1$ is the naive partition made of the p singletons $\{t_j\}$, $\mathcal{I}_p$ is the partition made of the unique cluster $\{t_1, \dots, t_p\}$, and the difference between $\mathcal{I}_l$ and $\mathcal{I}_{l+1}$ reduces to the merge of a single pair of clusters.

*A constrained based Ward's hierarchical clustering based on correlation (**adjclust**)*

One of the most used method to provide a hierarchy of clusters is "hierarchical clustering", which is based on a linkage, *e.g.*, the definition of a distance between clusters that is deduced from the distance of pairs of objects. One of the most used linkage is the so-called "Ward's linkage" [63]: It aims at minimizing the decrease in variance between clusters when merging two clusters. In its standard version, Ward's linkage computed the variance induced by the Euclidean distance. However, clustering $(X(t_j))_{j=1,\dots,p}$ based on their Euclidean distance is not relevant since we are more interested in clusters with strong correlations rather than in clusters with close values. Kernel hierarchical clustering [49, 1] extends Ward's hierarchical clustering to the case where the distance is induced by an arbitrary dot product given under the form of a *kernel* matrix [5]. Here, we proposed to use the constrained version of this method, as described in [2] (and implemented in the R package **adjclust**) and to use the empirical variance matrix

$$\Sigma \qquad \text{st} \qquad \Sigma_{jj'} = \frac{1}{n}\sum_i (\mathbf{x}_{ij} - \bar{\mathbf{x}}_j)(\mathbf{x}_{ij'} - \bar{\mathbf{x}}_{j'}) \qquad (2)$$

as the kernel. The method is detailed in Algorithm 1 of the appendix.

*A constrained-based hierarchical clustering based on PCA (**cclustofvar**)*

[17] also proposes a hierarchical clustering approach that is based on a correlation criterion through a PCA decomposition. More precisely, starting from the naive partition made of the p singletons $\{t_j\}$, two clusters are merged if they are the ones minimizing the overall loss in homogeneity, where the homogeneity of a cluster is defined as the first eigenvalue of the PCA of the variables $(X(t_j))_j$ included in this cluster. We used this approach, adding a contiguity constraint (the minimization is performed over any pair of two contiguous clusters only, where

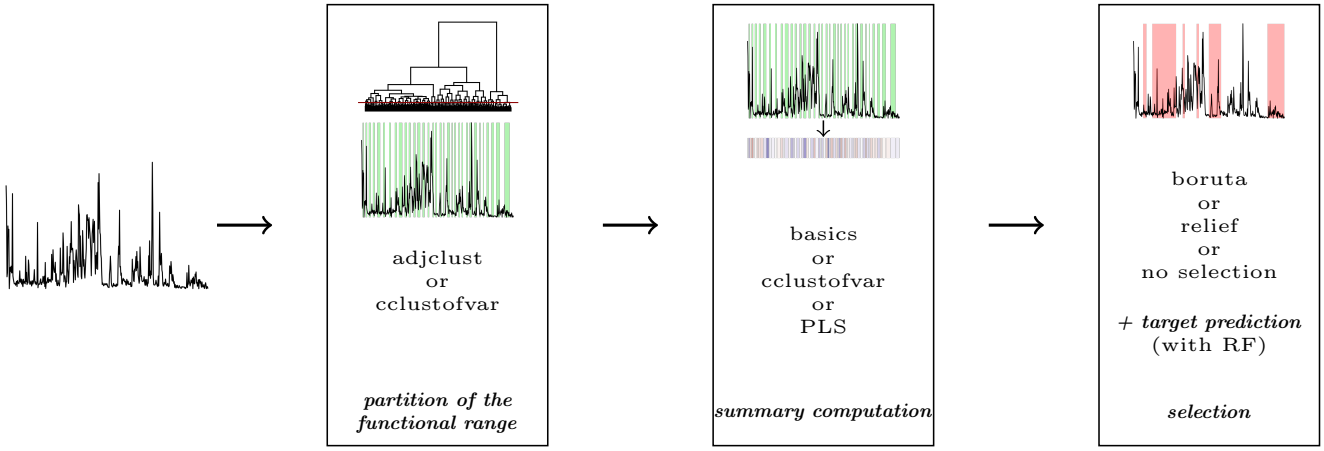

FIGURE 1 Overview of the different steps and variants of SFCB.

the contiguity is based on the values of t_j). The detailed method is given in Algorithm 2 of the appendix.

Note that, in addition to providing a hierarchy of partitions, the later approach offers a simple and interpretable summary of the intervals, as a by-product. Indeed, as described in [17], the homogeneity of an interval I_k of the partition $\mathcal{I}$ can be rewritten as

$$\mathcal{H}(I_k) = \sum_{j:\ t_j \in I_k} \mathrm{Cor}^2((\mathbf{x}_{ij})_i, \mathbf{c}_k), \tag{3}$$

where Cor^2 is the squared Pearson correlation and $\mathbf{c}_k \in \mathbb{R}^n$ is the first eigenvector of the PCA, which can be used as a summary of I_k.

Choice of the best partition

Once the hierarchy of partitions is obtained, we chose to retain only one partition, $\mathcal{I} = \{I_k\}_{k=1,\dots,K}$. This could be chosen using heuristics dedicated to the choice of the number of clusters in hierarchical clustering like inspection of the dendrogram[†], the broken stick heuristic [10], the slope heuristic [24, 4], the gap statistics [62] (unfortunately computationally intensive), etc. Based on our experiments, we advise choosing a level of the hierarchy where the number of clusters is not too low, to allow for sufficient precision in the detection of relevant intervals.

2.3 | Summarizing the predictors on intervals

The second step of our method consists in summarizing the information of $(\mathbf{x}_i)_{i=1,\dots,n}$ over each interval I_k ($k = 1,\dots,K$), hence, defining the transformation $\mathcal{T}_\mathcal{I}$. This approach has already been followed by [45], who used the mean, by [18], who used the mean, the standard deviation, and the slope as summaries, and by [44] who used catch22, a set of 22 features considered as the canonical time-series characteristics. The higher the number of summaries for a given interval is, the more complete and precise the information on the signal is. However, a high number of summaries brings the question of interpretability back, since many different intervals could be found important in the prediction for different summaries. This led us to consider different versions of this approach, all restricted to a small number $L \in \{1, 2\}$ of summaries.

A natural choice to summarize the $(\mathbf{x}_{ij})_{j:\ t_j \in I_k}$ for every interval k would be a linear combination of these values, leading to so-called "oblique splits" [11, 30]. Whilst very flexible and better adapted to the prediction of Y, these approaches often lead to difficult computational problems that are not suited to handle complex large-dimensional questions, such as functional regression. Hence, based on *a priori* grouping of variables, [48] propose to use (regularized) linear discriminant analysis (LDA) to automatically define such oblique splits at a low computational cost (see also [50], who use canonical correlation analysis for multiple outputs). Based on these ideas, we proposed 3 methods to summarize the variables, all fast to compute, and one using the information provided by the target, similarly to [48].

*Basic unsupervised summaries (**basics**)*

Similarly to [18], our first strategy consisted of simply using the mean and standard deviation as input of our RF. More precisely, the original functional observations $(\mathbf{x}_i)_i$ are transformed using the following mapping:

$$\mathcal{T}_\mathcal{I}:\ \mathbf{x}_i \in \mathbb{R}^p \longrightarrow (\tilde{\mathbf{x}}_{i,k}^{(l)})_{l=1,2,\ k=1\dots,K} \in \mathbb{R}^{2K}$$

[†] Note, however, that due to the contiguity constraints, the obtained dendrogram might include reversals and be harder than usual to read [53].

with

$$\tilde{\mathbf{x}}_{i,k}^{(1)} = \frac{1}{\#I_k} \sum_{j:\ t_j \in I_k} x_{ij} \quad \text{and} \quad \tilde{\mathbf{x}}_{i,k}^{(2)} = \sqrt{\frac{1}{\#I_k} \sum_{j:\ t_j \in I_k} \left(x_{ij} - \tilde{\mathbf{x}}_{i,k}^{(1)} \right)^2}.$$

Note that, contrary to [18] and [44], we did not include the slope or more sophisticated characteristics in our summaries to maintain a good ability to interpret the model.

*A composite unsupervised summary based on PCA (**cclustofvar**)*
As mentioned in Section 2.2, the constrained clustering of variable derived from [17] offers a natural composite summary of a given interval (cluster) I_k simply setting $\tilde{\mathbf{x}}_{i,k}^{(1)} = \mathbf{c}_{ki}$ where $\mathbf{c}_k$ is the $\mathbb{R}^n$ vector of Equation (3).

*A supervised summary based on PLS (**pls**).*
Similarly to [48], we also tested a supervised summary using the target Y. To avoid the need for regularization (or penalization) and subsequent tuning of the regularization parameter in large dimensions (when the number of observation points in I_k is close to or larger than n for instance), we replaced LDA with PLS (Partial Least Squares; [65]). $(\tilde{\mathbf{x}}_{i,k}^{(l)})_{l=1,\dots,L}$ thus correspond to the first L-*th* PLS scores ($L = 1$ in our implementation) of the PLS regression of Y on X.

2.4 | Interval selection

Variable importance in random forest is usually thought as an efficient and simple way to select variables from a trained random forest (see an application in [31] in the field of network inference, which compares extremely well with other methods [42]). In short, given a random forest predictor, based on i.i.d. observations of p predictors, $\mathbf{z}_i^j$, and their corresponding values to predict $y_i \in \mathbb{R}$, the importance of the j-th variable is defined as the mean decrease in (out-of-bag) accuracy in the prediction of y_i when replacing $\mathbf{z}_i^j$ by a permuted version of this predictor (over i).

Hence, a simple way to define $\mathcal{S}$ would be to set

$$I_k \in \mathcal{S} \Leftrightarrow \text{Importance}_{\text{RF}^*}\left(\tilde{\mathbf{x}}_{.k}^{(l)}\right) \geq \tau \text{ for at least one } l \in \{1, \dots, L\},$$

for a given chose threshold $\tau > 0$.

However, this selection is rather basic and sensitive to the arbitrary choice of τ, when state-of-the-art feature selection methods have been developed during the past years [38]. We propose alternative choices to make the interval selection based on the modified features $\{\tilde{\mathbf{x}}_{ik}^{(l)}\}_{k,l}$. We restricted our choices to *embedded methods* (in which the selection is directly embedded in the random forest algorithm) or to *wrapper methods* (in which it is made in relation to the prediction purpose without being embedded into the random forest algorithm). The first has the advantage of being adapted at best to random forest and the second to being fast whilst accounting for the analysis objective. We did not consider *filter methods*, often found to be poorly efficient.

*A wrapper method based on Relief (**relief**)*
One of the most popular and efficient wrapper methods for binary classification problems is the Relief algorithm introduced by [34]. This score has been adapted to regression by [54], which is the variant that we used. In short, RReliefF computes a score for each summary variable $\tilde{\mathbf{x}}_{.,k}^{(l)}$, RReliefF$(k,l)$, which is based on estimates of the differences between $\tilde{\mathbf{x}}_{i,k}^{(l)}$ and $\tilde{\mathbf{x}}_{i',k}^{(l)}$ with respect to the differences between y_i and $y_{i'}$ for all $i' \in \mathcal{NN}(i)$ (set of nearest neighbors of i according to the whole vector $\tilde{\mathbf{x}}_i$): the RReliefF score, $\tilde{\mathbf{x}}_{i,k}^{(l)}$ increases when large differences in $\|y_i - y_{i'}\|$ result in large differences between $\tilde{\mathbf{x}}_{i,k}^{(l)}$ and $\tilde{\mathbf{x}}_{i',k}^{(l)}$. It can be seen as a way to score variables $\tilde{\mathbf{x}}_{.,k}^{(l)}$ based on the estimation of

$$\mathbb{P}\left(\text{distance}(\tilde{\mathbf{x}}_{i,k}^{(l)}, \tilde{\mathbf{x}}_{i',k}^{(l)}) \mid |y_i - y_{i'}|,\ i' \in \mathcal{NN}(i)\right).$$

Variants of this method correspond to different choices for the computation of the distance between observations of $\tilde{\mathbf{x}}_{.,k}^{(l)}$ or for the weighting scheme to account for differences in $y_i - y_{i'}$. The precise description of the used algorithm is given in Algorithm 3 of the appendix (and is the one implemented with the option RReliefFexpRank in the R package **CORElearn**). Obtained values for $\tilde{\mathbf{x}}_{.,k}^{(l)}$ are then ordered and a broken stick heuristic [23] is then used to select the variables associated to the highest scores.

*An embedded method based on knockoffs (**boruta**)*
Another approach that combines feature selection with random forest is proposed in [36] (R package **Boruta**), which uses shadow variables (similar to the knockoffs approach of [8]). In short, a random forest is learned on an extended datasets where each predictor is augmented with its randomized version. Selected variables are those that have a normalized importance score larger than their randomized copy. The method returns "confirmed" and "tentative" selective features, the latter corresponding to variables for which the importance score can not be significantly distinguished from their copy. We kept the two types of selected features in our final selection.

Another embedded method, **VSURF** [26, 25], was also considered in preliminary simulations but was finally dismissed due to its large computation time (despite preliminary relevant selection results).

Note that other embedded methods for variable selection with random forest also exist, such as the recurrent relative variable importance [61] or Vita [33] (R package **Vita**, based on a

testing framework). We chose not to test these methods based on the results of [60], which reported **Boruta** and **VSURF** as the best-performing methods for variable selection with RF.

3 | EXPERIMENTS

All simulation scripts and used datasets are available at https://doi.org/10.57745/KMH2GP (datasets) and https://forgemia.inra.fr/sfcb/simus_sfcb.git (scripts and results), with supplemental results and quality criteria.

In all experiments, the evaluation mostly focuses on the quality of the reconstruction of $\mathcal{S}$, even if we also provide assessment of the quality of the prediction through mean square error.

3.1 | Datasets

The different variants of the proposed method were tested both on a simulated dataset and on a real-life dataset and compared with two other alternatives, the linear bliss method [7] and the semi-parametric SISIR method [47], which, as far as we can tell, were the only ones that provided readily available implementations (in R packages).

3.1.1 | STICS dataset

A simulated dataset has been obtained from the meteorological data simulator WACSGen [22]. This simulator has been calibrated to reproduce the meteorological characteristics of Lleida (Spain) for the years 1981 and 1982. These data have then been used in the agriculture model STICS [15] and different parameters have been varied to simulate wheat culture under different scenarios. The STICS simulation model also generates intermediate computed meteorological measurements, known to be relevant for various agronomic questions. In particular, in our simulations, we used the evapotranspiration (ETM in mm/day), which is computed from four classical meteorological measurements (rainfall, min, max, and average temperature) together with other data like CO_2 input, yield humidity, wind speed, etc. Note that the same input data had also been used to test the SISIR approach in [47].

More precisely, the predictors consisted of 1,000 ETM time series $(x_i)_i$, observed at 444 time points during the crop period, $[\![287, 730]\!]$ (in days). We then generated a dataset according to the following simulation model:

$$\forall\, i = 1, \ldots, 1,000, \qquad y_i = \log\left(1 + |\langle x_i, \beta\rangle|\right) + \epsilon_i,$$

where β corresponds to a function with varying influence on (known) selected intervals $\beta : t \in [\![287, 730]\!] \mapsto 4 \times \mathbf{1}_{\{t\in[320,410]\}} + 2 \times \mathbf{1}_{\{t\in[500,550]\}} - \mathbf{1}_{\{t\in[680,730]\}}$ (hence, $\mathcal{S} = [320, 410] \cup [500, 550] \cup [680, 730]$). ϵ_i are i.i.d. errors drawn from $\mathcal{N}(0, 0.5)$. The final signal-to-noise ratio was equal to ~ 130.

3.1.2 | Truffle dataset

Second, we used the dataset also used to test the bliss method in [7]. The goal of the regression in this dataset is to predict black truffle production given some meteorological measurements. The black Périgord truffle (Tuber Melanosporum Vitt.) is one of the most famous and valuable edible mushrooms, because of its excellent aromatic and gustatory qualities.

Relevant meteorological data correspond to the maximum monthly air temperature (Tmax) and the monthly rainfall (P) (recorded), the monthly climatic water balance (PmPET), and the cumulated climatic water deficit (CWD) (computed). PET was calculated according to Hargreaves equation based upon a monthly latitude factor, mean monthly air temperature, and a coefficient for monthly relative humidity and was used to obtain PmPET (P minus PET). CWD was calculated as the sum of the positive water balances on a defined period. For each of these variables, we have 15 monthly measures from January of year Y to March of year $Y + 1$ for $Y \in [\![1925, 1949]\!]$ and $n = 25$ yearly yield of truffles which are given for the same periods. In addition, experts provided, for each meteorological data, important intervals. This was used as ground truth set $\mathcal{S}$ in our simulations.

For more details on this dataset, we refer the reader to [7]. In addition, part of this dataset has been released on the **SISIR** package (corresponding to the input variable P).

3.2 | Variation of performances in different simulated scenarios

SFCB was further tested in different situations corresponding to variations of the initial simulation setting based on the STICS dataset (see Section 3.1.1). More precisely, we assessed the robustness of the proposed method with respect to these variations:

- the number, length(s), and effect size(s) of the influence of the ground truth intervals (unless stated otherwise, the effect size is set to 4);
- the shape of the link function (log in the baseline scenario) and the shape of β (piecewise constant in the baseline scenario);
- the signal-to-noise ratio;
- n (number of samples) and p (number of observation points in the functional predictor definition domain).

The precise definition of these different simulation scenarios is given in Table 1. In all scenarios, except for the "Signal-to-noise ratio" setting, we set the variance of the error so as to keep the signal-to-noise ratio approximately constant.

3.3 | Comparison methodology

The different options for the three steps of our approach described in Section 2 were tested on both datasets. We selected results corresponding to a number of intervals, K, equal to $\sim 0.15p$ (currently the default value in the implementation) since preliminary tests showed that this almost always led to a good trade-off between relevant interval definition and good mean square error (MSE) for the selected intervals.

In addition, for SISIR, we used the dedicated R package **SISIR** with default pipeline (where hyper-parameters are set by a cross-validation strategy). The non-parametric part of the model was estimated using Support Vector Machines (SVM; implementation based on **e1071**), as in the original paper. For bliss, we also used the dedicated R package **bliss** with the number of intervals K optimized for a BIC criterion using the dedicated `BIC_model_choice` function.

To compare SFCB to SISIR and bliss, two target objectives were mainly assessed:

- the relevance of the selected intervals compared to the ground truth. To do so, we computed precision and recall of the selection compared to the ground truth set $\mathcal{S} = \{\beta(t) \neq 0 : t \in (t_j)_{j=1,\dots,p}\}$, where β is the ground truth function (explicit value given in Section 3.1.1 for the simulated dataset obtained from STICS variables and observed values at $(t_j)_{j=1,\dots,p}$ provided by experts and available in the truffle datasets at https://doi.org/10.57745/KMH2GP). In addition, a standard trade-off criterion was computed between these two quantities as

$$F_1 = 2\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$

- the goodness-of-fit of the predictions obtained from the models was also evaluated using MSE.

In addition, the computational time required by the different methods were also compared. Computations were performed on a personal desktop computer with Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz for the small truffle dataset and on the Genotoul-bioinfo cluster https://bioinfo.genotoul.fr/ for the simulated dataset. Information on used OS, as well as R and package versions is available in the source code repository.

4 | RESULTS

4.1 | Method comparison for the STICS dataset

First, SFCB is compared to **bliss** and SISIR on the simulated dataset based on STICS ETM. The options used for this comparison are adjclust for the functional range partition, PLS for summary computations and boruta for the selection.

Figure 2 respectively gives precision and recall and F_1 score of the obtained interval selection (ground truth corresponds to non-zero time points for β; see Section 3.1.1), as well as mean square error (MSE) and computational time of the method. **bliss** does not appear in this comparison because it doesn't select any interval (although its computation time was approximately one day).

Overall, SFCB exhibits a strong improvement of the precision of the selected intervals (compared to SISIR), at the cost of a slight increase of the MSE. Note that SFCB and SISIR MSE are, however, not fully comparable since SFCB MSE is the out-of-bag MSE, whereas reported SISIR MSE is the training error of the SVM learned on the intervals (thus probably more optimistic). Finally, SFCB has a low computation time (only $\sim$ 8 minutes here), which, contrary to SISIR, includes the training of the prediction method based on the intervals.

In addition, Figure 3 shows the selected intervals (compared to ground truth) for the two methods, as well as the importance obtained for the SFCB method trained with adjclust for the group definition and PLS for the summary computation but with no variable selection. Overall, the figure confirms the previous findings: SFCB is much more specific than SISIR and clearly identifies the three relevant intervals, with only a few mistakes. The variable importance shows a good agreement with selected intervals but might not be enough, alone, to identify the last interval (which indeed has a smaller influence in the prediction model), validating the relevance of using a variable selection approach on top of the method.

4.2 | Method comparison for the truffle dataset

Four different predictors (CWD, P, PmETP, and Tmax) are available for the truffle datasets, for which the three methods could be fully compared (in this case, **bliss** gave selection results). Figure 4 provides quality criteria obtained by the three methods for the four targets. For a fairer comparison SFCB was used with the same variant as for the simulated dataset (adjclust / PLS / boruta) but the results for another variant (found better here), clustofvar (partition) / clustofvar (summary) / relief (selection), is provided in our code repository.

TABLE 1 Description of evaluated variations of the simulation setting of Section 3.1.1.

Variation type	Scenario identifier	Description
Intervals $\mathcal{S}$	SC1	One large interval in the middle of the definition domain: $I_1^* = [420, 570]$
	SC2	One large interval at a border of the definition domain: $I_1^* = [580, 730]$
	SC3	One small interval in the middle of the definition domain: $I_1^* = [480, 510]$
	SC4	One small interval at a border of the definition domain: $I_1^* = [700, 730]$
	SC5	Four small intervals: $I_1^* = [300, 330]$, $I_2^* = [420, 450]$, $I_3^* = [550, 580]$, and $I_4^* = [660, 690]$
	SC6	Two intervals with opposite influence (positive and negative, respectively): $I_1^* = [420, 570]$ and $I_2^* = [510, 580]$
	SC7	Two intervals with different effect sizes (respectively 2 and 20): $I_1^* = [330, 400]$ and $I_2^* = [600, 670]$
Link function L ($y_i = L(\lvert\langle x, \beta\rangle\rvert) + \epsilon_i$)	linear	$L: z \to z$
	quadratic	$L: z \to z^2$
Signal-to-noise ratio		$10^{\{-2, -1, 1, 2\}}$ of the original signal-to-noise ratio
Shape of β	park1 and park2	see [46]
n		$n \in \{50, 100, 200, 500, 1000\}$
p		$p \in \{111, 222, 333, 444\}$

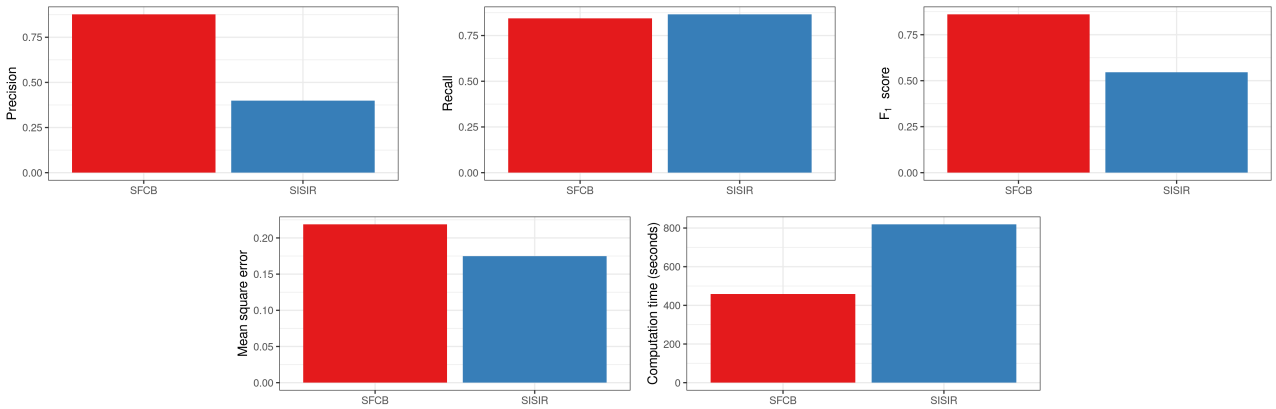

FIGURE 2 Quality criteria (precision, recall, F_1 score, mean square error and computation time) obtained by SFCB (red, left) and SISIR (blue, right) on the simulated dataset.

For most prediction tasks, **bliss** is providing a much better selection with respect to what was expected by experts. In particular, F_1 score is much stronger for CWD and PmETP predictors but slightly worse than that of SFCB for Tmax. Note that the bliss method has originally been tuned on this dataset so it is expected to perform particularly well. In all cases, SISIR obtained average results, comparable to SFCB (with the same precision and recall for CWD and PmETP) or worse.

For the MSE, SFCB ranked average to low, whereas bliss gave very bad prediction results. An explanation for this contradictory result is that the intervals selected as relevant by experts are always very short, except for Tmax that contains more than half of the initial time points) and probably not sufficient to achieve a good prediction result. This also explains the low overall F_1 score obtained by all methods except for bliss (the recall is always very high or equal to 1 but with a poor precision). In this situation, bliss confirms its stringent selection, matching the experts' expectations but not selecting other variables important for the prediction task. In all cases, SFCB obtained a good trade-off between accuracy of the prediction and relevance of the variable selection at a very low computation cost.

4.3 | Influence of the different options on the STICS dataset

The different variants for SFCB methods are also compared on the simulated dataset based on STICS ETM. Figure 5 gives precision & recall, F_1 score, and computation time for these variants. Overall, the selection based on boruta obtains better precision for all variants, resulting in an increased F_1 score. If not always the best, the scenarios with PLS for summary tend to give good results, whatever the other options. As

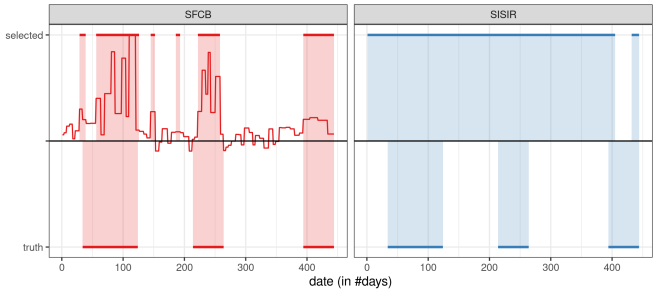

FIGURE 3 True (bottom) versus selected (top) intervals obtained by SFCB (left, red) and SISIR (right, blue) on the simulated dataset. The curve in the SFCB panel corresponds to the variable importance for the SFCB method with no variable selection (but the same choice for the groups and the summaries).

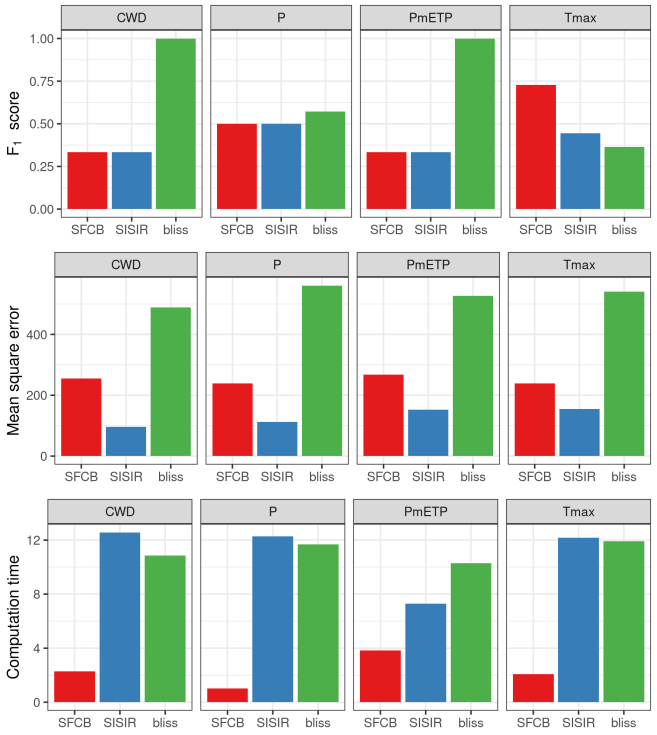

FIGURE 4 Quality criteria (F_1 score, mean square error, and computation time) obtained by SFCB (red, left) and SISIR (blue, middle), and bliss (green, right) on the simulated dataset.

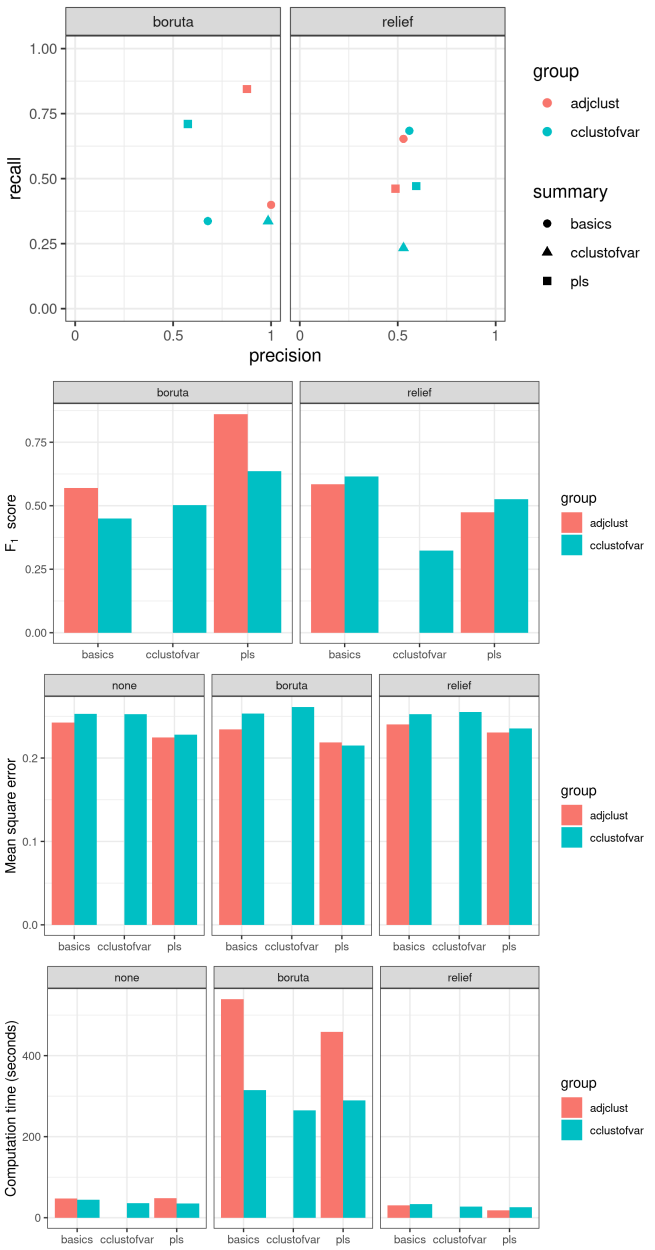

FIGURE 5 Precision & recall (first row), F_1 score (second row), mean square error (third row) and computational time (fourth row) obtained by the different variants of SFCB on the simulated dataset.

for the partition method, adjclust shows superior results compared to cclustofvar when combined with boruta but not when combined with relief.

MSE is rather stable over the different variants but adjclust / PLS / boruta stands out as one of the methods with the lowest error, indicating that MSE can be used as a mean to select the most relevant variant for SFCB in real-life situations. Also worth noting is the fact that selecting variables instead of just using the obtained summaries does not strongly deteriorate MSE in all situations, advocating again for variable selection instead of just relying on variable importance.

As expected, using boruta is more demanding in terms of computational effort. For large datasets, the use of relief can thus come as a benefit for selection. The difference in computation between adjclust and cclustofvar when boruta is used for selection is explained by a difference in the selection step, which might come from the fact that cclustofvar produces

groups with more uneven sizes (including many groups with only one variable in them). However, the group procedure itself is always faster with adjclust.

4.4 | Variation of performances in different simulated scenarios

Figure 6 provides the precision and recall obtained under the different simulation scenarios of Table 1.

As expected, a large number or a small size of ground truth intervals tend to hinder the precision and sometimes the recall as well (scenarios SC5, SC3, SC4 in the top left figure). On the contrary, large intervals (SC1) are more difficult to identify entirely and result in a smaller recall.

Intervals with opposite influence or with different effect sizes (scenario SC6 and SC7) also lead to deteriorated results in terms of both precision and recall. As expected, results are improved by larger sample size but are only marginally impacted by p (top right figure). The link function has a small effect on the performance (second row, left) but the fact that the relevant intervals are smooth (park1 and park2 in second row, right) rather than piecewise constant makes it more difficult to identify the intervals properly. Note that park2 has two intervals with opposite influences and (as for SC6) this leads also to deteriorated precision and recall. Finally, as expected, the level of the noise in simulations negatively impacts both the precision and the recall (last row).

Overall, the method is less impacted by the number of measurement points (p) or by the type of regression (F in Equation 1) than by small sample size, noise level, or adverse situations for $\mathcal{S}$. In particular, it seems to be better suited to detect a small number of impact intervals with similar effect sizes and well distinct, in terms of their contribution to Y, from the rest of the measurement points.

5 | CONCLUSION

Motivated by the idea of selecting relevant intervals in a functional regression framework, this paper proposes an innovative random forest approach. This approach is based on three several steps: an automatic construction of intervals, the computing of relevant summaries on these intervals, and an automatic selection of relevant intervals. Comparing variants for each step on simulated and real datasets, we found that adjclust / PLS / boruta is a relevant combination in all cases and that the selection step improves the performances of the approach.

In future works, we would like to investigate the use of the multi-resolution results provided by SFCB. Indeed, SFCB is a greedy algorithm that generates many different models (with different numbers of intervals) and, then, selects the best model in the sense of a given criterion. An alternative to model choice would be to find a way to aggregate these different models.

SUPPORTING INFORMATION

All used datasets, along with scripts that were used to generate simulated data, are available at https://doi.org/10.57745/KMH2GP.

The method described in this paper is implemented in the function `sfcb` of the R package **SISIR** https://cran.r-project.org/package=SISIR. The code used to perform the experiments included in this article is available at https://forgemia.inra.fr/sfcb/simus_sfcb.git along with resulting notebooks and supplemental results and quality criteria.

ACKNOWLEDGMENTS

We thank Meïli Baragatti and Paul-Marie Grollemund for having shared their truffle dataset and for their help with the **bliss** package, Victor Picheny for useful discussion, and the Genotoul-bioinfo team for providing computing facilities. This work has been partially funded by the INRAE/DIGIT-BIO network “PhenoDyn”.

REFERENCES

1. J. Ah-Pine and X. Wang, *Similarity based hierarchical clustering with an application to text collections*, H. Boström, A. Knobbe, C. Soares, and P. Papapetrou (eds.), *Proceedings of the 15th International Symposium on Intelligent Data Analysis (IDA 2016)*, Lecture Notes in Computer Sciences, Stockholm, Sweden, 2016, 320–331, . URL https://hal.archives-ouvertes.fr/hal-01437124.
2. C. Ambroise, A. Dehman, P. Neuvial, G. Rigaill, and N. Vialaneix, *Adjacency-constrained hierarchical clustering of a band similarity matrix with application to genomics*, Algorithms for Molecular Biology **14** (2019), 22.
3. G. Aneiros and P. Vieu, *Variable in infinite-dimensional problems*, Statistics and Probability Letters **94** (2014), 12–20.
4. S. Arlot, A. Celisse, and Z. Harchaoui, *A kernel multiple change-point algorithm via model selection*, Journal of Machine Learning Research **20** (2019), no. 162, 1–56. URL https://jmlr.org/papers/v20/16-155.html.
5. N. Aronszajn, *Theory of reproducing kernels*, Transactions of the American Mathematical Society **68** (1950), no. 3, 337–404.
6. C. Bahlmann and H. Burkhardt, *The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping*, IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004), no. 3, 299–310.
7. M. Baragatti, P.-M. Grollemund, P. Montpied, J.-L. Dupouey, J. Gravier, C. Murat, and F. Le Tacon, *Influence of annual climatic variations, climate changes, and sociological factors on the production of the Périgord black truffle (Tuber melanosporum Vittad.) from 1903-1904 to 1988-1989 in the Vaucluse (France)*, Mycorrhiza **29** (2019), no. 2, 113–125.
8. R. F. Barber and E. Candès, *Controlling the false discovery rate via knockoffs*, Annals of Statistics **43** (2015), no. 5, 2055–2085.
9. M. Baydogan and G. Runger, *Learning a symbolic representation for multivariate time series classification*, Data Mining and Knowledge

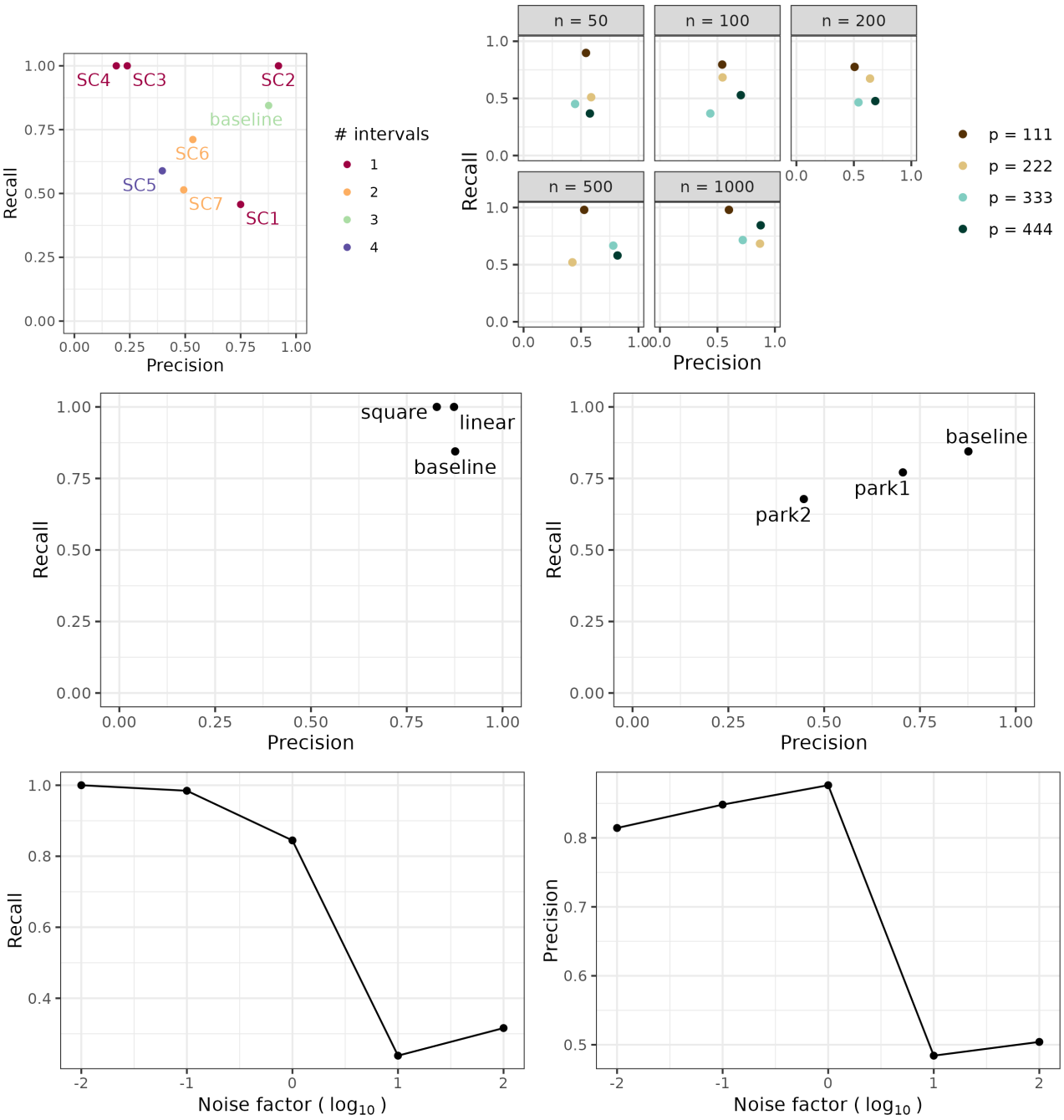

FIGURE 6 Impact of different variations of the simulation setting on the precision and recall of the method: (from left to right and top to bottom) number and size of intervals; n and p; shape of the link function; shape of β; signal-to-noise ratio.

Discovery **29** (2014), 1–23.

10. K. D. Bennett, *Determination of the number of zones in a biostratigraphical sequence*, New Phytologist **132** (1996), no. 1, 155–170.
11. D. Bertsimas and J. Dunn, *Optimal classification trees*, Machine Learning **106** (2017), no. 7, 1039–1082.
12. G. Biau, F. Bunea, and M. Wegkamp, *Functional classification in Hilbert spaces*, IEEE Transactions on Information Theory **51** (2005), no. 6, 2163–2172.
13. R. Blanquero, E. Carrizosa, C. Molero-Rıo, and D. Romero, *On optimal regression trees to detect critical intervals for multivariate functional data*, Computers & Operations Research **152** (2023), 106152.
14. L. Breiman, *Random forests*, Machine Learning **45** (2001), no. 1, 5–32. URL http://www.springerlink.com/content/u0p06167n6173512/fulltext.pdf.
15. N. Brisson et al., *An overview of the crop model STICS*, European Journal of Agronomy **18** (2003), no. 3-4, 309–332.

16. L. Capitaine, J. Bigot, R. Thiébaut, and R. Genuer, *Fréchet random forests for metric space valued regression with non Euclidean predictors*, 2020, . Preprint arXiv:1906.01741 (submitted for publication).
17. M. Chavent, B. Liquet, V. Kuentz-Simonet, and J. Saracco, *ClustOfVar: an R package for the clustering of variables*, Journal of Statistical Software **50** (2012), no. 13, 1–16.
18. H. Deng, G. Runger, E. Tuv, and V. Martyanov, *A time series forest for classification and feature extraction*, Information Science **239** (2013), 142–153.
19. F. Ferraty, P. Hall, and P. Vieu, *Most-predictive design points for functiona data predictors*, Biometrika **97** (2010), no. 4, 807–824.
20. F. Ferraty and P. Vieu, *NonParametric Functional Data Analysis*, Springer, 2006.
21. L. Ferré and N. Villa, *Multi-layer perceptron with functional inputs: an inverse regression approach*, Scandinavian Journal of Statistics **33** (2006), no. 4, 807–823.
22. C. Flecher, P. Naveau, D. Allard, and N. Brisson, *A stochastic daily weather generator for skewed data*, Water Resources Reasearch **46** (2010), no. 7, W07519.
23. S. Frontier, *étude de la décroissance des valeurs propres dans une analyse en composantes principales : comparaison avec le modèle du bâton brisé*, Journal of Experimental Marine Biology and Ecology **25** (1976), no. 1, 67–75.
24. D. Garreau and S. Arlot, *Consistent change-point detection with kernels*, Electronic Journal of Statistics **12** (2018), no. 2, 4440–4486.
25. R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, *Variable selection using random forests*, Pattern Recognition Letters **31** (2010), no. 14, 2225–2236.
26. R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, *VSURF: an R package for variable selection using random forests*, The R Journal **7** (2015), no. 2, 19–33.
27. B. Goehry, H. Yan, Y. Goude, P. Massart, and J.-M. Poggi, *Random forests for time series*, REVSTAT – Statistical Journal **21** (2023), no. 2, 283–302.
28. P.-M. Grollemund, C. Abraham, M. Baragatti, and P. Pudlo, *Bayesian functional linear regression with sparse step functions*, Bayesian Analysis **14** (2019), no. 1, 111–135.
29. N. Hernández, R. J. Biscay, N. Villa-Vialaneix, and I. Talavera, *A non parametric approach for calibration with functional data*, Statistica Sinica **25** (2015), 1547–1566.
30. R. Hornung and A.-L. Boulesteix, *Interaction forests: identifying and exploiting interpretable quantitative and qualitative interaction effects*, Computational Statistics & Data Analysis **171** (2022), 107460.
31. V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts, *Inferring regulatory networks from expression data using tree-based methods*, PLoS ONE **5** (2010), no. 9, e12776.
32. G. M. James, J. Wang, and J. Zhu, *Functional linear regression that's interpretable*, Annals of Statistics **37** (2009), no. 5A, 2083–2108.
33. S. Janitza, E. Celik, and A.-L. Boulesteix, *A computationally fast variable importance test for random forests for high-dimensional data*, Advances in Data Analysis and Classification **12** (2018), 885–915.
34. K. Kira and L. A. Rendell, *A practical approach to feature selection*, D. Sleeman and P. Edwards (eds.), *Proceedings of the International Conference on Machine Learning (ICML 1992)*, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1992, 249–256, .
35. A. Kneip, D. Poß, and P. Sarda, *Functional linear regression with points of impact*, Annals of Statistics **44** (2016), no. 1, 1–30.
36. M. Kursa and W. Rudnicki, *Feature selection with the Boruta package*, Journal of Statistical Software **36** (2010), no. 11, 1–13.
37. G. Lance and W. Williams, *A general theory of classificatory sorting strategies: 1. Hierarchical systems*, The Computer Journal **9** (1967), no. 4, 373–380.
38. J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, *Feature selection: a data perspective*, ACM Computing Surveys **50** (2018), no. 6, 94:1–94:45.
39. J. Lines, S. Taylor, and A. Bagnall, *Time series classification with HIVE-COTE: the hierarchical vote collective of transformation-based ensembles*, ACM Transactions on Knowledge Discovery from Data **12** (2018), no. 5, 1–35.
40. B. Lucas et al., *Proximity forest: an effective and scalable distance based classifier for time series*, Data Mining and Knowledge Discovery **33** (2019), 607–635.
41. D. Makowski and M. Chen, *Apprentissage supervisé pour simuler l'effet du changement climatique sur les rendements agricoles*, *Proceedings of 54èmes Journées de Statistique de la SFdS*, Société Française de Statistique, Bruxelles, Belgium, 2023. URL https://drive.google.com/file/d/16m3wh2Mf0RFtQ_I4gIb9ks2wFP07KivU/view.
42. D. Marbach et al., *Wisdom of crowds for robust gene network inference*, Nature Methods **9** (2012), no. 8, 796–804.
43. I. W. McKeague and B. Sen, *Fractals with point impact in functional linear regression*, Annals of Statistics **38** (2010), no. 4, 2559–2586.
44. M. Middlehurst, J. Large, and A. Bagnall, *The canonical interval forest (CIF) classifier for time series classification*, X. Wu et al. (eds.), *Proceedings of IEEE International Conference on Big Data*, IEEE, Atlanta, GA, USA, 2020, 188–195, .
45. A. Möller, G. Tutz, and J. Gertheiss, *Random forests for functional covariates*, Journal of Chemometrics **30** (2016), no. 12, 715–725.
46. J. Park, J. Ahn, and Y. Jeon, *Sparse functional linear discriminant analysis*, Biometrika **109** (2022), no. 1.
47. V. Picheny, R. Servien, and N. Villa-Vialaneix, *Interpretable sparse sliced inverse regression for functional data*, Statistics and Computing **29** (2019), no. 2, 255–267.
48. A. Poterie, J.-F. Dupuy, V. Monbet, and L. Rouviere, *Classification tree algorithm for grouped variables*, Computational Statistics **34** (2019), no. 4, 1613–1648.
49. J. Qin, D. P. Lewis, and W. S. Noble, *Kernel hierarchical gene clustering from microarray expression data*, Bioinformatics **19** (2003), no. 16, 2097–2104.
50. T. Rainforth and F. Wood, *Canonical correlation forests*, 2017. URL http://arxiv.org/abs/1507.05444, arXiv: 1507.05444.
51. J. O. Ramsay and B. W. Silverman, *Functional Data Analysis*, Springer Verlag, New York, 1997.
52. J. O. Ramsay and B. W. Silverman, *Applied Functional Data Analysis*, Springer Verlag, 2002.
53. N. Randriamihamison, N. Vialaneix, and P. Neuvial, *Applicability and interpretability of Ward's hierarchical agglomerative clustering with or without contiguity constraints*, Journal of Classification **38** (2021), 363–389.
54. M. Robnik-Šikonja and I. Kononenko, *An adaptation of Relief for attribute estimation in regression*, D. H. Fisher (ed.), *Proceedings of the International Conference on Machine Learning (ICML 1997)*, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1997, 296–304.
55. A. Roche, *Lasso in infinite dimension: application to variable selection in functional multivariate linear regression*, Electronic Journal of Statistics **17** (2023), no. 2, 3357–3405.
56. F. Rossi and B. Conan-Guez, *Functional multi-layer perceptron: a nonlinear tool for functional data analysis*, Neural Networks **18** (2005), no. 1, 45–60.
57. F. Rossi and N. Villa, *Support vector machine for functional data classification*, Neurocomputing **69** (2006), no. 7-9, 730–742.
58. P. Schäfer, *The BOSS is concerned with time series classification in the presence of noise*, Data Mining and Knowledge Discovery **29** (2015), no. 6, 1505–1530.
59. A. Shifaz, C. Pelletier, F. Petitjean, and G. J. Webb, *TS-CHIEF: a scalable and accurate forest algorithm for time series classification*, Data Mining and Knowledge Discovery **34** (2020), 742–775.
60. J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, *A comparison of random forest variable selection methods for classification prediction modeling*, Expert Systems with Applications **134** (2019), 93–101.

61. S. Szymczak et al., *r2VIM: a new variable selection method for random forests in genome-wide association studies*, BioData Mining **9** (2016), 7.
62. R. Tibshirani, G. Walther, and T. Hastie, *Estimating the number of clusters in a data set via the gap statistic*, Journal of the Royal Statistical Society Series B **63** (2001), no. 2, 411–423.
63. J. H. Ward, *Hierarchical grouping to optimize an objective function*, Journal of the American Statistical Association **53** (1963), no. 301, 236–244.
64. B. H. Williams, M. Toussaint, and A. J. Storkey, *Extracting motion primitives from natural handwriting data*, S. Kollias, A. Stafylopatis, W. Duch, and E. Oja (eds.), *In Proceedings of the International Conference on Artificial Neural Networks (ICANN 2006)*, *Lecture Notes in Computer Science*, vol. 4132, 2006, 634–643, .
65. H. Wold, *Soft modeling by latent variables; the nonlinear iterative partial least square approach*, Journal of Applied Probability **12** (1975), no. S1, 117–142.

How to cite this article: Servien R., and Vialaneix N.. A random forest approach for interval selection in functional regression for application to agronomy *Stat Anal Data Min: The ASA Data Sci Journal.* 2021;00(00):1–18.

APPENDIX

DETAILED METHODS FOR HIERARCHICAL CLUSTERING

Algorithm 1 Adjacency constrained hierarchical clustering (**adjclust**)

Initialization: $\mathcal{I}_1 = \{I_1^{(1)}, \ldots, I_p^{(1)}\}$ where $I_j^{(1)} = \{t_j\}$.

Define the ``distance'' between t_j and t_{j+1} as the distance induced by the kernel $K(t_j, t_{j'}) = \Sigma_{jj'}$ (where Σ is given by Equation (2)):

$$D(t_j, t_{j'}) := \sqrt{K(t_j, t_j) + K(t_{j'}, t_{j'}) - 2K(t_j, t_{j'})}$$

Set the linkage between $I_j^{(1)}$ and $I_{j+1}^{(1)}$ as in Ward [63]:

$$\delta(I_j^{(1)}, I_{j+1}^{(1)}) = \frac{1}{2} D^2(t_j, t_{j+1})$$

for $t = 1$ to $p-1$ **do**

  Merge the two *contiguous* clusters, $I_{j^*}^{(t)}$ and $I_{j^*+1}^{(t)}$ with minimal linkage value to obtain the next partition $\mathcal{I}_{t+1} = \{I_j^{(t+1)}\}_{j=1,\ldots,p-t}$

  Update linkage values between contiguous clusters using the Lance-Williams formula [37]:

$$\delta(I_{j^*-1}^{(t)}, I_{j^*}^{(t)} \cup I_{j^*+1}^{(t)}) = \frac{|I_{j^*}^{(t)}| + |I_{j^*-1}^{(t)}|}{|I_{j^*}^{(t)}| + |I_{j^*+1}^{(t)}| + |I_{j^*-1}^{(t)}|} \delta(I_{j^*}^{(t)}, I_{j^*-1}^{(t)}) + \frac{|I_{j^*+1}^{(t)}| + |I_{j^*-1}^{(t)}|}{|I_{j^*}^{(t)}| + |I_{j^*+1}^{(t)}| + |I_{j^*-1}^{(t)}|} \delta(I_{j^*+1}^{(t)}, I_{j^*-1}^{(t)}) - \frac{|I_{j^*-1}^{(t)}|}{|I_{j^*}^{(t)}| + |I_{j^*+1}^{(t)}| + |I_{j^*-1}^{(t)}|} \delta(I_{j^*}^{(t)}, I_{j^*+1}^{(t)})$$

end for

return $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_n\}$

Algorithm 2 Constrained clustering of variables (**cclustofvar**)

Initialization: $\mathcal{I}_1 = \{I_1^{(1)}, \ldots, I_p^{(1)}\}$ where $I_j^{(1)} = \{t_j\}$.

Set the homogeneity of any initial cluster to 1: $\mathcal{H}(I_j^{(1)}) := 1$

for $t = 1$ to $p-1$ **do**

  For any pair of *contiguous* clusters, $I_u^{(t)}$ and $I_{u+1}^{(t)}$, compute the correlation matrix, $\mathbf{C}_u^{(t)}$, from $(X(t_j))_{t_j \in I_u^{(t)} \cup I_{u+1}^{(t)}}$: $[\mathbf{C}_u^{(t)}]_{jj'} = \frac{\Sigma_{jj'}}{\sigma_j \sigma_{j'}}$, where σ_j is the empirical standard error of $X(t_j)$

  PCA step: Obtain the first eigenvalue of $\mathbf{C}_u^{(t)}$, $\lambda_u^{(t)}$ and set $\mathcal{H}(I_u^{(t)} \cup I_{u+1}^{(t)}) := \lambda_u^{(t)}$

  Merge the two *contiguous* clusters, $I_{j^*}^{(t)}$ and $I_{j^*+1}^{(t)}$ with minimal loss in homogeneity to obtain the next partition $\mathcal{I}_{t+1} = \{I_j^{(t+1)}\}_{j=1,\ldots,p-t}$:

$$\mathcal{H}(I_{j^*}^{(t)}) + \mathcal{H}(I_{j^*+1}^{(t)}) - \mathcal{H}(I_{j^*}^{(t)} \cup I_{j^*+1}^{(t)})$$

end for

return $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_n\}$

Algorithm 3 Relief algorithm for selection of summary variables (and corresponding intervals) (**cclustofvar**)

Initialization: Set $\mathcal{S}_Y$, $\mathcal{S}_{k,l}$, $\mathcal{S}_{k,l,Y}$ to 0.
$\forall\, i = 1, \ldots, n$, $\mathcal{NN}(i)$ is the set of U nearest neighbors of $\tilde{\mathbf{x}}_{i,k}^{(l)}$ in $(\tilde{\mathbf{x}}_{i',k}^{(l)})_{i'=1,\ldots,n}$.
Compute distances between observations i and $i' \in \mathcal{NN}(i)$ as

$$d(i,i') := \frac{d_1(i,i')}{\sum_{\ell=1}^{n} d_1(i,\ell)} \qquad \text{where } d_1(i,\ell) = e^{-(\text{Rank}(i,i')/\sigma)^2},$$

with $\text{Rank}(i,i')$ the rank of i' in the nearest neighbors of i.
Set $\text{Diff}((k,l),i,i') = \frac{\left|\tilde{\mathbf{x}}_{i,k}^{(l)} - \tilde{\mathbf{x}}_{i',k}^{(l)}\right|}{R_{k,l}}$, with $R_{k,l}$ the range of $\tilde{\mathbf{x}}_{\cdot,k}^{(l)}$ (maximum minus minimum)
for $t = 1$ to T do
  Randomly select $i \in \{1, \ldots, n\}$
  for $i' \in \mathcal{NN}(i)$ do
    $\mathcal{S}_Y \leftarrow \mathcal{S}_Y + |y_i - y_{i'}| \times d(i,i')$ ▷ measures the typical distance between $(y_i)_i$
    $\forall\,(k,l)$, $\mathcal{S}_{k,l} \leftarrow \mathcal{S}_{k,l} + \text{Diff}((k,l),i,i') \times d(i,i')$ ▷ measures the typical distance between $(\tilde{\mathbf{x}}_{i,k}^{(l)})_i$
    $\forall\,(k,l)$, $\mathcal{S}_{k,l,Y} \leftarrow \mathcal{S}_{k,l,Y} + |y_i - y_{i'}| \times \text{Diff}((k,l),i,i') \times d(i,i')$
  end for
end for
return Scores: $\forall\,(k,l)$, $\text{RReliefF}(k,l) := \frac{\mathcal{S}_{k,l,Y}}{\mathcal{S}_Y} - \frac{\mathcal{S}_{k,l} - \mathcal{S}_{k,l,Y}}{T - \mathcal{S}_Y}$