BIOINFORMATICS

Vol. 00 no. 00 2009
Pages 1–8

Permutation importance: a corrected feature importance measure

André Altmann [1,*], Laura Toloşi [1,*], Oliver Sander [1,†], Thomas Lengauer [1]

[1]Department of Computational Biology and Applied Algorithmics, Max Planck Institute for Informatics, Saarbrücken, Germany.

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: In life sciences, interpretability of machine learning models is as important as their prediction accuracy. Linear models are probably the most frequently used methods for assessing feature relevance, despite their relative inflexibility. However, in the past years effective estimators of feature relevance have been derived for highly complex or non-parametric models like Support Vector Machines and RandomForest models. Recently, it has been observed that RandomForest models are biased in such a way that categorical variables with a large number of categories are preferred.

Results: In this work we introduce a heuristic for normalising feature importance measures that can correct the feature importance bias. The method is based on repeated permutations of the outcome vector for estimating the distribution of measured importance for each variable in a non-informative setting. The p-value of the observed importance provides a corrected measure of feature importance. We apply our method to simulated data and demonstrate that I) non-informative predictors do not receive significant p-values, II) informative variables can successfully be recovered among non-informative variables and III) p-values computed with permutation importance are very helpful for deciding the significance of variables and therefore improve model interpretability. Furthermore, permutation importance was used to correct RandomForest based importance measures for two real-world case studies. We propose an improved RandomForest model that uses the significant variables with respect to the permutation importance measure and show that its prediction accuracy is superior to that of other existing models.

Availability: R code for the method presented in this article is available at:
`http://www.mpi-inf.mpg.de/~altmann/download/PIMP.R`

Contact: laura.tolosi@mpi-inf.mpg.de, altmann@mpi-inf.mpg.de

1 INTRODUCTION

In recent years statistical learning has gained increased attention in a large number of research fields. There exist two main goals for the application of statistical learning: either the generation of a (possibly black box) model that predicts a variable of interest given a number of putatively predictive features, or the generation of insight into how the predictive features impact on the variable of interest (given that the prediction model performs reasonably well). This latter task of feature discovery or feature ranking is the essence of biomarker discovery in bioinformatics and life sciences, for instance. Unfortunately, not all statistical learning methods can be used for identifying interesting features because their underlying methods are too complex to analyze contributions of single covariates to the overall results. This problem applies, for instance, to artificial neural networks and support vector machines (SVMs) with non-trivial kernels. However, in the case of SVMs recently approaches to interpreting models that apply sequence kernels were presented (Sonnenburg *et al.*, 2008).

In life sciences, the most frequently applied methods for quantifying feature importance are linear models and decision trees. Linear SVM and linear logistic regression are well studied theoretical models that can provide interpretable classification rules via model parameters. Moreover, in difficult situations when the number of predictors exceeds greatly the number of available samples, regularizers such as the Lasso penalty can be used for obtaining sparse models. However, linear classifiers fail to discover complex dependencies in the training data. This is clearly a drawback when biological data are analyzed, since biological processes usually involve intricate interactions.

Decision trees are suitable for finding nonlinear prediction rules that are also interpretable, although their instability and lack of smoothness have been a cause of concern (Hastie *et al.*, 2001). The RandomForest (Breiman, 2001) classifier was designed to overcome these problems and recently became very popular because it combines the interpretability of decision trees with the performance of modern learning algorithms like artificial neural networks and SVMs. The author of RandomForest proposes two measures for feature ranking, the *variable importance* and *Gini importance*. A recent study showed that, if predictors are categorical, both measures are biased in favor of variables taking more categories (Strobl *et al.*, 2007). The authors of the article ascribe the bias to the use of bootstrap sampling and Gini split criterion for training CART trees (Breiman *et al.*, 1984). In the literature, the bias induced by the Gini coefficient has been reported for years (Bourguignon, 1979; Pyatt *et al.*, 1980), and it affects not only categorical variables but also grouped variables (i.e. values of the variable cluster into well separated groups – e.g. multimodal Gaussian distributions), in general. In biology, predictors often have categorical or grouped

[*]to whom correspondence should be addressed; these authors contributed equally to the article

[†]Current address: Oliver Sander, Novartis Pharma AG, Novartis CampusWSJ-27.6.80, CH-4056 Basel, Switzerland

values (e.g. microarrays, sequence mutations). Strobl *et al.* (2007) propose a new algorithm (cforest) for building RandomForest models based on conditional inference trees (Hothorn *et al.*, 2006) and computing variable importance values that correct for the bias.

Learning on biological data is often characterized by a large number of features and few available samples. A common practice is filtering out unimportant features prior to model fitting, for example, by rejecting the ones that least associate with the outcome. Mutual Information is one measure of association frequently used in this context (Guyon and Elisseeff, 2003). It is closely related to the Gini index and it has been proven also to be biased in favor of variables with more categories (Achard *et al.*, 2005).

In this article, we introduce a heuristic for correcting biased measures of feature importance, called *permutation importance* (PIMP). The method normalizes the biased measure based on a permutation test and returns significance p-values for each feature. In order to preserve the relations between features, we use permutations of the outcome. We show that this method can be used to correct for the bias of feature importance computed with RandomForest and Mutual Information. Moreover, our method can be used together with any learning method that assesses feature relevance, providing significance p-values for each predictor variable. Permutation tests have been previously proposed for assesing significance of feature relevance given by Mutual Information (François *et al.*, 2006), but the authors did not demonstrate that the bias towards features with many categories is alleviated by the procedure.

The paper is organized as follows: in the Methods section we introduce background on RandomForest models and the Mutual Information measure and then present our method for correcting feature importance. The Data section contains a detailed description of the simulated and real data that were used for validating our method. In the Results section we show that our method corrects successfully for the bias of feature ranking pertaining to RandomForest and Mutual Information. We also introduce an improved RandomForest model termed PIMP-RandomForest whose computation is based on the significant features and which incurs clear improvement in prediction accuracy. Finally, we demonstrate the effectiveness of our method on two real world datasets. In the Discussion section we argue that our method can be used in combination with any learning method that provides feature ranking, because it assigns significance p-values to each variable, which improves model interpretability.

2 METHODS

2.1 RandomForest classifier

RandomForest models (Breiman, 2001) use bagging (bootstrap aggregating) of decision trees in order to reduce variance of single trees and thus improve prediction accuracy. They have become a very popular learning method, probably because of their interpretability, in spite of their nonlinearity. Typically, a collection of T decision trees using CART methodology Breiman *et al.* (1984) are trained on T bootstrap samples of the data, respectively. At each node of each tree, a random subset of a fixed size is selected from the features and the one yielding the maximum decrease in Gini index is chosen for the split. The trees are fully grown and left unpruned. The class of a new sample is determined by the majority of the votes of all trees in the RandomForest. The test error of RandomForest models is estimated on the out-of-bag (OOB) data, as follows: after each tree has been grown, the inputs that did not participate in the training bootstrap sample are used as test set, then averaging over all trees gives the test error estimate. Thus, it is possible to avoid the time-consuming cross-validation.

The author of RandomForest proposes two measures for feature importance, the *Variable Importance* (VI) and the *Gini Importance* (GI). The VI of a feature is computed as the average decrease in model accuracy on the OOB samples when the values of the respective feature are randomly permuted. The GI uses the decrease of Gini index (impurity) after a node split as a measure of feature relevance. In general, the larger the decrease of impurity after a certain split, the more informative the corresponding input variable. The average decrease in Gini index over all trees in the RandomForest defines the GI. It should be observed that the Gini index is closely related to the entropy, both being measures of impurity. In this study, we will analyze only the GI measure. The VI was shown to be highly correlated with the GI and shares the same bias (Strobl *et al.*, 2007).

In our simulations, the **R** package `randomForest` has been used for training and evaluating RandomForest models. A few parameters influence the performance of RandomForest models, such as the number of trees in the forest (*ntree*) and the number of variables considered at each split (*mtry*). In our experiments, we use $ntree = 100$ and the recommended value for $mtry = \sqrt{number\ of features}$. Diáz-Uriarte and Alvarez de Andrés (2006) evaluate the performance of RandomForest models depending on *mtry* in ten real-world learning instances. Their results suggest that the default value of *mtry* yields always optimal or close to optimal performance.

2.2 Mutual Information

Mutual information (MI) originates from information theory and measures how much a random variable X is informative about another random variable Y. It is closely related to the concept of entropy. The entropy of a random variable X, denoted traditionally by $H(X)$, measures the level of uncertainty in variable X. It is computed as:

$$H(X) = -\sum_{x} P_X(x) \log P_X(x) \qquad (1)$$

where $P_X(x)$ is the probability distribution of X. The conditional entropy $H(X|Y)$ measures the average of the uncertainty in X given the observed variable Y. Then the mutual information $\mathrm{MI}(X,Y)$ is defined as the decrease in uncertainty about X after observing Y:

$$\mathrm{MI}(X,Y) = H(X) - H(X|Y) \qquad (2)$$

Low, close to zero, MI means that the variables are close to independent. The larger the MI, the larger the reduction of uncertainty in X when Y is known. Mutual information is often used for a quick search of relevant features, when training statistical learning models requires too much computational effort due to the large number of features, e.g. in the case of artificial neural networks (Battiti, 1994). Typically, the MI between each feature and the outcome is computed and a ranking of the inputs results.

For estimating the MI of two vectors, we use the following formula, which is an immediate equivalent transformation of Equation 2:

$$\mathrm{MI}(X,Y) = H(X) + H(Y) - H(X,Y) \qquad (3)$$

Since the probability distributions of X and Y are unknown, in general, we compute frequency-based estimators (Guyon and Elisseeff, 2003).

2.3 Permutation importance (PIMP)

The permutation importance (PIMP) is a heuristic for correcting for the bias of the GI of RandomForest models. The PIMP algorithm can also be used to correct for the bias of the Mutual Information criterion for variable selection.

In a general setting, assume given an algorithm that assesses the relevance of a set of features with respect to a response vector. The PIMP algorithm permutes the response vector s times. For each permutation of the response vector the variable importance for all predictor variables is assessed. This leads to a vector of s importance measures for every variable, which we call the *null importances*. The PIMP algorithm fits a probabilty distribution to the population of null importances, which the user can choose from the following: Gaussian, lognormal, or gamma. Maximum likelihood estimators of the parameters of the selected distribution are computed. Given the fitted distribution, the probability of observing a variable importance of v or higher using the true response vector, can be computed (PIMP p-value). If the user does not know which distribution is most suitable for his or her problem, the PIMP algorithm uses Kolmogorov-Smirnov (KS) tests in order to automatically identify the most appropriate distribution. However, if the tests show little resemblence to any of the three proposed distributions, a non-parametric estimation of the PIMP p-values is used, simply by determining the fraction of null importances that are more extreme than the true importance v (see Algorithm 1 in Supplementary material for an illustration of the PIMP method with Gaussian distribution).

In practical applications the variance of the null importances may be very small and therefore small deviations from the mean lead to artificially boosted variable importances. In order to prevent this artifact, we apply a simple heuristic: variances that are smaller than the mean variance of all variable importances are set to the mean variance.

Permuting the response vector has several advantages. First, the dependence between predictor variables remains unchanged. Second, the number of permutations (s) can be much smaller than the number of predictor variables (p). Third, the approach is general, it can be used together with any method that generates measures for variable importance (biased or unbiased). In this study we demonstrate that PIMP is effective if used with the two mentioned approaches of determining variable importance: GI of RandomForest and MI of predictor variables and the response variable.

2.4 Corrected RandomForest models

The CART methodology uses the Gini index as a criterion for choosing best splits during tree construction and thus the resulting model incorporates the bias of this measure. As a consequence, both the CART and the RandomForest models are both biased themselves, not only their derived feature importance measures. Here, we propose a method for improving the RandomForest models that uses the PIMP algorithm. The method has the following steps: 1) training a classical RandomForest model on the training data; 2) computing the PIMP scores of the covariates; 3) training a new model with the classical RandomForest but now using only the significant variables (w.r.t. PIMP scores), by applying for example the classical 0.05 significance threshold. We will call the improved model *PIMP-RandomForest*. The idea of using the most predictive features for re-training RandomForest model in order to reduce variance and improve accuracy has been proposed previously. For instance, Diáz-Uriarte and Alvarez de Andrés (2006) investigate its benefits on several real-world datasets. The authors show that in some of the instances, the procedure gives good results. However, it may also occur that the RandomForest models built on the reduced set of features exhibit a slightly decreased performance compared to full RandomForest model.

In order to assess the improvement in prediction accuracy of the PIMP-RandomForest model, we use an independent test set and we compute the corresponding error rates. Since the PIMP-RandomForest model uses fewer features than the initial RandomForest, an increase in accuracy can be solely due to decrease of model variance. Thus, we also compare PIMP-RandomForest with classical RandomForest models trained using only the top ranking features of the initial method (biased) as well as with the (corrected) cforest model proposed by Strobl *et al.* (2007) on all features.

3 DATA

3.1 Simulations

Simulation A: For demonstrating the degree of bias in the established measures of importance a dataset comprising 1000 instances was simulated. The predictor variables consist of 31 categorical variables with 2 to 32 categories. The response is a binary variable. Predictor variables and response were independently sampled from a uniform distribution. Since input and output were randomly generated, no predictor variable is informative. Given an unbiased measure of variable importance all variables should receive equally low values. For verification, the GI and MI were computed for each variable. Then, the PIMP of all measures was computed using $s = 100$. The simulation was repeated 100 times.

Simulation B: The second simulation was targeted at the question of how efficiently predictive variables can be recovered among a large set of non-predictive variables. We generated an artificial dataset with a large number of predictors (p) and a small number of samples (n), with $p = 500$ and $n = 100$. In analogy to analyzing aligned amino acid sequence data the variables had 1 to 21 categories (i.e. 20 amino acids and a gap symbol). In the following, variables are referred to as positions (in an alignment) and amino acids denote categories. The number of amino acids for every position was randomly determined, and positions with few different amino acids were more likely than positions with many amino acids. Precisely, a position with m different amino acids had likelihood $1/m \cdot C^{-1}$, with $C = \sum_{i=1}^{21} 1/i$. Moreover, for every position the amino acids were not equally likely, but were sampled from a randomly generated distribution as follows: For each amino acid $j \in \{1, ..., m\}$ at an individual position an integer x_j between 1 and 100 was uniformly sampled. Then the probability of amino acid j at that position was set to $x_j / \sum_{k=1}^{m} x_k$. The output vector comprises two classes that are randomly sampled with probability 0.5. In order to challenge the ability of the feature importance methods to discover the relevant covariates, a number of relevant positions with a small number of categories were intermixed among the non-informative positions as follows: the first 12 positions comprised the same two amino acids and were conditionally dependent (to different degrees) on the binary response variable. Precisely, if the outcome was positive (negative) the amino acid ”a” was sampled with probability $0.5 + r$ $(0.5 - r)$ and amino acid ”b” was sampled with probability of $0.5 - r$ $(0.5 + r)$, where r varied from 0.24 to 0.02 in steps of 0.02. Apart from the first 12 positions all positions were ordered increasingly with respect to the number of amino acids occurring at that position. MI and GI as well as the PIMP scores of these measures with values of $s \in \{10, 50, 100, 500, 1000\}$ were applied for generating feature rankings. An optimal feature ranking method would rediscover all 12 positions that were associated with the outcome. However, since the relation of some positions with the outcome was very weak, these positions were likely to be ranked too low. The simulation was repeated 100 times.

Simulation C: The third simulation aims at showing that p-values greatly improve model interpretation, by adding a statistical significance measure to feature importance. In RandomForest models, importance of variables from a group of highly correlated relevant variables is divided among variables in the group and therefore decreases with the group size. This effect is due to the sampling of features and inputs for the estimation of each tree in the model. We show that the PIMP p-values of correlated variables are significant even when the group size is relatively large. The setting is similar to the simulation B, with $n = 100$, $p = 500$ and the variables having 1 to 21 categories. Again, the binary output vector was randomly sampled from an uniform distribution. The first variable was copied from the output vector only that a random 15% of the binary components were negated. This way, we ensure that the first variable has a high correlation with the outcome and consequently a high relevance. Then, for different values of $k \in \{1, 5, 10, 25, 50\}$, the following k binary variables were constructed such as to be conditionally dependent of the outcome and in addition being mutually correlated. In order to ensure the predictive value of the group, the correlated variables were generated based on a ”seed” variable that was obtained by negating 25% of the outcome components, selected at random. The seed variable is expected to have a correlation coefficient of 0.5 with the outcome. Then, each variable of the correlated group was generated by negating 5% of the components of the seed variable, also randomly selected. Ideally, a learning model would rank the first variable highest, followed by all the k variables in the correlated group, with equal importance independent of the group size (k).

3.2 Real Data

The permutation importance was also evaluated on two real world datasets. The first dataset is concerned with the prediction of sites in the mitochondrial RNA of plants that are edited from cytidine (C) to uridine (U) before translation (C-to-U). The second dataset was collected for answering the question which human chemokine receptor the human immunodeficiency virus (HIV) uses for invading the host cell.

For comparison with previously published methods we reanalyzed the C-to-U dataset published by Cummings and Myers (2004). The dataset comprises 2694 sequences from three different species (*Arabidopsis thaliana*, *Brassica napus*, and *Oryza sativa*). The output vector of the dataset was balanced, i.e. one half of the sequences in the data (1347) were modified at the potential edit site and the other half constitutes a constructed null-set of non-edited sites. Predictive variables were the 20 nucleotides upstream and downstream of the potential edit site, respectively. Additionally, the codon position of the potential edit site (*cp*), the estimated free folding energy of the 41 nucleotide sequence (*fe*) – i.e. the C at the potential edit site and the 20 nucleotides up- and downstream, and the difference in estimated free folding energy between the edited and non-edited version of the 41 nucleotide sequence (*dfe*) were used as predictive variables. The sequence positions comprised up to 5 categories (4 nucleotides; one symbol for ambiguities), *cp* comprised 4 categories (3 positions and none), and *fe* and *dfe* were continuous variables. Using these covariates the aim was to infer whether the cytidine in the center of the sequence was edited or not. In order to compute standard deviations, variable importance was assessed in a 10-fold cross-validation setting by GI and PIMP. For GI RandomForest with 100 and 1000 trees were explored. PIMP was executed with $s = 50$ and a RandomForest size of 100 trees.

The HIV dataset comprised 355 sequences of the Envelope (Env) protein of HIV and the human coreceptors that the virus can use for entering a human host cell. Briefly, during the entry process the glycoprotein gp120, a subunit of Env, attaches to a CD4 receptor and induces a conformational change in the viral protein. Subsequently, the virus needs to bind to a cellular chemokine receptor (coreceptor) for a successful cell entry. The data for this case study were collected from the Los Alamos HIV Sequence Database (http://www.hiv.lanl.gov/). In this analysis only one sequence per patient was used and selected viruses were required to use the CCR5 or CXCR4 coreceptors, i.e. the only coreceptors that are relevant *in vivo*. The predictor variables were the 1030 positions of the multiple amino acid alignment of all 355 sequences, where each position could theoretically take up to 22 different entries (i.e. 20 amino acids, one symbol for ambiguities, and a gap symbol). The binary response variable was defined by the coreceptor usage of the virus. Precisely, negative response was defined as the capability of using the CXCR4 coreceptor, which is associated with advanced stages of the disease. The aim of this analysis was the discovery of amino acid positions that are determinants for the coreceptor usage of the virus. In general, the HIV Env protein contains five loops that are highly variable in sequence, therefore these loop regions are also referred to as variable regions V1 to V5. The V3 loop reached particular interest in the past since it was found to be the major determinant of the virus' coreceptor usage (Lengauer *et al.*, 2007). However, other parts of the Env protein might be associated with the coreceptor usage as well. Moreover, generating a stable alignment in the variable regions is difficult and often leads to alignment positions that take many different amino acids and therefore might artificially boost variable importance. For computing the GI and its standard deviation we use the RandomForest with 500 trees in a 10-fold cross-validation setting and the PIMP algorithm was executed with 50 permutations and 500 trees for every cross-validation model.

4 RESULTS

4.1 Simulations

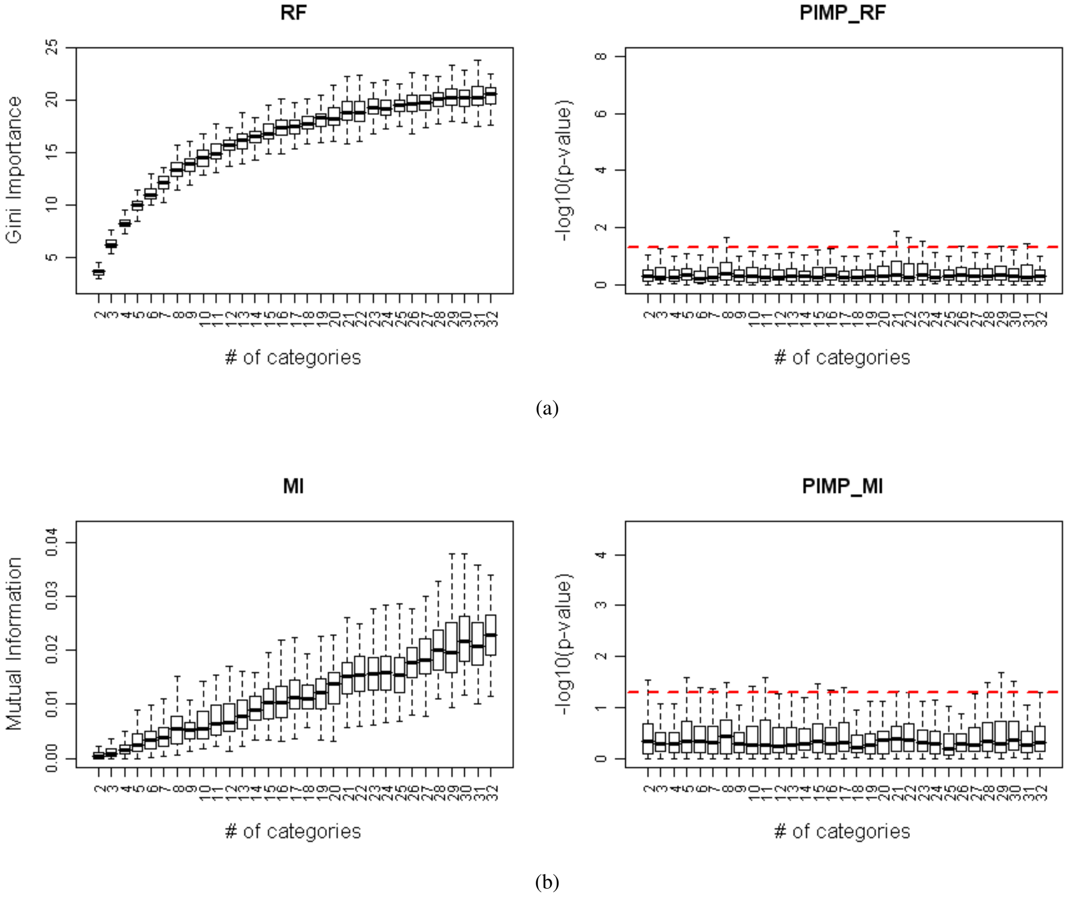
Simulation A demonstrated clearly that MI and RandomForest GI are biased such that variables with a large number of categories receive a higher variable importance (Figure 1 a)-b), left column). In contrast, the PIMP scores (p-values) computed using a gamma distribution (see Figure S1 in Supplementary material for results of the KS tests) for both importance measures are no longer afftected by the bias (Figure 1a)-b), right column). Moreover, none of the candidate variables is significantly dependent on the response variable at a 5% threshold (dashed red line).

Figure 2 a) shows box plots of the RF variable importance computed in the simulation scenario B. The features were ranked with respect to their mean importance in all simulations. For the sake of visualization only the top 25 of the 500 features were displayed. In the first setting, the first 12 variables were selected to be predictive. Using GI (top left), only the first five positions ($r = 0.24$ to $r = 0.16$) were recovered perfectly. By comparison, using the PIMP (gamma distribution; Figure S2 in Supplementary material) of GI with $s = 10$, the first six positions ($r = 0.24$ to $r = 0.14$) were recovered perfectly and positions seven to nine were ranked eighth to tenth. Larger values of s led to perfect recovery of the first eight positions ($r = 0.24$ to $r = 0.10$) and the ninth position ($r = 0.08$) is always among the top 13. MI recovered only the position with the strongest relation ($r = 0.24$) to the response (Figure 2, b)), however wrongly ranked second. This weak performance could already be improved by computing the PIMP of the MI with $s = 10$: the first eight positions were recovered ($r = 0.24$ to $r = 0.10$), and the ninth position ($r = 0.08$) was ranked at position 10. Larger values of s led to perfect recovery of the first nine positions.

Simulation scenario C shows that PIMP p-values can be very useful in learning datasets whose instances entail groups of highly correlated features. As the size of the correlated group increases, the GI of each variable in the group decreases to the point of apparent non-significance. The relative importance of the first feature and correlated group increases with the group size while, in fact, it should remain constant (Figure S4, left column, Supplementary material). When the size of the group is very large ($k = 50$), the common GI is close to zero, which would probably lead to the exclusion of the corresponding variables from the relevance list. In contrast, PIMP (gamma distribution; Figure S3 in Supplementary material) can help determine the relevance of the group. In our simulations, the variables in the correlated group are significant even for a group size as large as 50, which is 10% of the total number of features (Figure S4, right column, Supplementary material).

4.2 Real Data

The RandomForest prediction model achieved a mean area under the ROC curve (AUC) of 0.93 (±0.014) in 10-fold cross-validation. The cforest method yielded only an AUC of 0.89 (±0.023). The box plots in Figure 3 show the variable importance computed from 10 cross-validation runs on the C-to-U dataset. GI was computed from 100 trees and rated the position upstream of the site of interest (-1) as the most informative predictor. This was followed by *fe*, the position after the site of interest, and *dfe*. The importance remained unchanged when a forest of 1000 trees was used to compute the GI (data not shown). However, the PIMP (with $s = 50$; normal



(a)

(b)

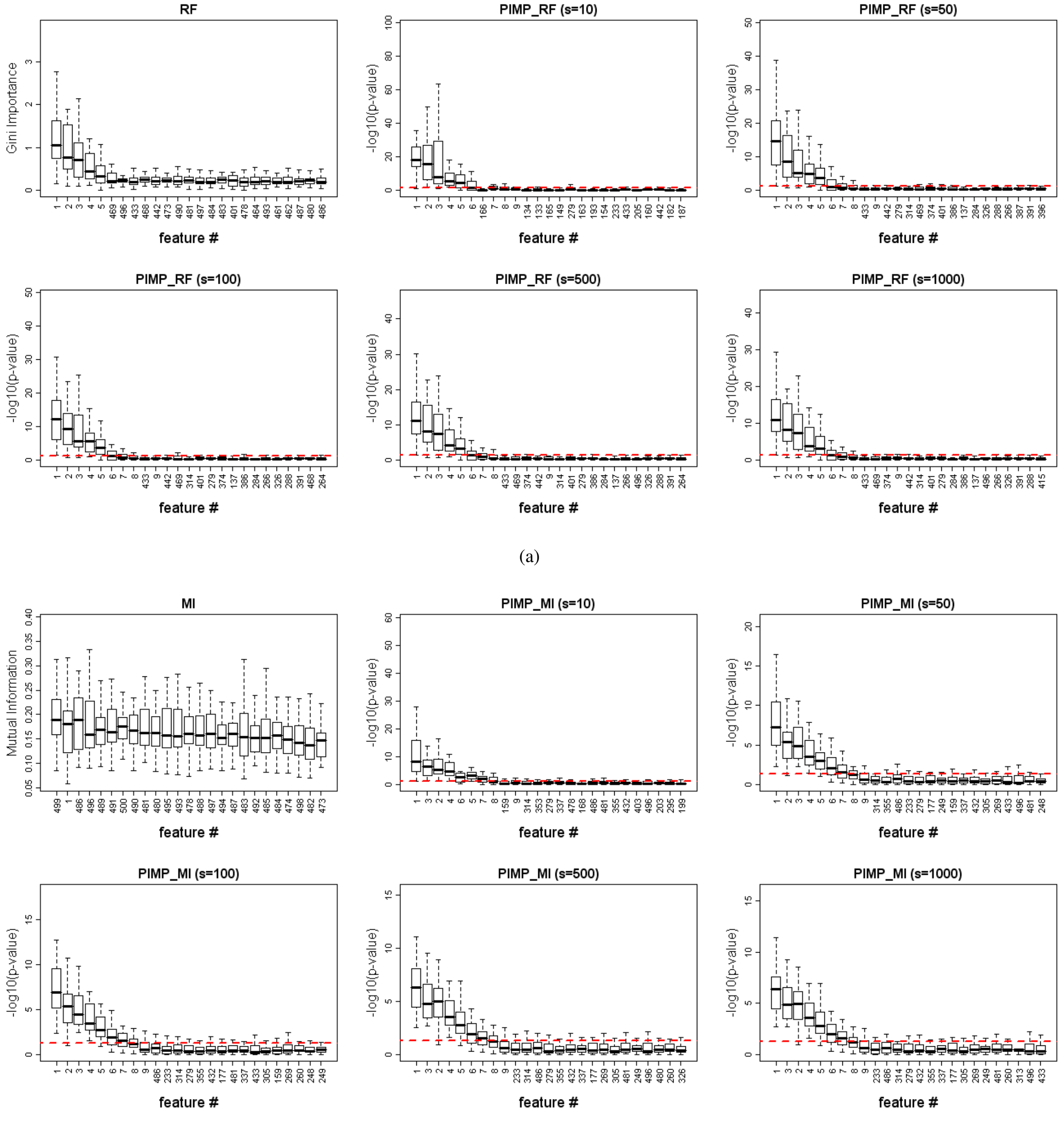
Fig. 1. Simulation A: variable importance in dependence of number of categories: a) GI, b) MI.

distribution; Supplementary material Figure S5) of the GI computed from 100 trees showed a somewhat different picture. Here, the positions adjacent to the site of interest (-1 and 1) were the most informative ones. *fe*, second most important predictor under GI, yielded only moderate importance using PIMP. Moreover, several sequence positions upstream of the site of interest (i.e. -2, -4, and -5) showed higher importance than *fe* using PIMP. Interestingly, all three positions achieved a GI lower than *dfe*, which was rated as completely uninformative by PIMP.

The RandomForest model for predicting HIV coreceptor usage achieved a mean AUC of 0.94 (± 0.029) in 10-fold cross-validation. For comparison, the cforest method yielded an AUC of 0.80 (± 0.014). The box plots in Figure 4 depict the variable importance of all alignment positions in the HIV Env protein in terms of coreceptor usage. Importance was measured with GI (500 trees) and PIMP ($s = 50$ and 500 trees; lognormal distribution; Supplementary material Figure S6). At first glace the GI confirms the importance of the V3 loop for determining coreceptor usage and also suggests that positions in other variable loops (V1, V2, V4, and V5) are associated with coreceptor usage (although at lower levels). Application of PIMP also confirms the important role of V3. In contrast to the GI measure, which suggested that V1 and V2 are equally important, only positions in the variable loop V2 are related to coreceptor usage after the correction with PIMP. Recently, it was shown that incorporation of the V2 sequence information improves the performance of prediction tools for HIV coreceptor usage (Thielen *et al.*, 2008).

4.3 Model improvement

We used simulation B and both real-world case studies to validate our improved PIMP-RandomForest model. For simulation B we ran 100 simulations and compared the accuracy of RandomForest, PIMP-RandomForest, RandomForest re-trained only using the top ranking features and the cforest model. The error rates were computed on an independent test set. Table 1 shows the improvements of accuracy of different methods over the classical RandomForest. The PIMP-RandomForest model performs significantly better than the RandomForest, with an average decrease of OOB error rate of 10%. The RandomForest trained on the top-ranking 1%, 5% and 10% of the features also yields better models, due to the decrease in variance. Choosing the top 5% results in a model with accuracy comparable (although still inferior) to the PIMP-RandomForest. However, it is not clear *a priori* how many top ranking features should be selected for a refined model. With the p-values provided by the PIMP algorithm, one can simply use the classical 0.05 significance threshold for selecting the most relevant



(a)

(b)

Fig. 2. Discovery of relevant features in simulation scenario B: a) GI, b) MI

Table 1. Comparison of different RandomForest models on data from simulation B and both real-world case studies.

	RF baseline	PIMP-RF		RF Top 1%		RF Top 5%		RF Top 10%		cforest	
	error rate	error rate	Δ error rate	error rate	Δ error rate	error rate	Δ error rate	error rate	Δ error rate	error rate	Δ error rate
Sim B	0.35 ± 0.06	0.25 ± 0.05	0.10 ± 0.05	0.27 ± 0.08	0.08 ± 0.07	0.26 ± 0.06	0.09 ± 0.06	0.28 ± 0.06	0.07 ± 0.06	0.32 ± 0.09	0.03 ± 0.07
C-to-U	0.18 ± 0.02	0.22 ± 0.03	-0.04 ± 0.03	0.30 ± 0.03	-0.12 ± 0.03	0.28 ± 0.02	-0.10 ± 0.02	0.24 ± 0.03	-0.06 ± 0.03	0.20 ± 0.03	-0.02 ± 0.03
HIV	0.13 ± 0.04	0.10 ± 0.04	0.03 ± 0.04	0.12 ± 0.04	0.01 ± 0.04	0.11 ± 0.04	0.02 ± 0.04	0.10 ± 0.06	0.03 ± 0.04	0.21 ± 0.09	-0.08 ± 0.08

Performance of different RandomForest models on different datasets. The name of the dataset is given in the first column. The baseline is the classical RandomForest (RF). For comparison, the average error rates and average improvement (Δ error rate) w.r.t. baseline are shown for PIMP-RandomForest(PIMP-RF), for RandomForest models trained on the top ranking 1%, 5% and 10% features and for the cforest algorithm.

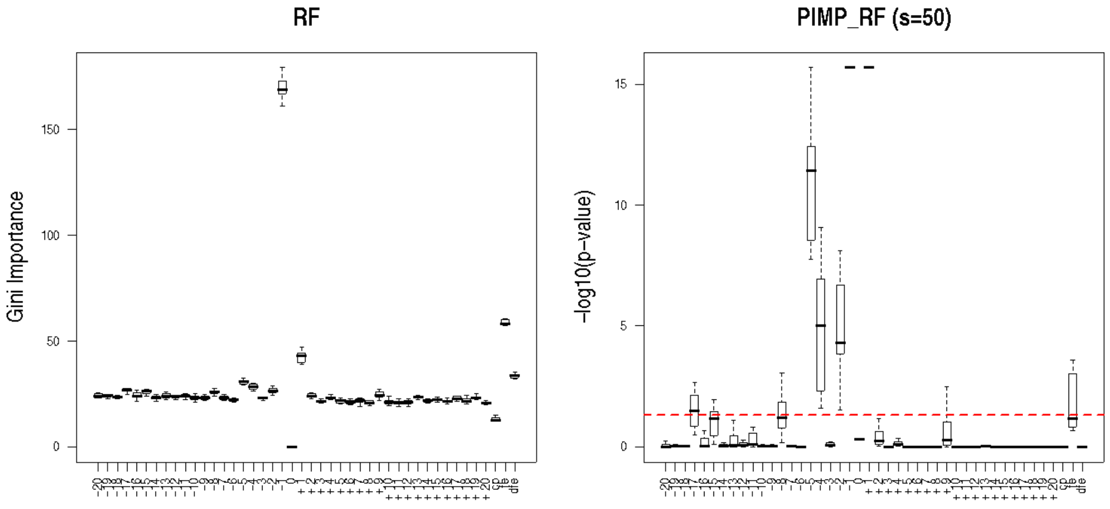


Fig. 3. Variable importance on the C-to-U dataset. GI (left) was computed using 10-fold cross-validation and a RandomForest with 100 trees. PIMP using a normal distribution with $s = 50$ permutations (right) was executed for each cross-validation model.

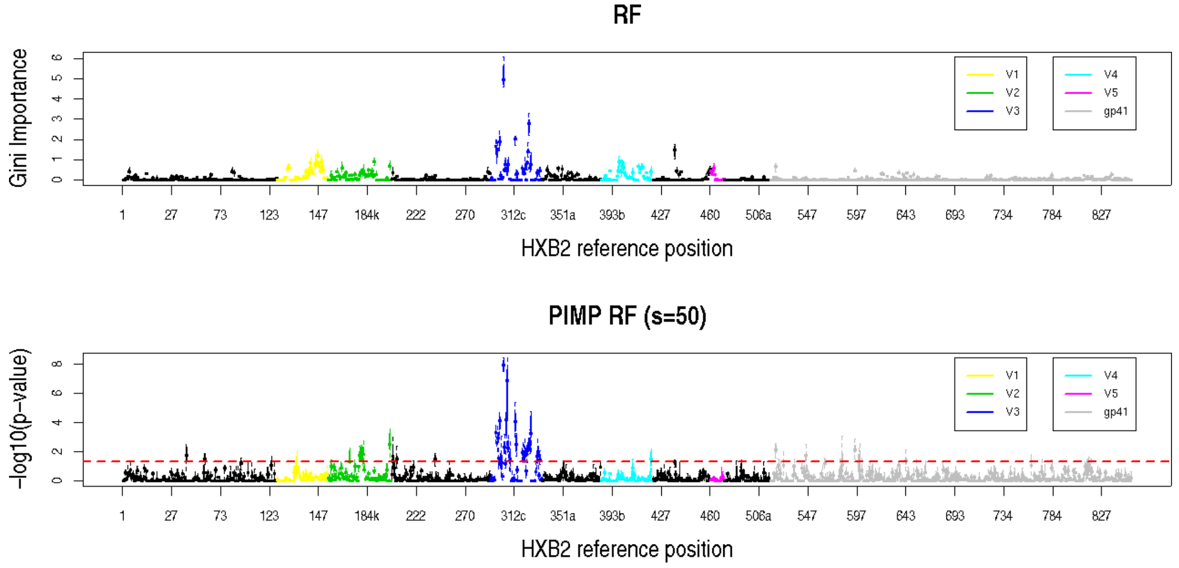


Fig. 4. Variable importance on the HIV dataset. GI (top) was computed using 10-fold cross-validation and a RandomForest with 500 trees. PIMP using a lognormal distribution with $s = 50$ permutations (bottom) was executed for each cross-validation model. Alignment positions are annotated with respect to the HBX2 reference strain (genbank accession number: K03455), i.e. 393b reads as ”second amino acid insertion after amino acid 393 in HXB2”.

variables. Notably, the cforest algorithm is superior to the classical RandomForest, but the average decrease of error rate is significantly smaller than the one achieved by PIMP-RandomForest.

Error rate for the two real-world case studies was determined using 10-fold cross-validation, and feature selection was carried out for each cross-validation model separately. Results are summarized in Table 1. On the HIV dataset, the cforest method exhibits an increased error rate compared to the RF model with all features, while the PIMP-RF shows the best performance together with the RF trained on the top 10% ranked features. In contrast, on the C-to-U dataset, all RF-based models shows an increased error rate compared to the RF model using all features. However, among the RF-based models, PIMP-RF shows the smallest increase in error rate. On this dataset, the cforest method shows the overall slightest increase in error rate. Decrease in performance of RF models with a restricted feature set is not uncommon: for instance, on seven of the 10 microarray datasets in the work of Diáz-Uriarte and Alvarez de Andrés (2006) the restricted RF models perform worse than the full RF model.

5 DISCUSSION

In this work, we proposed an algorithm for correcting for two biased measures of variable importance. The method permutes the response vector for estimating the random importance of a feature. Under the assumption that the random importance of a feature follows some distribution (Gaussian, lognormal or gamma), the likelihood of the measured importance on the unpermuted outcome vector can be assessed. The resulting p-value can serve as a corrected measure of variable relevance. We showed how this method can successfully adjust the feature importance computed with the classical RandomForest algorithm, or with the Mutual Information measure. We also introduced an improved RandomForest model that is computed based on the most significant features determined with the PIMP algorithm.

Simulation A demonstrated that the Gini importance of the RandomForest and Mutual Information favor features with large number of categories and showed how our algorithm alleviates the bias. Simulation B demonstrated the usefulness of the algorithm for generating a correct feature ranking. For all methods, the feature ranking based on the unprocessed importance measures could be improved. When variable importances of RandomForest are distributed among correlated features, our method assigns significant scores to all the covariates in the correlated group, even for very large group size. This improves model interpretability in applications like microarray data classification, where groups of functionally related genes are highly correlated.

Permutation importance was used to correct for RandomForest based Gini Importance measures for two real world datasets. Both cases studies use features based on nucleotide or amino acid sequences. As already discussed by Strobl *et al.* (2007) categorical features (e.g. nucleotide sequences) are often used together with derived continuous features (e.g. free fold energy) for improving the prediction model. In this case it may happen that the continuous variables are preferred by tree-based classifiers as they provide more meaningful cut points for decisions. Permutation importance on the C-to-U dataset demonstrated successful post-processing of the original importance measure (GI). The HIV case study exclusively employed categorical features in the form of amino acids in an alignment. The sequences, however, contained highly variable regions in which many different amino acids were observed in one alignment position. The original RandomForest-based importance measure was successfully corrected for with the proposed method.

We proposed a corrected RandomForest model based on the PIMP scores of the features and we demonstrated that in most of the cases it is superior in accuracy to the cforest model. The major drawback of the PIMP method is the requirement of time-consuming permutations of the response vector and subsequent computation of variable importance. However, our simulations showed that already a small number of permutations (e.g. 10) provided improvements over a biased base method. For stability of the results any number from 50 to 100 permutations is recommended. The algorithm can easily be parallelized, since computations of the random variable importance for every permutation are independent, and therefore allow for an even better scalability with respect to available computational resources. With parallelization, the running time of our algorithm is only a few times longer than the running time of a classical RandomForest, which is very fast even for large instances.

We argue that the PIMP algorithm can also be used as a post-processing step with other learning methods that provide (unbiased) measures of feature relevance, such as linear models, logistic regression, SVM, etc. The raw scores given by these models provide with a feature ranking, but usually it is difficult to choose a significance threshold. The PIMP p-values are easier to interpret and provide a common measure that can be used to compare feature relevance among different models.

ACKNOWLEDGEMENT

We would like to thank Alexander Thielen for helpful discussions on the HIV corecptor case study.

Funding: This work was supported by the German National Genome Research Network (NGFNplus) [01GS08100 to L.T.]; and the Commission of the European Communities [HEALTH-F3-2009-223131 to A.A.].

REFERENCES

Achard, S., Pham D.T. and Jutten, C. (2005) Criteria based on mutual information minimization for blind source separation in post nonlinear mixtures, *Signal Processing*, **85**, 965-974.

Battiti, R. (1994) Using Mutual Information for Selecting Features in Supervised Neural Net Learning, *IEEE Trans. on Neural Networks*, **5**(4), 537-550.

Bourguignon, F. (1979) Decomposable income inequality measures, *Econometrica*, **47**, 901-920.

Breiman, L., *et al.* (1984) Classification and regression trees, Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software, ISBN 978-0412048418.

Breiman, L., (2001) Random Forests, *Machine Learning*, **45**(1), 5-32.

Cummings, M.P. and Myers, D.S. (2004) Simple statistical models predict C-to-U edited sites in plant mitochondrial RNA, *BMC Bioinformatics*, **5**, 132.

Diáz-Uriarte, R. and Alvarez de Andrés, S. (2006) Gene selection and classification of microarray data using random forest, *BMC Bioinformatics*, **7**:3.

François, D., Wertz, V. and Verleysen, M. (2006) The permutations test for feature selection by mutual information, ESANN'2006 Proceedings.

Guyon, I. and Elisseeff, A. (2003) An introduction to variable and feature selection, *The Journal of Machine Learning Research*, **3**, 1157-1182.

Hastie, T., Tibshirani, R. and Friedman, J.H. (2001) The Elements of Statistical Learning, Springer, New York.

Hothorn, T., Hornik, T. and Zeileis, A. (2006) Unbiased Recursive Partitioning: A Conditional Inference Framework, *Journal of Computational and Graphical Statistics*, **15**(3), 651-674.

Lengauer, T., *et al.* (2007) Bioinformatics prediction of HIV coreceptor usage, *Nature Biotechnology*, **25**, 1407-1410.

Pyatt,G., Chen,C. and Fei,J. (1980) The distribution of income by factor components, *The quarterly journal of economics*, **95**, 451-473.

Sonnenburg, S., *et al.* (2008) POIMs: positional oligomer importance matrices – understanding support vector machine-based signal detectors, *Bioinformatics*, **24**, i6-i14.

Strobl, C., *et al.* (2007) Bias in random forest variable importance measures: Illustrations, sources and a solution, *BMC Bioinformatics*, **8**:25.

Thielen, A., *et al.* (2008) Improved genotypic prediction of HIV-1 coreceptor usage by incorporating V2 loop sequence variation, *Antiviral Therapy*, **13 Suppl 3:**A100.