Sanz et al. BMC Bioinformatics (2018) 19:432
https://doi.org/10.1186/s12859-018-2451-4

BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

SVM-RFE: selection and visualization of the most relevant features through non-linear kernels

Hector Sanz[1*], Clarissa Valim[2,3], Esteban Vegas[1], Josep M. Oller[1] and Ferran Reverter[1,4]

Abstract

Background: Support vector machines (SVM) are a powerful tool to analyze data with a number of predictors approximately equal or larger than the number of observations. However, originally, application of SVM to analyze biomedical data was limited because SVM was not designed to evaluate importance of predictor variables. Creating predictor models based on only the most relevant variables is essential in biomedical research. Currently, substantial work has been done to allow assessment of variable importance in SVM models but this work has focused on SVM implemented with linear kernels. The power of SVM as a prediction model is associated with the flexibility generated by use of non-linear kernels. Moreover, SVM has been extended to model survival outcomes. This paper extends the Recursive Feature Elimination (RFE) algorithm by proposing three approaches to rank variables based on non-linear SVM and SVM for survival analysis.

Results: The proposed algorithms allows visualization of each one the RFE iterations, and hence, identification of the most relevant predictors of the response variable. Using simulation studies based on time-to-event outcomes and three real datasets, we evaluate the three methods, based on pseudo-samples and kernel principal component analysis, and compare them with the original SVM-RFE algorithm for non-linear kernels. The three algorithms we proposed performed generally better than the gold standard RFE for non-linear kernels, when comparing the truly most relevant variables with the variable ranks produced by each algorithm in simulation studies. Generally, the RFE-pseudo-samples outperformed the other three methods, even when variables were assumed to be correlated in all tested scenarios.

Conclusions: The proposed approaches can be implemented with accuracy to select variables and assess direction and strength of associations in analysis of biomedical data using SVM for categorical or time-to-event responses. Conducting variable selection and interpreting direction and strength of associations between predictors and outcomes with the proposed approaches, particularly with the RFE-pseudo-samples approach can be implemented with accuracy when analyzing biomedical data. These approaches, perform better than the classical RFE of Guyon for realistic scenarios about the structure of biomedical data.

Keywords: Support vector machines, Relevant variables, Recursive feature elimination, Kernel methods

* Correspondence: hsrodenas@gmail.com
[1]Department of Genetics, Microbiology and Statistics, Faculty of Biology, Universitat de Barcelona, Diagonal, 643, 08028 Barcelona, Catalonia, Spain
Full list of author information is available at the end of the article

Background

Analysis of investigations aiming to classify or predict response variables in biomedical research oftentimes is challenging because of data sparsity generated by limited sample sizes and a moderate or very large number of predictors. Moreover, in biomedical research, it is particularly relevant to learn about the relative importance of predictors to shed light in mechanisms of association or to save costs when developing biomarkers and surrogates. Each marker included in an assay increases the price of the biomarker and several technologies used to measure biomarkers can accommodate a limited number of markers. Support Vector Machine (SVM) models are a powerful tool to identify predictive models or classifiers, not only because they accommodate well sparse data but also because they can classify groups or create predictive rules for data that cannot be classified by linear decision functions. In spite of that, SVM has only recently became popular in the biomedical literature, partially because SVMs are complex and partially because SVMs were originally geared towards creating classifiers based on all available variables, and did not allow assessing variable importance.

Currently, there are three categories of methods to assess importance of variables in SVM: filter, wrapper, and embedded methods. The problem with the existing approaches within these three categories is that they are mainly based on SVM with linear kernels. Therefore, the existing methods do not allow implementing SVM in data that cannot be classified by linear decision functions. The best approaches to work with non-linear kernels are wrapper methods because filter methods are less efficient than wrapper methods and embedded methods are focused on linear kernels. The gold standard of wrapper methods is recursive feature elimination (RFE) proposed by Guyon et al. [1]. Although wrapper methods outweigh other procedures, there is no approach implemented to visualize RFE results. The RFE algorithm for non-linear kernels allows ranking variables but not comparing the performance of all variables in a specific iteration, i.e., interpreting results in terms of: association with the response variable, association with the other variables and magnitude of this association, which is a key point in biomedical research. Moreover, previous work with the RFE algorithm for non-linear kernels has generally focused on classification and disregarded time-to-event responses with censoring that are common in biomedical research.

The work presented in this article expands RFE to visualize variable importance in the context of SVM with non-linear kernels and SVM for survival responses. More specifically, we propose: i) a RFE-based algorithm that allows visualization of variable importance by plotting the predictions of the SVM model; and ii) two variants from the RFE-algorithms based on representation of variables into a multidimensional space such as the KPCA space. In the first section, we briefly review existing methods to evaluate importance of variables by ranking, by selecting variables, and by allowing visualization of variable relative importance. In the Methods section, we present our proposed approaches and extensions. Next, in Results, we evaluate the proposed approaches using simulated data and three real datasets. Finally, we discuss the main characteristics and obtained results of all three proposed methods.

Existing approaches to assess variable importance

The approaches to assess variable importance in SVM can be grouped in filter, embedded and wrapper method classes. Filter methods assess the relevance of variables by looking only at the intrinsic properties of the data without taking into account any information provided by the classification algorithm. In other words, they perform variable selection before fitting the learning algorithm. In most cases, a variable relevance score is calculated, and low-scoring variables are removed. Afterwards, the "relevant" variable subset is input into the classification algorithm. Filter methods include the F-score [2, 3].

Embedded methods, are built into a classifier and, thus, are specific to a given learning algorithm. In the SVM framework, all embedded methods are limited to linear kernels. Additionally, most of these methods are based on a somewhat penalization term, i.e., variables are penalized depending on their values with some methods explicitly constraining the number of variables, and others penalizing the number of variables [4, 5]. An additional exact algorithm was developed for SVM in classification problems using the Benders decomposition algorithm [6]. Finally, a penalized version of the SVM with different penalization terms was suggested by Becker et al. [7, 8]

Wrapper methods evaluate a specific subset of variables by training and testing a specific classification model, and are thus, tailored to a specific classification algorithm. The idea is to search the space of all variable subsets with an algorithm wrapped around the classification model. However, as the space of variables subset grows exponentially with the number of variables, heuristic search methods are used to guide the search for an optimal subset. Guyon et al. [1] proposed one of the most popular wrapper approaches for variable selection in SVM. The method is known as SVM-Recursive Feature Elimination (SVM-RFE) and, when applied to a linear kernel, the algorithm is based on the steps shown in Fig. 1. The final output of this algorithm is a ranked list with variables ordered according to their relevance. In the same paper, the authors proposed an approximation for non-linear

Data : Dataset with p^* variables and binary outcome.

Output: Ranked list of variables according to their relevance.

Find the optimal values for the tuning parameters of the SVM model;

Train the SVM model;

$p \leftarrow p^*$;

while $p \geq 2$ **do**

  $SVM_p \leftarrow$ SVM with the optimized tuning parameters for the p variables and observations in **Data**;

  $w_p \leftarrow$ calculate weight vector of the SVM_p $(w_{p1}, \ldots, w_{pp})$;

  $rank.criteria \leftarrow (w_{p1}^2, \ldots, w_{pp}^2)$;

  $min.rank.criteria \leftarrow$ variable with lowest value in $rank.criteria$ vector;

  Remove $min.rank.criteria$ from **Data**;

  $Rank_p \leftarrow min.rank.criteria$;

  p ← p - 1 ;

end

$Rank_1 \leftarrow$ variable in **Data** $\notin (Rank_2, \ldots, Rank_{p^*})$;

return $(Rank_1, \ldots, Rank_{p^*})$

Fig. 1 Pseudo-code of the SVM-RFE algorithm using the linear kernel in a model for binary classification

kernels. The idea is based on measuring the smallest change in the cost function by assuming no change in the value of the estimated parameters in the optimization problem. Thus, one avoids to retrain a classifier for every candidate variable to be eliminated.

SVM-RFE method is basically a backward elimination procedure. However, the variables that are top ranked (eliminated last) are not necessarily the ones that are individually most relevant but the most relevant conditional on the specific ranked subset in the model. Only taken together the variables of a subset are optimal in some sense. So for instance, if we are focusing on a variable that is p ranked we know that in the model with the 1 to p ranked variables, p is the variable least relevant.

The wrapper approaches include the interaction between variable subset search and model selection as well as the ability to take into account variable correlations. A common drawback of these techniques is that they have a higher risk of overfitting than filter methods and are computationally intensive, especially if building the classifier has a high computational cost [9]. Additional work has been done to assess variable importance in non-linear kernels SVM by modifying SVM-RFE [3, 10, 11].

The methods we propose in the next section are based on a wrapper approach, specifically in the RFE algorithm, allowing visualization and interpretation of the relevant variables in each RFE iteration using linear or non-linear kernels and fitting SVM extensions such as SVM for survival analysis,

Methods

RFE-pseudo-samples

One of our proposed methods follows and extends the idea proposed in Krooshof et al. [12] and Postma et al. [13] to visualize the importance of variables using pseudo-samples in the kernel partial least squares and the support vector regression (SVR) context, respectively. The proposed is applicable to SVM classifying binary outcomes. Briefly, the main steps are the following:

1. Optimize the SVM method and tune the parameters.
2. For each variable of interest, create a pseudo-samples matrix with equally distanced values $z_*$from the original variable, while maintaining the other variables set to their mean or median (1). z_q can be quantiles of the variable for an arbitrary q that is the number of selected quantiles. As the data is usually normalized, we assume that the mean is 0. There will be p pseudo-samples matrices of dimension q x p. For instance, for variable 1, the pseudo-sample matrix will look like in (1) with q pseudo-samples vectors.

$$\begin{pmatrix} V_1 & V_2 & V_3 & \dots & V_p \\ z_1 & 0 & 0 & \dots & 0 \\ z_2 & 0 & 0 & \dots & 0 \\ z_3 & 0 & 0 & \dots & 0 \\ & & \vdots & & \\ z_q & 0 & 0 & \dots & 0 \end{pmatrix} \begin{matrix} pseudo{-}samples_1 \\ pseudo{-}samples_2 \\ pseudo{-}samples_3 \\ \vdots \\ pseudo{-}samples_q \end{matrix} \quad (1)$$

3. Obtain the predicted decision value (not the predicted class) from SVM (a real negative or positive value) for each pseudo-sample using the SVM model fitted in step 1. Basically, this decision value corresponds to the distance of each observation from the SVM margins.
4. Measure the variability of each variable's prediction using the univariate robust metric median absolute deviation (MAD). This mesure is expressed for a given variable p as

$$MAD_p = median(|D_{qp} - median(D_p)|)c$$

being D_{qp} the decision value of the variable p for the pseudo-sample q and being $median(D_p)$ the median of all decision values for the evaluated variable p. The constant c is equal to 1.4826, and it is incorporated in the expression to ensure consistency in terms of expectation so that

$$E(MAD(D_1, ..., D_n)) = \sigma$$

for D_i distributed as $N(\mu, \sigma^2)$ and large n [14, 15].

5. Remove the variable with the lowest MAD value.
6. Repeat steps 2–5 until there is only one variable left (applying in this way the RFE algorithm as detailed in Fig. 2).

The rationale of the proposed method is that for variables associated with the response, modifications in the variable will affect predictions. On the contrary, for variables not associated with the response, changes in the variable value will not affect predictions and the decision value will be approximately constant. Therefore, since the decision value can be used as a score that measure distance to the hyperplane, the larger the absolute value the more confident we are that the observation belongs to the predicted class defined by the sign.

Data : Dataset with p^* variables, time-to-event and status.
Input : Number of equidistant cutoff points c^*.
Output: Ranked list of variables according to their relevance.

Find the optimal values for the tuning parameters of the SVM model;
$p \leftarrow p^*$;
while $p \geq 2$ **do**
  $SVM_p \leftarrow$ SVM with the optimized tuning parameters for the p variables and observations in **Data**;
  for $i = 1$ **to** p **do**
    $pseudo_i \leftarrow$ prediction vector of c^* pseudo-samples for variable i;
    $rank.criteria_i \leftarrow$ Median Absolute Deviation of $pseudo_i$ vector;
  end
  $min.rank.criteria \leftarrow$ variable with lowest value in $(rank.criteria_1, \dots, rank.criteria_p)$;
  Remove $min.rank.criteria$ from **Data**;
  $Rank_p \leftarrow min.rank.criteria$;
  p ← p - 1 ;
end
$Rank_1 \leftarrow$ variable in **Data** $\notin (Rank_2, \dots, Rank_{p^*})$;
return $(Rank_1, \dots, Rank_{p^*})$

Fig. 2 Pseudo-code of the RFE-pseudo-samples algorithm applied to a time-to-event (right-censored) response variable

Visualization of variables

The RFE-pseudo-samples algorithm allows us to plot the decision values and the range of all variables, in this way we account for:

- Strenght and direction of the association between individual variables and the response: since we are plotting the range of the variable and the decision value, we are able to detect whether larger values of the variable are protective or risk factors.
- The proposed method fix the values of the non-evaluated variables to 0 but this can be modified to evaluate the performance of the desired variables fixing the values to any other biologically meaningful value.
- The distribution of the data can be indicative of the type of association of each variable with respect the response, i.e., U-shaped, linear or exponential, for example.
- The variability on the decision values can be indicative of the relevance of the variable with the response. Given a variable, the more variability on the decision values along its range the more associated is the variable with the response.

RFE-kernel principal components input variables

Reverter et al. [16] proposed a method using the kernel principal component analysis (KPCA) space (more detail on the KPCA methodology in Additional file 1) to represent, for each variable, the direction of maximum growth locally. So, given two leading components the maximum growth for each variable is indicated in a plot in which each axis is one of the components. After representing all observations in the new space, if a variable is relevant under this context will show a clear direction across all samples and if it's not the sample's direction will be random. In the same work the authors suggest to incorporate functions of the original variables into the KPCA space, so it's possible to plot not only growth of individual variables but combination of them if makes sense within the research study. Our proposed method, referred as RFE-KPCA-maxgrowth, consists of the following steps:

1. Fit the SVM.
2. Create the KPCA space using the tuned parameters found in the SVM process with all variables if possible, for example, when the kernel used in SVM is the same than in KPCA.
3. Represent the observations with respect the two first components of the KPCA.
4. Compute and represent the input variables and the decision function of the SVM into the KPCA output, as detailed in Representation of input variables section.
5. Compute the average angle of each variable-observation with the decision function into the KPCA output. Therefore, an average angle using all observations, can be calculated for each variable (Ranking of variables section).
6. Calculate the difference for each variable between the average angle and the median of all variables average angle. The variable closest to the median is classified as the less relevant, as detailed in Ranking of variables section.
7. Remove the least relevant variable.
8. Repeat all the process from 1 to 7 until there is one variable left.

Representation of input variables

We approach the problem of the interpretability of kernel methods by mapping simultaneously data points and relevant variables in a low dimensional linear manifold immersed in the kernel induced feature space H [17]. Such linear manifold, usually a plane, can be determined according to some statistical requirement, for instance, we shall require that the final Euclidean interdistances between points in the plot have to be, as far as possible, similar to the interdistances in the feature space, which shall lead us to the KPCA. We have to distinguish between the feature space H and the surface in that space to which points in input space $\mathbb{R}^p$ actually map, which we denote by $\phi(\mathcal{X})$. In general is a dimensional manifold embedded in H. We assume here that $\phi(\mathcal{X})$ is sufficiently smooth that a Riemannian metric can be defined on it [18].

The intrinsic geometrical properties of $\phi(\mathcal{X})$ can be derived once we know the Riemannian metric induced by the embedding of $\phi(\mathcal{X})$ in H. The Riemannian metric can be defined by a symmetric metric tensor g_{ab}. The explicit mapping to construct g_{ab} is unkonwn; it can be written solely in terms of the kernel [17].

Any relevant variable can be described by a real valued function f defined on the input space $\mathbb{R}^p$. Since we assume that the feature map ϕ is one-to-one, we can identify f with $\tilde{f} \equiv f \circ \phi^{-1}$ defined on $\phi(\mathcal{X})$. We aim to represent the gradient of $\tilde{f}$. The gradient of $\tilde{f}$ is a vector field defined on $\phi(\mathcal{X})$ through its components under the coordinates $\boldsymbol{x} = (x^1, \ldots, x^p)$ as

$$\operatorname{grad}\left(\tilde{f}\right)^a = \sum_{b=1}^{p} g^{ab}(\boldsymbol{x}) D_b f(\boldsymbol{x}) \; a = 1, \ldots, p \qquad (2)$$

where g^{ab} is the inverse of the metric matrix $G = (g_{ab})$ and D_b denotes the partial derivative with respect the b variable.

The curves ν corresponding to the integral flow of the gradient, i.e., the curves whose tangent vectors at t are $\nu'(t) = grad(\tilde{f})$. These curves indicate, locally, the maximum variation directions of $\hat{f}$. Under the coordinates $\boldsymbol{x} = (x^1, \ldots, x^p)$ the integral flow is the general solution of the first order differential equation system

$$\frac{dx^a}{dt} = \sum_{b=1}^{p} g^{ab}(\boldsymbol{x}) D_b f(\boldsymbol{x}) \; a = 1, \ldots, p \tag{3}$$

which has always local solution given initial conditions $\nu(t_0) = \boldsymbol{w}$.

To help interpreting the KPCA output, we can plot the projected $\nu(t)$ curves (obtained in eq. 3) that indicates, locally, the maximum variation directions of $\tilde{f}$, or also, the corresponding gradient vector given in (2).

Let $\nu(t) = k(\cdot, \boldsymbol{x}(t))$ where $\boldsymbol{x}(t)$ are the solutions of (3). If we define

$$\boldsymbol{Z}_t = (k(\boldsymbol{x}(t), \boldsymbol{x}_i))_{n\text{x}1}, \tag{4}$$

the induced curve, $\tilde{\nu}(t)$, expressed in matrix form, is given by the row vector

$$\tilde{\nu}(t)^q_{1\text{x}r} = \left(\boldsymbol{Z}_t' - \frac{1}{n}\mathbf{1}_n' K\right)\left(\boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n'\right)\tilde{\boldsymbol{V}} \tag{5}$$

where $\boldsymbol{Z}_t$ has the form (4), and $'$ symbol indicates transposed.

We can also represent the gradient vector field of $\hat{f}$, that is, the tangent vector field corresponding to curve $\nu(t)$ through its projection into the KPCA output. The tangent vector at $t = t_0$, if $x_0 = \phi^{-1} \circ \nu(t_0)$ is given by $\frac{d\nu}{dt}|_{t=t_0}$, and its projection, in matrix form, is given by the row vector

$$\left(\frac{d\tilde{\nu}}{\mathrm{d}t}\bigg|_{t=t_0}\right)_{1\text{x}r} = \frac{d\boldsymbol{Z}_t'}{dt}\bigg|_{t=t_0}\left(\boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n'\right)\tilde{\boldsymbol{V}} \tag{6}$$

with

$$\frac{d\boldsymbol{Z}_t'}{dt}\bigg|_{t=t_0} = \left(\frac{d\boldsymbol{Z}_t^1}{dt}\bigg|_{t=t_0}, \ldots, \frac{d\boldsymbol{Z}_t^n}{dt}\bigg|_{t=t_0}\right)', \tag{7}$$

and,

$$\begin{aligned}\frac{d\boldsymbol{Z}_t^i}{dt}\bigg|_{t=t_0} &= \frac{dk(\boldsymbol{x}(t), \boldsymbol{x}_i)}{dt}\bigg|_{t=t_0} \\ &= \sum_{a=1}^{p} D_a k(\boldsymbol{x}_0, \boldsymbol{x}_i)\frac{dx^a}{dt}\bigg|_{t=t_0}\end{aligned} \tag{8}$$

where $\frac{dx^a}{dt}|_{t=t_0}$ is defined in (3).

Ranking of variables

Our proposal is to take advantage of the representation of direction of input variables applying two alternative approaches:

- To include the SVM predicted decision values for each training sample as an extra variable, what we call *reference variable*. Then, compare directions of each one of the input variables with the reference.
- To include the direction of the SVM decision function and use it as the *reference direction*. Since it is as a real-valued function of the original variables we can represent the direction of this expression. Specifically, the decision function removing the sign function of the expression of SVM is given by

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{9}$$

we can reformulate (9) to

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \varrho_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{10}$$

where $\varrho_i = \alpha_i y_i$. Applying the representation of input variables methodology to function (10) and assuming Gaussian kernel expressed as $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\frac{1}{\sigma}\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2)$, from formula (8),

we obtain

$$\begin{aligned}\frac{d\boldsymbol{Z}_t^i}{dt}\bigg|_{t=0} &= k(\boldsymbol{x}_i, \boldsymbol{x}) \sum_{a=1}^{p}\left(x_i^a - x^a\right) \\ &\times \left[\sum_{j=1}^{n} \varrho_i \, \sigma\left(x_j^a - x^a\right) k\left(\boldsymbol{x}_j, \boldsymbol{x}\right)\right]\end{aligned}$$

For both prediction values and decision function, we can calculate the overall similarity of one variable with respect the reference (either the prediction or the decision function) by averaging the angle of the maximum growth vector for all training points with the reference. So, if, for a given training point, the angle of the direction of maximum growth of variable p with the reference is 0 (0 rad) would mean that the vector of directions overlap and they are perfectly positively associated. If the angle is 180 (π radians) they go in opposite direction, indicating that they are perfectly negatively associated (Fig. 3). By averaging the angle of all training points we obtain a summary of the similarity of each variable with the reference and, consequently, whether is relevant or not. Assuming that there is noise in real data, a variable is classified as relevant or not compared to the others: the variable closest to the overall angle taking into account all variables is assumed

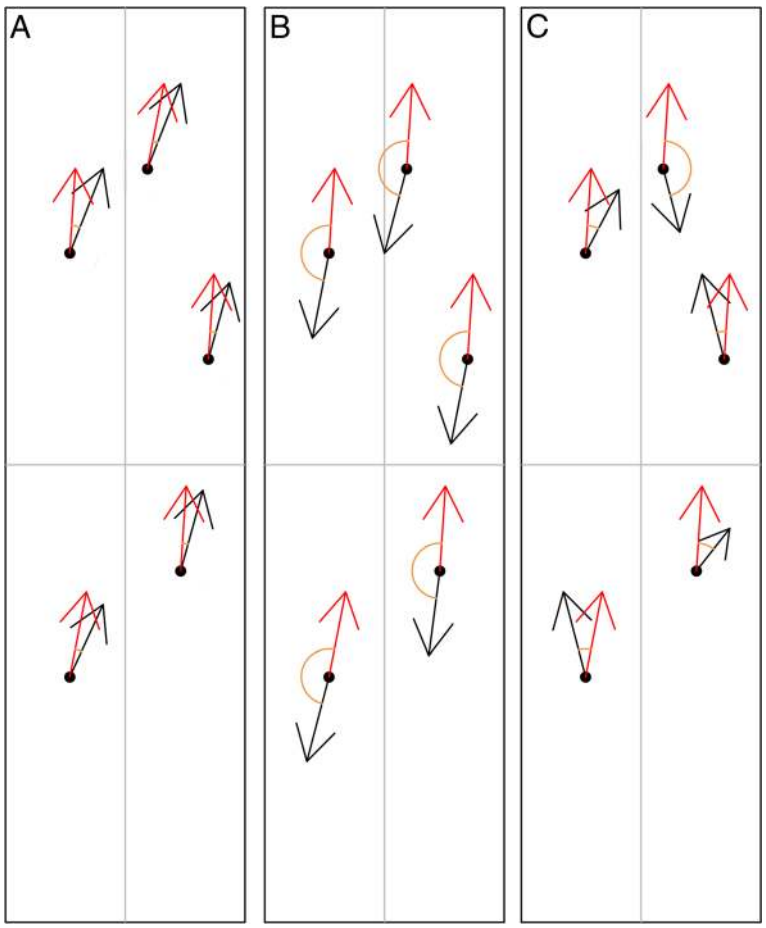

Fig. 3 Visual representation of variable importance. Vectors are the projection on the two leading KPCA axes of the vectors in the kernel feature space pointing to the direction of maximum locally growth of the represented variables. In this scheme, the reference variable is in red and original variables are in black. Each sample point anchors a vector representing the direction of maximum locally growth. **a** When an original variable is associated with the reference variable, the angle between both vectors, averaged across all samples, is close to zero radians. **b** In contrast, when an original variable is negatively associated with the reference variable, the angle between both vectors, averaged across all samples, is close to π radians. **c** When an original variable does not show any association with the reference variable, the angle changes non-consistently among the samples. In noisy data, behavior (**c**) is expected to occur in most variables, so the variable with average angle closest to the overall angle after accounting for all variables is assumed to be the least relevant

to be the least relevant. Based on this, we can apply a RFE-KPCA-maximum-growth approach for prediction and for decision function as defined by Fig. 4.

Visualization of importance of variables

We can represent for each observation the original variables as vectors (with a pre-specified length), that indicate the direction of maximum growth in each variable or a function of each variable. When two variables are positively correlated, the directions of maximum growth for all samples should appear in the same direction and in the perfect scenario samples should overlap. When two variables are negatively correlated the direction should be overall opposite, i.e., should be a mirror image, and if they are no correlated, directions should be random (Fig. 3).

Compared scenarios

To fix ideas, we applied the three proposed approaches: RFE-pseudo-samples, RFE-KPCA-maxgrowth-prediction and RFE-KPCA-maxgrowth-decision and compared them to the RFE-Guyon for non-linear kernels. These methods are applied to analyse simulated and real time-to-event data with SVM. We simulated a time-to-event response variable and the corresponding censoring distribution. To evaluate the performance of the proposed methods in this survival framework, several scenarios involving different correlated variables have been simulated.

Data : Dataset with p^* variables, time-to-event and status.
Input : Method of KPCA-maxgrowth to be applied (*Prediction* or *Function*).
Output: Ranked list of variables according to their relevance.

Find the optimal values for the tuning parameters of the SVM model.;
$p \leftarrow p^*$;
while $p \geq 2$ **do**
  Fit SVM with the optimized tuning parameters for the p variables and observations in **Data**.;
  if *Method* = *Prediction* **then**
    Calculate the decision value for each training point;
    Incorporate that decision value vector as a variable k;
    Calculate the KPCA space with all $(p+1)$ variables (including k);
    Project all variables into KPCA space for the first two components;
  end
  if *Method* = *Function* **then**
    Calculate the KPCA space with all p variables;
    Project all variables into KPCA space for the first two components;
    Project the SVM decision function, k, into the KPCA space;
  end
  $angle_{ij} \leftarrow$ calculate the angle of each training point i and variable j with respect k;
  $angle.mean_j \leftarrow$ average $angle_{ij}$ values by each variable j obtaining a vector of p components;
  $med_p \leftarrow$ overall median of all $angle.mean_1, \ldots, angle.mean_p$;
  if $p \geq 3$ **then**
    $rank.criteria_p \leftarrow (angle.mean_j - med_p)^2$;
  else
    $rank.criteria_p \leftarrow (angle.mean_j - 90°)^2$;
  end
  $min.rank.criteria \leftarrow$ variable with lowest value in $(rank.criteria_1, \ldots, rank.criteria_p)$;
  Remove $min.rank.criteria$ from **Data**;
  $Rank_p \leftarrow min.rank.criteria$;
  p ← p - 1 ;
end
$Rank_1 \leftarrow$ variable in **Data** $\notin (Rank_2, \ldots, Rank_{p^*})$;
return $(Rank_1, \ldots, Rank_{p^*})$

Fig. 4 Pseudo-code of the RFE-KPCA-maximum-growth algorithm for both function and prediction approach. The algorithm is applied to a time-to-event (right-censored) response variable

Simulation of scenarios and data generation

We generated 100 datasets with a time-to-event response variable and 30 predictor variables following a multivariate normal distribution. The mean of each variable was a realization of a Uniform distribution U(0.03,0.06) and the covariance matrix was computed so that all variables were classified in four groups according to their pairwise correlation: no correlation (around 0), low correlation (around 0.2), medium correlation (around 0.5) and high correlation (around 0.8). The variance distribution of each variable was fixed to 0.7 (see correlation matrix at Additional File 2).

The time-to-event variable was simulated based on the proportional hazards assumption through a Gompertz distribution [19]:

$$T = \frac{1}{\alpha}\left(1 - \frac{\alpha \log(U)}{\gamma \exp(\langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle)}\right) \tag{11}$$

where U is a variable following a Uniform(0,1) distribution, $\boldsymbol{\beta}$ is the coefficients variable vector, $\alpha \in (-\infty, \infty)$ and $\gamma > 0$ are the scale and shape parameters of the Gompertz distribution. These parameters were selected so that overall survival was around 0.6 at 18 months follow-up time.

The number of observations in each dataset was 50 and the time of censoring distribution followed a Uniform allowing around 10% censoring.

Relevance of variables scenarios

To evaluate the proposed methods, we generated the time-to-event response variable assuming the following scenarios: i) large and low pairwise correlation among predictors, some of them with variables highly associated with the response and others not, ii) positive and negative association with the response variable, and iii) linear and non-linear associations with the response variable and, in some cases, interaction among predictor variables. The relevant variables for each one of the 6 simulated scenarios are:

1. Variable 1.
2. -Variable 29 + Variable 30.
3. -Variable 1 + Variable 8 + Variable 20 + Variable 29 - Variable 30.
4. Variable 1 + Variable 2 + Variable 1 x Variable 2.
5. Variable 1 + Variable 30 + Variable 1 x Variable 30 + Variable 20 + (Variable 20)^2.
6. Variable 1 + (Variable 1)^2 + exp(Variable 30).

Real-life datasets

The PBC, Lung and DLBCL datasets freely available at the CRAN repository were used as real data to test the performance of the proposed methods. Briefly, datasets of the following studies were analyzed:

- PBC: this data is from the Mayo Clinic trial in primary biliary cirrhosis of the liver conducted between 1974 and 1984. The study aimed to evaluate the performance of the drug D-penicillamine in a placebo controlled randomized trial. This data contains 258 observations and 22 variables (17 of them are predictors). From the whole cohort 93 observations experienced the event, 65 finalized the follow-up period being a non-event, and thus were censored, and 100 were censored before the end of the follow-up time of 2771 days, with an overall survival probability of 0.57.
- Lung: this study was conducted by the North Central Cancer Treatment Group (NCCTG) and aimed to estimate the survival of patients with advanced lung cancer. The available dataset included 167 observations, experiencing 89 events during the follow-up time of 420 days, and 10 variables. A total of 36 observations were censored before the end of follow-up. The overall survival was 0.40.
- DLBCL: this dataset contains gene expression data from diffuse large B-cell lymphoma (DLBCL) patients. The available dataset contains 40 observations and 10 variables representing the mean gene expression in 10 different clusters. From the analysed cohort 20 patients experienced the event, 10 finalized the follow-up and 8 were right-censored during the 72 months follow-up period.

Cox proportional-hazards models were used and compared with the proposed methods. We applied the RFE algorithm and in each iteration the variable with lowest proportion of explainable log-likelihood in the Cox model was removed. To compare the obtained rank of variables the correlation between the ranks was computed. Additionally, the C statistic was computed by ranked variable and method to evaluate its discriminative ability.

Probabilistic SVM

The data was analysed with a modified SVM for survival analysis that was previously considered optimal to handle censored data [20]. The method, known as probabilistic SVM [21] (more details on this method on Additional file 3), allows not perfectly defining some observations and give them an uncertainty in their class. For these uncertainties a confidence level or probability regarding the class is provided.

Comparison of methods

The parameters selected to perform the grid-search for Gaussian kernel were 0.25, 0.5, 1, 2 and 4. The C and $\tilde{C}$ values were 0.1, 1, 10 and 100. For each combination of parameters, a tunning parameter step with 10 training datasets were fitted and validated using 10 different validation datasets. Additionally, 10 training datasets, different from all datasets used in the tuning parameters step, were simulated and fitted with the best combination found in tuning parameters step. The tuned parameters were fixed for each RFE iteration, i.e., were not estimated at each iteration. Once the optimal parameters for the pSVM were found the methods compared were:

- RFE-Guyon for non-linear data: this method was considered the gold standard.
- RFE-KPCA-maxgrowth-prediction: the KPCA is based on Gaussian kernel with parameters obtained in the pSVM model.
- RFE-KPCA-maxgrowth-decision: the KPCA is based on Gaussian kernel with parameters obtained in the pSVM model.
- RFE-pseudo-samples: the range of the data, to create the pseudo-samples is created split- ting data into 50 equidistant points. The range of the pseudo-samples goes from − 2 to 2, since variables are normally distributed around 0 approximately.

Metrics to evaluate algorithm performance

The mean and standard deviation of the rank obtained in 100 simulated datasets was used to summarize the performance by method and scenario. For the RFE-pseudo-samples algorithm the first iteration figure with all 100 datasets was created summarizing the information by variable. For the RFE-maxgrowth approach, as example, one of the datasets was presented in order to interpret the method, since it was not possible to summarize all 100 principal components plots in one figure.

Results

Simulated datasets

In this section, main results are described by algorithm and scenario. Results are structured according to overall ranking of variables and visualization and interpretation of two scenarios for illustrative purposes.

Overall ranking comparison

Scenario 1 results are shown in Fig. 5. All 4 methods identified the relevant variable being the RFE-maxgrowth-prediction the one with the lowest average rank (thus, optimal), followed by the RFE-maxgrowth-function, RFE-pseudo-samples and RFE-Guyon. For all methods, except the RFE- Guyon, a set of variables was closest to the Variable 1 rank (variables 2 to 8). These variables were highly correlated with Variable 1.

For scenario 2 (Fig. 6), the true relevant variables were identified for all 4 algorithms, being the average rank pretty similar, except the RFE-maxgrowth-function. The specific overall rank order was RFE-Guyon, RFE-maxgrowth-prediction, RFE-pseudo-samples and RFE-maxgrowth-function. The average rank for the other non-relevant variables was similar for all methods. In this scenario the relevant variables were not correlated with any other variable in the dataset.

In scenario 3 (Fig. 7), 5 variables are relevant in the true model. The algorithms were able to detect the relevant non-correlatedvariables (variables 20, 29 and 30), except the RFE-maxgrowth-function, that for this set of variables was the worst method. For the other 3 algorithms and this set of variables, the RFE-pseudo-samples was slightly better and the RFE-Guyon slightly worst than the others. For the other 2 highly correlated variables (Variable 1 and Variable 8) the two best methods were clearly RFE-pseudo-samples and RFE-maxgrowth-function.

In Scenario 4 (Fig. 8), all methods, except RFE-Guyon, detected the two relevant variables. However, RFE-maxgrowth-function identified as relevant, with a pretty similar rank, variables 3 to 8 (highly correlated with the true relevant ones). The RFE-pseudo-samples algorithm ranks increased as the correlation with the true relevant variables decreased.

For Scenario 5 (Fig. 9) three variables were relevant (1, 20 and 30). An interaction and a quadratic term were included. RFE-pseudo-samples was clearly the method that

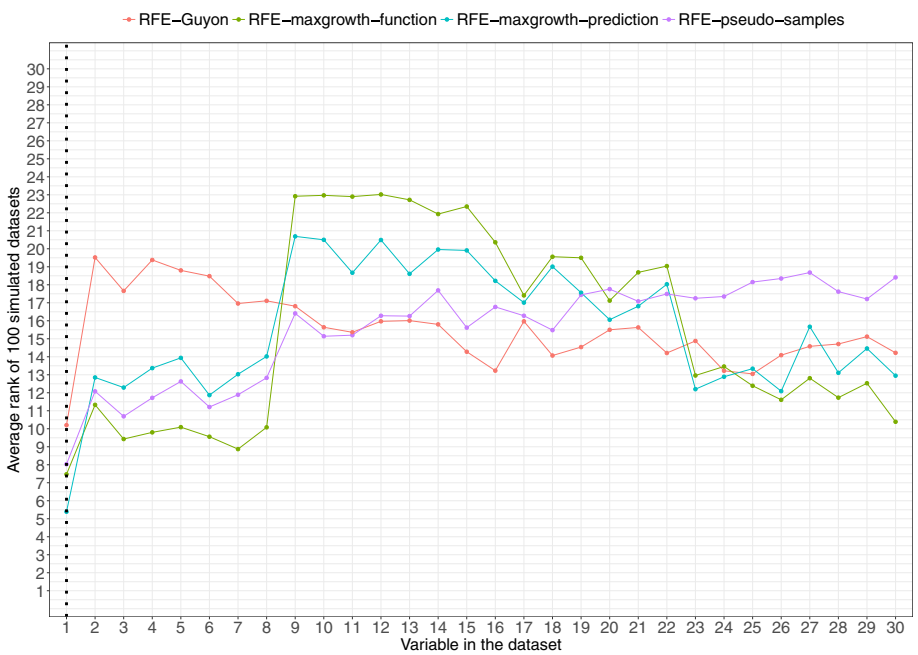

Fig. 5 Scenario 1 results. Average rank by variable and method for the 100 simulated datasets for Scenario 1 (being Variable 1 the relevant variable). Dotted vertical black line represents the variable used to generate the time-to-event variable. The lower the rank, the more relevant the variable is for the specific algorithm

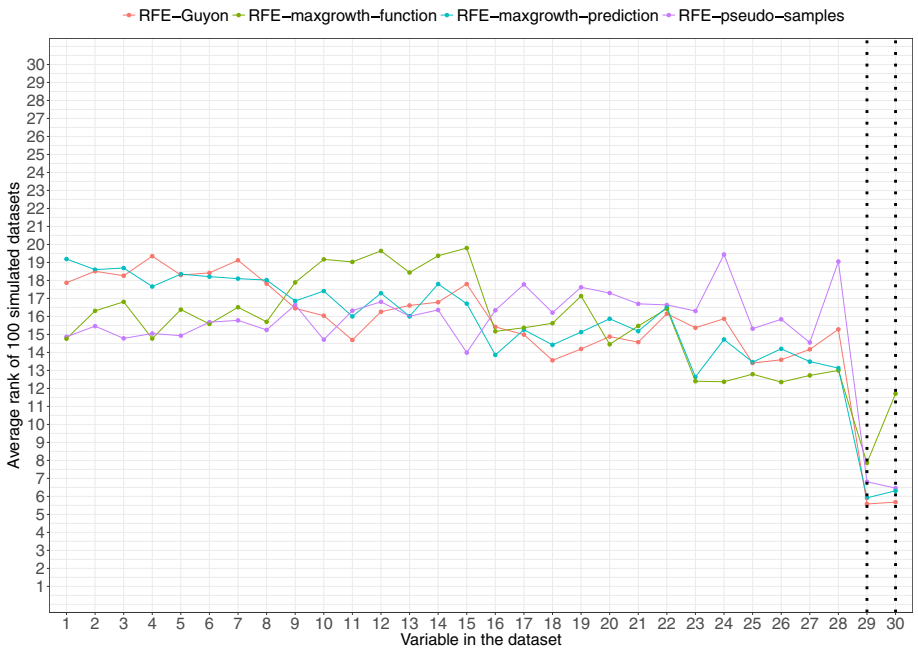

Fig. 6 Scenario 2 results. Average rank by variable and method for the 100 simulated datasets for Scenario 2 (being variables 29 and 30 the relevant variables). Dotted vertical black lines represent the variable used to generate the time-to-event variable. The lower the rank, the more relevant the variable is for the specific algorithm

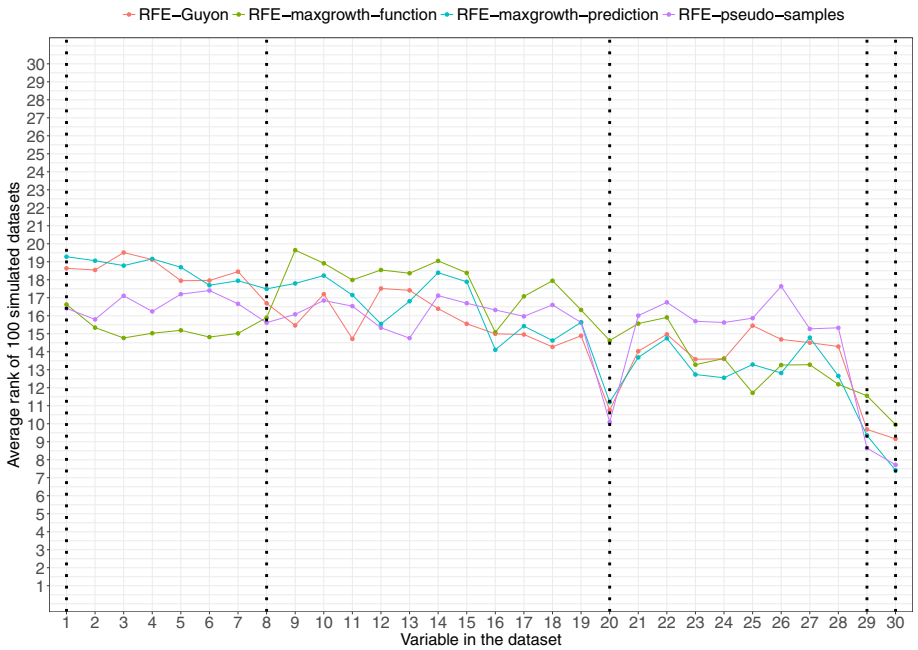

Fig. 7 Scenario 3 results. Average rank by variable and method for the 100 simulated datasets for Scenario 3 (being variables 1, 8, 20, 29 and 30 the relevant variables). Dotted vertical black lines represent the variables used to generate the time-to-event variable. The lower the rank, the more relevant the variable is for the specific algorithm

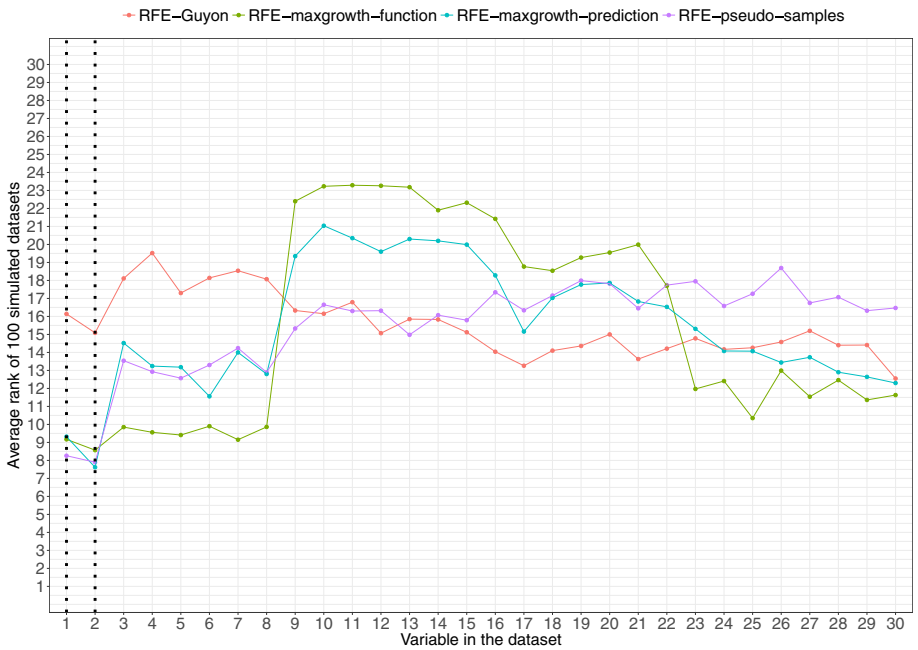

Fig. 8 Scenario 4 results. Average rank by variable and method for the 100 simulated datasets for Scenario 4 (being variables 1 and 2 the relevant variables). Dotted vertical black lines represent the variables used to generate the time-to-event variable. The lower the rank the more relevant the variable is for the specific algorithm

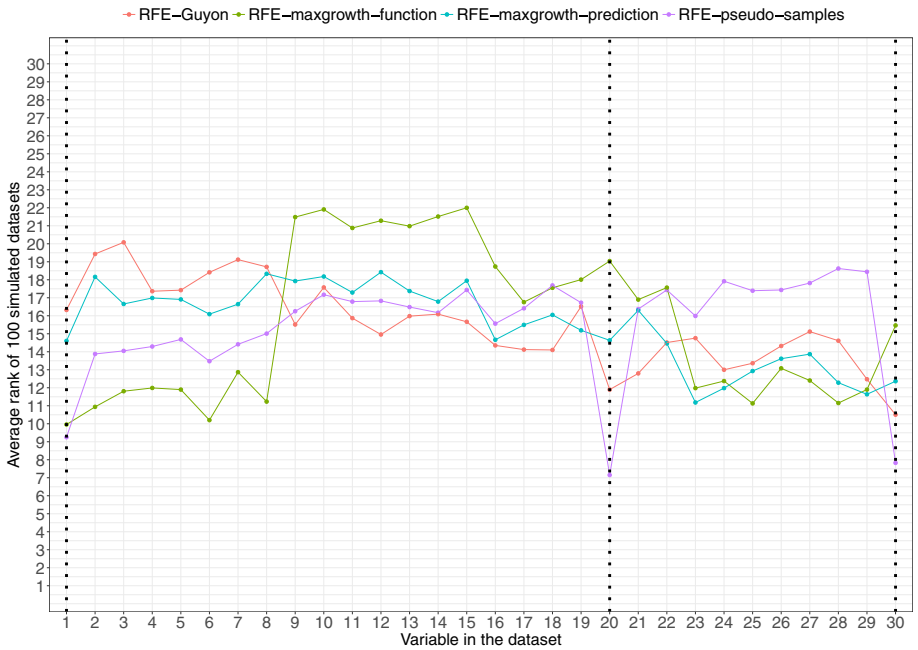

Fig. 9 Scenario 5 results. Average rank by variable and method for the 100 simulated datasets for Scenario 5 (being variables 1, 20 and 30 the relevant variables). Dotted vertical black lines represent the variable used to generate the time-to-event variable. The lower the rank the more relevant the variable is for the specific algorithm

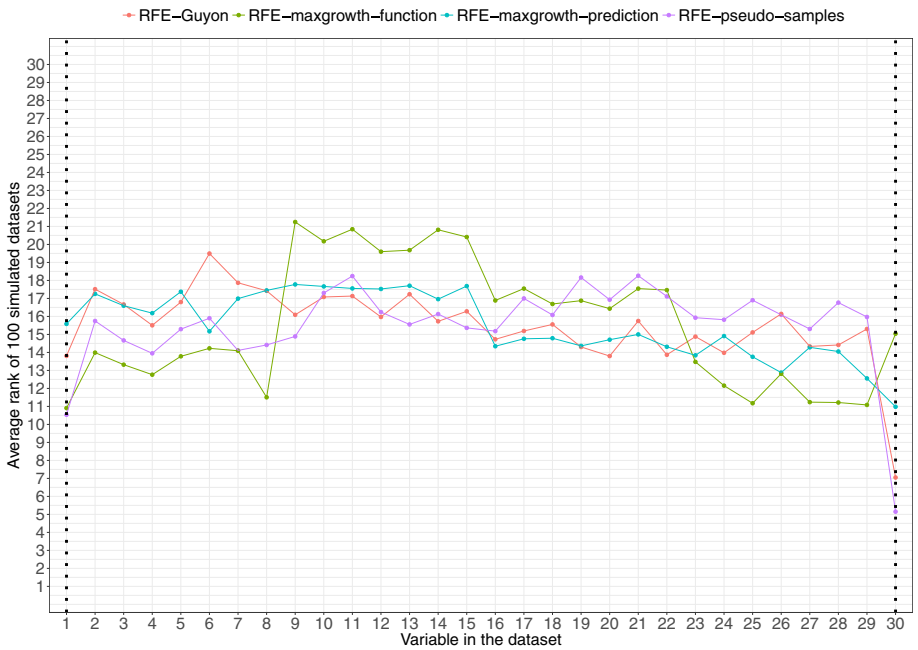

Fig. 10 Scenario 6 results. Average rank by variable and method for the 100 simulated datasets for Scenario 6 (being variables 1 and 30 the relevant variables). Dotted vertical black lines represent the variable used to generate the time-to-event variable. The lower the rank the more relevant the variable is for the specific algorithm

best identified the relevant variables. The other three algorithms were not able to detect the three variables, although RFE-maxgrowth-function was able to identify as relevant, with a similar rank, variables 1 to 8 (highly correlated among them).

In Scenario 6 (Fig. 10), Variable 1 and Variable 30 were selected as relevant; being the former included as main effect with a quadratic term and the latter exponentiated. All methods, except RFE-maxgrowth-function, were able to detect the importance of Variable 30. With respect to Variable 1, RFE-pseudo-samples and RFE-Maxgrowth-function yielded a similar rank of approximately 10.5. The other two algorithms, RFE-Guyon and RFE-maxgrowth-prediction, were not able to identify as relevant Variable 1 with the ranks for this variable comparable to to other non-relevant variables.

Visualization of proposed methods

RFE-pseudo-samples

An example of the results for Scenario 2 (all other scenarios are included as Additional files, from Additional Files 4, 5, 6, 7, 8 and 9), the 100 simulated datasets and first iteration of the RFE algorithm is shown in Fig. 11. Two variables show a completely different pattern from the others: Variable 29 and Variable 30. The association with the response of them was a mirror image of each other: for Variable 30, the larger the pseudo-sample value the larger the decision value and for Variable 29, the larger the pseudo-sample the lower the decision value. The other variables are pretty constant along the pseudo-samples range.

RFE-KPCA-maxgrowth prediction and function

Figure 12 shows an example of RFE-maxgrowth-prediction algorithm, Scenario 1, and iteration 25. To make the plot more interpretable, we only displayed the 5 variables selected as the most relevant: 1, 2, 25, 26 and 28. The first two were highly correlated (in average, a 0.8 Pearson correlation) and the others were independent by design. The reference is the prediction approach, but it is equivalent to function approach. The first component (PC1) is the one that classifies the event group, most events are negative and non-events are positive. For the reference, the directions are going from non-event to event along the PC1 and PC2. With respect to the other variables, only Variable 1 and Variable 2 present a pattern in terms of directions for each observation similar to the reference. Variables 25, 26 and 28 look pretty random. The interpretation of this is: variables 1, 2 and the reference perform similarly, thus, Variable 1 and Variable 2 are relevant and the others are not. Besides that, since 25, 26 and 28 directions are random between them, they are not associated with the response and they are not correlated, which is true by the data generation mechanism.

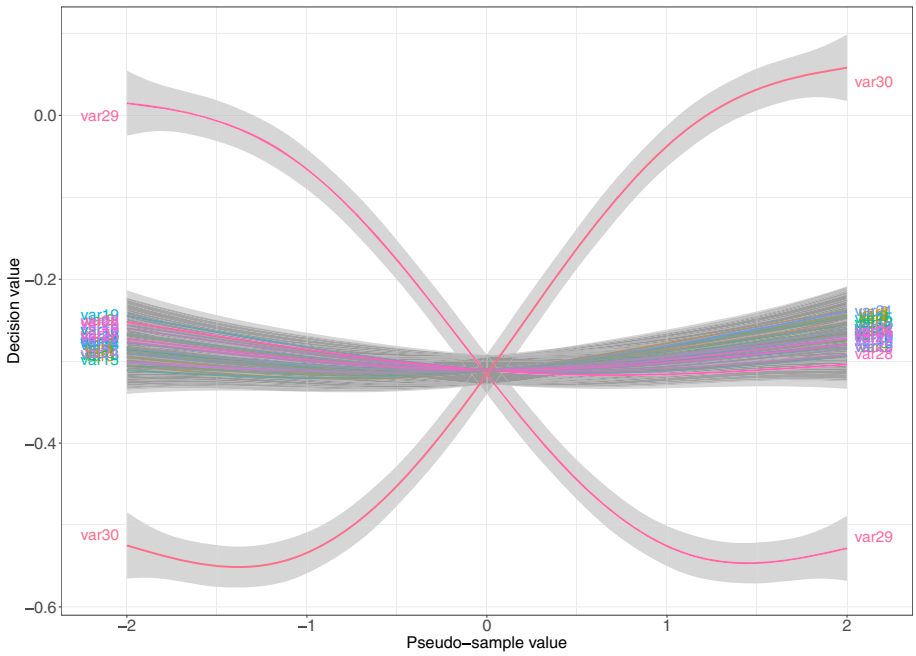

Fig. 11 Visualization of RFE-pseudo-samples results for Scenario 2. Results for Scenario 2 (in which variables 29 and 30 were the relevant variables) over all 100 simulated datasets, all 30 variables, and first iteration of the RFE-pseudo-samples algorithm. The pseudo-samples distribution for each variable is shown with a non-parametric local regression estimation (LOESS) with the corresponding 95% confidence interval

Real-life datasets

In Fig. 13 the Spearman correlation between each method comparing the obtained ranks for each one of the variables in the three dataset is shown. In all three compared real datasets the RFE-pseudo-samples and RFE-maxgrowth-prediction were the methods most correlated with the Cox model. In the Additional Files 10, 11 and 12, the rank comparison between each method and PBC, DLBCL and Lung datasets, respectively, is presented.

From Figs. 14, 15 and 16 the C statistic results by method and real dataset are shown. The RFE-pseudo-samples method discriminative ability is better than the other ones, especially in the DLBCL and PBC dataset, were the C statistics of the top ranked variables (the ones classified by the algorithm as more relevants) are larger. The RFE-maxgrowth methods perform slightly better than the RFE-Guyon except in DLBCL dataset (Fig. 16) were RFE-Guyon performance is overall better being the C statistic better in larger ranks.

Discussion

In biomedical research, it is important to select the variables most associated with the studied outcome and to learn about the strength of this association. In SVM with non-linear kernels, variable selection is particularly challenging because the feature and input spaces are different, thus learning about variables in the feature space does not address the main question about variables in the original space. Although non-linear kernels, specially the Gaussian kernel, are widely used, little work has been done comparing methods to select variables in SVM with non-linear kernels. Moreover, almost no work has focused on interpretation and visualization of the association predictor-response in SVM with linear or non-linear kernels to help the analyst to not only select variables but also learn about the strength and direction of the association. The algorithms we proposed here for SVM aimed to fill this gap and allow analysts to use SVM to better address common scientific questions, i.e.: select variables when using non-linear kernels and learn about the strength of associations of predictor-response. Moreover, the algorithms presented are applicable for analysis of time-to-event responses that are often the primary outcomes in biomedical research.

The three algorithms we proposed performed generally better than the gold standard RFE-Guyon for non-linear kernels. As expected, results for all methods were better when the true relevant variables were independent, i.e., they were no correlated with the other variables in the SVM model. However, this scenario is rarely the case in biomedical research, particularly when analysis includes several variables. Generally, the RFE-pseudo-samples outperformed the other three methods in all tested scenarios. Additionally, the RFE-pseudo-samples algorithm rendered a more friendly visualization of results than RFE-Guyon.

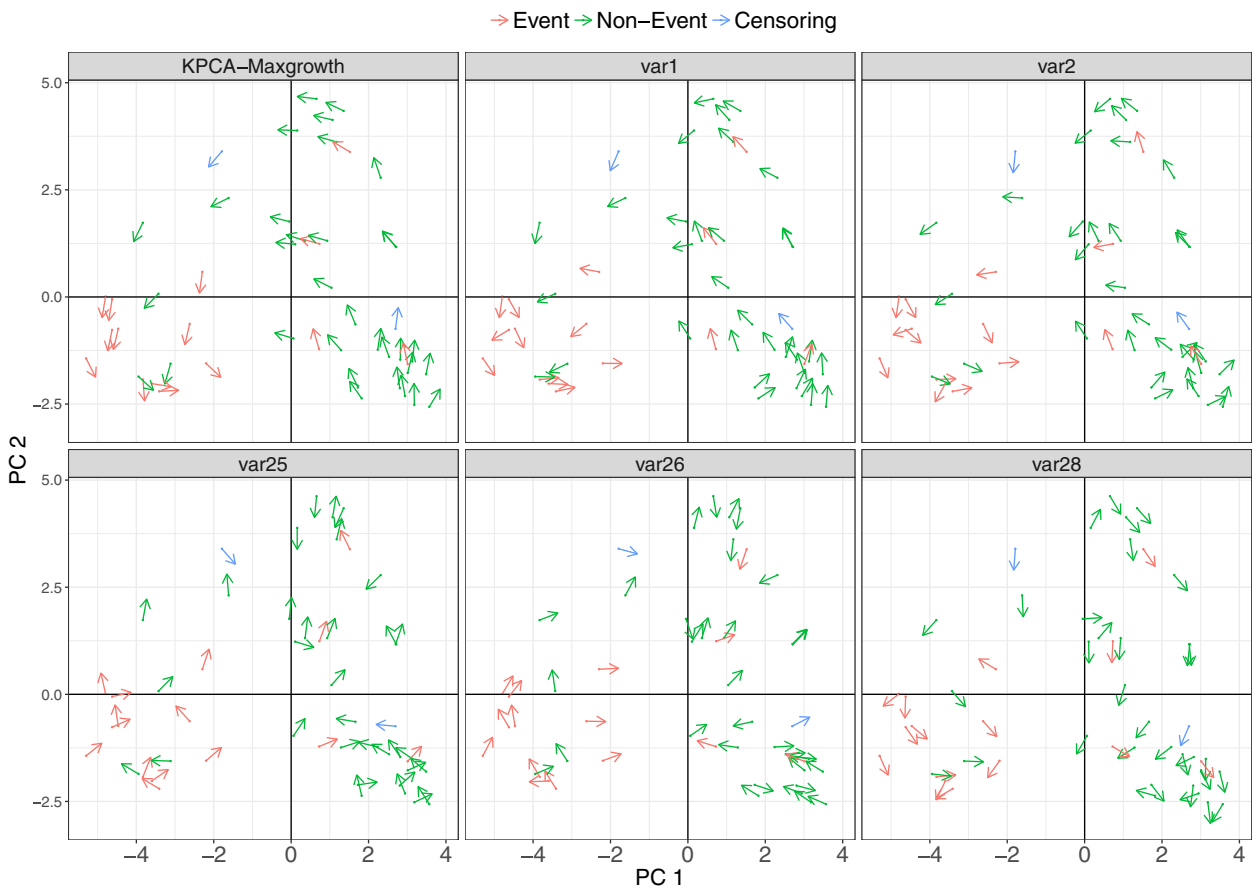

Fig. 12 Visualization of RFE-KPCA-maxgrowth results for Scenario 1. Scenario 1 (being Variable 1 the relevant variable) results for a random simulated dataset and iteration 25 of the RFE-KPCA-maxgrowth-prediction approach. The first component of the KPCA (PC1) is represented in the X-axis and the second component (PC2) is represented in the Y-axis. Events, non-events (censored at the end of follow-up time) and losses to follow-up (censored during follow-up) are represented by red, green and blue color, respectively

With regards to the RFE-maxgrowth, both prediction and function approaches performed similarly. The prediction approach identified the relevant variables better than the fuction approach and the function was less time consuming. The prediction approach can be interpreted as an instance of the function. Although the RFE-maxgrowth-function was based on the explicit decision function and, thus, was expected to outperform the other three approaches, it did not perform as accurately as the other three approaches. One explanation could be that by

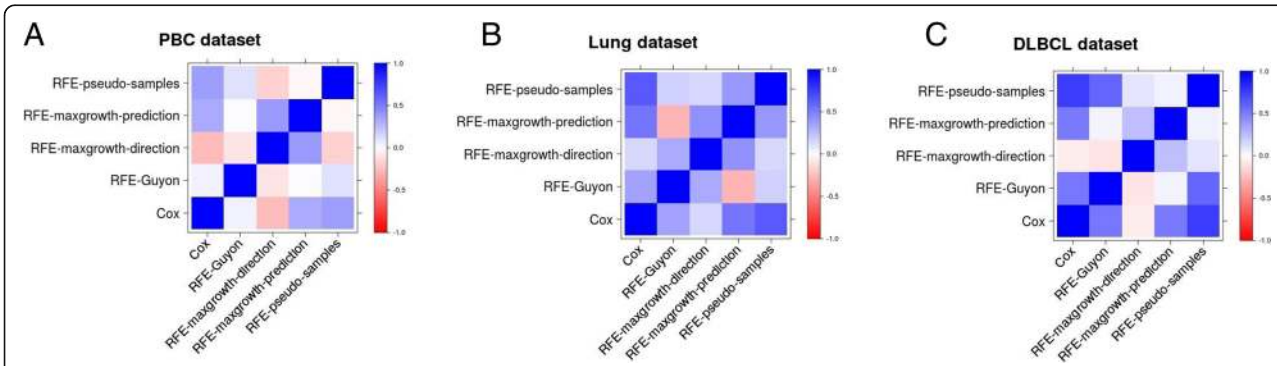

Fig. 13 Spearman correlation matrix comparing 5 methods in the **a** PBC, **b** Lung and **c** DLBCL datasets. The Spearman correlation was computed comparing the ranks obtained by each one of the variables in the dataset

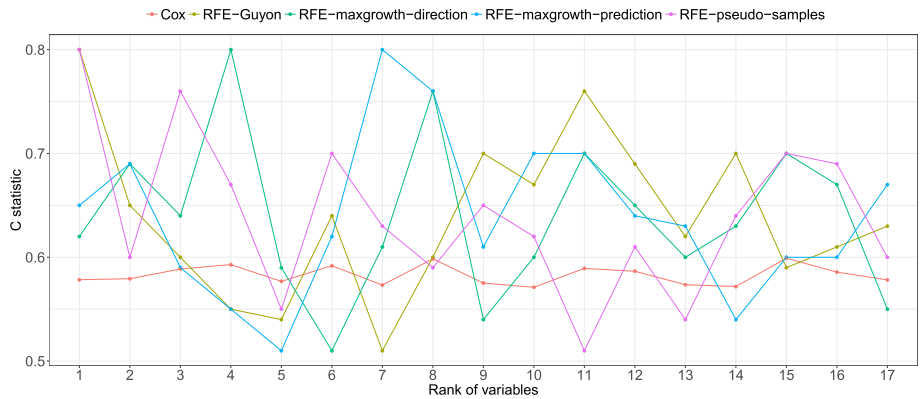

Fig. 14 Discriminative ability, measured as C statistic, by method and ranked variable in the PBC dataset. The X-axis shows the rank of each one of the variables in the dataset after applying the RFE algorithm. The lower the rank the more relevant the variable is and the larger the C statistic is expected. As each method can rank differently the variables, given a rank the variable can be different between methods, due to this the C statistic (Y-axis) is different

approaching the decision function with a non-linear kernel as a combination of variables we are loosing more information than by using the RFE-maxgrowth-prediction.

In the RFE-maxgrowth-prediction algorithm, the prediction was included as an extra variable into the KPCA space. When including this extra variable, the constructed space accounts for the patterns that define event and non-event into the KPCA and is different from the constructed space ignoring the prediction variable. However, in the RFE-maxgrowth-function the KPCA space does not take into account any specific variable directly related to the classes.

The interpretation of the RFE-maxgrowth algorithm is more complex than the RFE-pseudo-samples algorithm because it includes interpretation of the components of the KPCA, the directions of maximum growth of each input variable, and the comparison of the direction of the maximum growth of the input variables between the event and non-events. Although this approach is more informative, it can only be interpreted for a reduced number of variables.

When analyzing the three real datasets the three SVM methods performed overall better than Cox model which is the classical statistical model to analyze time-to-event data. Moreover, the three real datasets fit in terms of sample size and number of variables into the Cox assumptions. Within the proposed methods the RFE-pseudo-samples performed better than the others, being the top-ranked variables the ones with largest discriminative power. The RFE-maxgrowth methods performed slightly better than RFE-Guyon. The obtained results in the real datasets are consistent with the ones obtained in the simulation study.

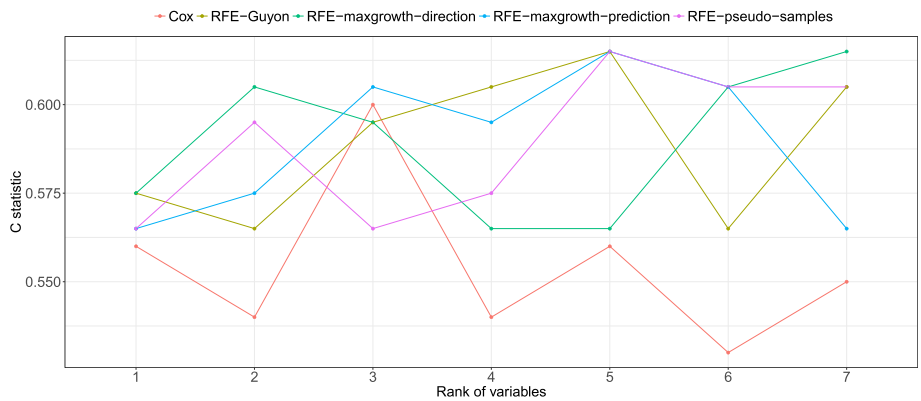

Fig. 15 C statistics results by method and ranked variable in the Lung dataset. The X-axis shows the rank of each one of the variables in the dataset after applying the RFE algorithm. The lower the rank the more relevant the variable is and the larger the C statistic is expected. As each method can rank differently the variables, given a rank the variable can be different between methods, due to this the C statistic (Y-axis) is different

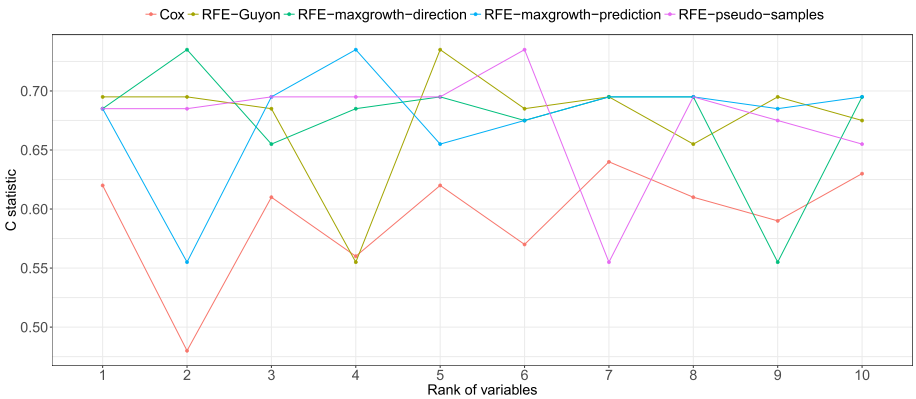

Fig. 16 C statistics results by method and ranked variable in the DLBCL dataset. The X-axis shows the rank of each one of the variables in the dataset after applying the RFE algorithm. The lower the rank the more relevant the variable is and the larger the C statistic is expected. As each method can rank differently the variables, given a rank the variable can be different between methods, due to this the C statistic (Y-axis) is different

The main limitation of the proposed methods is that they are more computationally intensive than classical RFE-Guyon. That could be a limitation depending on the size of the database, the proportion of censored observations during the follow-up period or the SVM extension model used to analyze the time-to-event data. However, this shouldn't be an extra complexity point when analyzing binary response data with no censored obsevations.

Further extensions of the presented work are the comparisons of the proposed methods with other machine learning agorithms used to identify relevant variables such as Random Forest, Elastic Net or Correlation-based Feature Selection evaluator, by analyzing simulated scenarios and real datasets. Additionaly, future work should focus in another important part of the identification of relevant features which is finding the method with largest accuracy or discriminatory ability and not only the identification of the true relevant variables.

Conclusion

Conducting variable selection and interpreting associations between predictors and response variables with the proposed approaches when analyzing biomedical data using SVM with non-linear kernels has some advantadges over the currently available RFE of Guyon. Additionally, the proposed approaches can be implemented with high level of accuracy and speed, and with low computational cost, particularly when using the RFE-pseudo-samples algorithm. Although the proposed methods had more difficulties to identify relevant variables when those variables were highly correlated, they performed better than the classical RFE algorithm with non-linear kernels proposed by Guyon.

Additional files

Additional file 1: Kernel feature space and kernel principal component analysis methodology. (DOCX 42 kb)

Additional file 2: Pearson correlation matrix of the 30 variables simulated. (PDF 131 kb)

Additional file 3: Probabilistic support vector machine methodology. (DOCX 36 kb)

Additional file 4: Visualization of RFE-pseudo-samples results for Scenario 1. Scenario 1 (being Variable 1 the relevant variable) results for all 100 simulated datasets, all 30 variables and first iteration of the RFE-pseudo-samples algorithm. The pseudo-samples distribution for each variable is shown with a non-parametric local regression estimation (LOESS) with the corresponding 95% confidence interval. (PDF 61 kb)

Additional file 5: Visualization of RFE-pseudo-samples results for Scenario 2. Scenario 2 (being Variable 29 and 30 the relevant variables) results for all 100 simulated datasets, all 30 variables and first iteration of the RFE-pseudo-samples algorithm. The pseudo-samples distribution for each variable is shown with a non-parametric local regression estimation (LOESS) with the corresponding 95% confidence interval. (PDF 59 kb)

Additional file 6: Visualization of RFE-pseudo-samples results for Scenario 3. Scenario 3 (being Variable 1, 8, 20, 29 and 30 the relevant variables) results for all 100 simulated datasets, all 30 variables and first iteration of the RFE-pseudo-samples algorithm. The pseudo-samples distribution for each variable is shown with a non-parametric local regression estimation (LOESS) with the corresponding 95% confidence interval. (PDF 57 kb)

Additional file 7: Visualization of RFE-pseudo-samples results for Scenario 4. Scenario 4 (being Variable 1 and 2 the relevant variables) results for all 100 simulated datasets, all 30 variables and first iteration of the RFE-pseudo-samples algorithm. The pseudo-samples distribution for each variable is shown with a non-parametric local regression estimation (LOESS) with the corresponding 95% confidence interval. (PDF 61 kb)

Additional file 8: Visualization of RFE-pseudo-samples results for Scenario 5. Scenario 5 (being Variable 1, 20 and 30 the relevant variables) results for all 100 simulated datasets, all 30 variables and first iteration of the RFE-pseudo-samples algorithm. The pseudo-samples distribution for each variable is shown with a non-parametric local regression estimation (LOESS) with the corresponding 95% confidence interval. (PDF 53 kb)

Additional file 9: Visualization of RFE-pseudo-samples results for Scenario 6. Scenario 6 (being Variable 1 and 30 the relevant variables) results for all 100 simulated datasets, all 30 variables and first iteration of

the RFE-pseudo-samples algorithm. The pseudo-samples distribution for each variable is shown with a non-parametric local regression estimation (LOESS) with the corresponding 95% confidence interval. (PDF 60 kb)

Additional file 10: Results for PBC dataset comparing the four RFE algorithms and the Cox model. (PDF 6 kb)

Additional file 11: Results for DLBCL dataset comparing the four RFE algorithms and the Cox model. (PDF 5 kb)

Additional file 12: Results for Lung dataset comparing the four RFE algorithms and the Cox model. (PDF 5 kb)

Abbreviations
KPCA: Kernel principal component analysis; RFE: Recursive Feature Elimination; SVM: Support vector machines

Acknowledgements
Not applicable.

Funding
This work was funded by Grant MTM2015–64465-C2–1-R (MINECO/FEDER) from the Ministerio de Economía y Competitividad (Spain) to JMO and EV. The funding didn't play any role in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

Availability of data and materials
Simulated datasets during the current study are available from the corresponding author on reasonable request.
PBC and Lung datasets are freely available at https://CRAN.R-project.org/package=survival.
DLBCL dataset is freely available at https://CRAN.R-project.org/package=ipred.

Authors' contributions
HS designed the study and carried out all programming work. FR supervised and provided input on all aspects of the study. CV provided helpful information from the design of the study perspective. FR, EV and JMO contributed algorithms for kernel methods. HS and FR discussed the results and wrote the manuscript. All authors have read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Author details
[1]Department of Genetics, Microbiology and Statistics, Faculty of Biology, Universitat de Barcelona, Diagonal, 643, 08028 Barcelona, Catalonia, Spain. [2]Department of Osteopathic Medical Specialties, Michigan State University, 909 Fee Road, Room B 309 West Fee Hall, East Lansing, MI 48824, USA. [3]Department of Immunology and Infectious Diseases, Harvard T.H. Chen School of Public Health, 675 Huntington Ave, Boston, MA 02115, USA. [4]Centre for Genomic Regulation (CRG), The Barcelona Institute for Science and Technology, Dr. Aiguader 88, 08003 Barcelona, Spain.

Received: 7 May 2018 Accepted: 30 October 2018
Published online: 19 November 2018

References
1. Guyon I, Weston J, Barnhill S, Vapnik V. Gene selection for cancer classification using support vector machines. Mach Learn. 2002;46:389–422.
2. Chen Y-W, Lin C-J: Combining SVMs with various feature selection strategies. In Feature extraction. Berlin, Heidelberg: Springer; 2006:315–324.
3. Maldonado S, Weber R. A wrapper method for feature selection using support vector machines. Inf Sci. 2009;179:2208–17.
4. Aytug H. Feature selection for support vector machines using generalized benders decomposition. Eur J Oper Res. 2015;244:210–8.
5. Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T, Vapnik V. Feature selection for SVMs. In: Proceedings of the 13th International Conference on Neural Information Processing Systems: Neural information processing systems Foundation. Cambridge: MIT Press; 2000. vol. 13, p. 647–53.
6. Benders JF. Partitioning procedures for solving mixed-variables programming problems. Numer Math. 1962;4:238–52.
7. Becker N, Werft W, Toedt G, Lichter P, Benner A. penalizedSVM: a R-package for feature selection SVM classification. Bioinformatics. 2009;25:1711–2.
8. Becker N, Toedt G, Lichter P, Benner A. Elastic SCAD as a novel penalization method for SVM classification tasks in high-dimensional data. BMC Bioinformatics. 2011;12(1):138.
9. Saeys Y, Inza I, Larrañaga P. A review of feature selection techniques in bioinformatics. Bioinformatics. 2007;23:2507–17.
10. Liu Q, Chen C, Zhang Y, Hu Z. Feature selection for support vector machines with RBF kernel. Artif Intell Rev. 2011;36:99–115.
11. Alonso-Atienza F, Rojo-Álvarez JL, Rosado-Muñoz A, Vinagre JJ, Garc'\ia-Alberola A, Camps-Valls G. Feature selection using support vector machines and bootstrap methods for ventricular fibrillation detection. Expert Syst Appl. 2012;39:1956–67.
12. Krooshof PWT, Üstün B, Postma GJ, Buydens LMC. Visualization and recovery of the (bio) chemical interesting variables in data analysis with support vector machine classification. Anal Chem. 2010;82:7000–7.
13. Postma GJ, Krooshof PWT, Buydens LMC. Opening the kernel of kernel partial least squares and support vector machines. Anal Chim Acta. 2011; 705:123–34.
14. Ruppert D. Statistics and data analysis for financial engineering. Springer: New York; 2011.
15. Leys C, Ley C, Klein O, Bernard P, Licata L. Detecting outliers: do not use standard deviation around the mean, use absolute deviation around the median. J Exp Soc Psychol. 2013;49:764–6.
16. Reverter F, Vegas E, Oller JM. Kernel-PCA data integration with enhanced interpretability. BMC Syst Biol. 2014;8(2):S6.
17. Scholkopf B, Smola AJ. Learning with kernels: support vector machines, regularization, optimization. MIT Press: Cambridge; 2001.
18. Scholkopf B, Mika S, Burges CJC, Knirsch P, Muller K-R, Ratsch G, Smola AJ. Input space versus feature space in kernel-based methods. IEEE Trans Neural Netw. 1999;10:1000–17.
19. Bender R, Augustin T, Blettner M. Generating survival times to simulate cox proportional hazards models. Stat Med. 2005;24:1713–23.
20. Shiao H-T, Cherkassky V. SVM-based approaches for predictive modeling of survival data. In: In Proceedings of the International Conference on Data Mining (DMIN); 2013. p. 1.
21. Niaf E, Flamary R, Lartizien C, Canu S. Handling uncertainties in SVM classification. In: Statistical Signal Processing Workshop (SSP); 2011. p. 757–60.