Supplementary Material 1 to the paper:

Interaction Forests: Identifying and exploiting interpretable quantitative and qualitative interaction effects

Roman Hornung*,[1], Anne-Laure Boulesteix[1]

[1] Institute for Medical Information Processing, Biometry and Epidemiology, University of Munich, Marchioninistr. 15, 81377 Munich, Germany

*To whom correspondence should be addressed: hornung@ibe.med.uni-muenchen.de

Contents

A Descriptions of existing work on multivariate trees, multivariate tree ensembles, and approaches to identifying interactions from tree ensembles

A.1 Overview and discussion of multivariate tree approaches

In the following, the term "local split optimization" will denote the process of finding a split that divides the current node optimally with respect to some specified criterion. The term "global split optimization", in contrast, will denote the process of finding the values of all splits in a tree that deliver an optimal tree with respect to a specified criterion. Lastly, multivariate trees with linear decision boundaries will be referred to as "oblique (decision) trees" (Murthy *et al.*, 1994).

An early example of multivariate trees are multivariate CARTs (Breiman *et al.*, 1984). In this procedure, to select each split, a (locally) best split candidate is considered, as well as one multivariable linear split candidate that uses all variables, where the multivariable split candidate is optimised using an adhoc procedure. Utgoff and Brodley (1990) present an algorithm to construct oblique trees that uses the absolute error correction procedure and the pocket algorithm for optimisation. These trees use varying subsets of variables for the splits, where these subsets are obtained via backward selection. The use of backward selection in this context seems problematic because iteratively removing variables can result in members of strongly interacting variable pairs being removed if there are confounding effects caused by the influence of other variables that mask the interaction effects (Gheyas and Smith, 2010). Sethi and Yoo (1994) again use the pocket algorithm in the construction of oblique trees, but in their approach all variables are used in each split. These trees are not likely suitable for use in a random forest. As shown by Breiman (2001), for a good predictive performance of a random forest, it is important that the predictions of the trees are diverse in addition to being precise. The latter is unlikely if all variables are used in each split. For a comparison study of early approaches to construct multivariate trees, see Brodley and Utgoff (1995). Like Breiman *et al.* (1984), Murthy *et al.* (1994) use the best univariable split and an optimised multivariable split, where they employ randomisation to improve the multivariable split. More precisely, they use an algorithm similar to that of Breiman *et al.* (1984), but employ two randomisation procedures to improve the found split. First, they perform multiple searches starting from different random splits and second, they attempt to improve the split resulting from the optimisation algorithm by shifting it into a random direction. They refrain from spending excessive effort on locally optimising splits because even if the best locally optimal split is found, this split will likely not lead to the best possible tree. When the search space holds an abundance of good solutions, a randomised search performs well (Gupta *et al.*, 1994). The algorithm by Murthy *et al.* (1994), however, uses all variables in the multivariable splits by default; the authors suggest using existing variable selection techniques, such as stepwise selection, to choose the relevant variables in the splits.

Wickramarachchi *et al.* (2015) distinguish three types of algorithms for constructing multivariate trees: 1) algorithms that use optimisation techniques for finding the splits; the classical approaches described above fall into this category, a recent example are "optimal classification trees" (Bertsimas and Dunn, 2017) (see further down for details); 2) algorithms that use existing statistical approaches (e.g., linear discriminant analysis) to find splits (Loh and Shih, 1997; Gama

and Brazdil, 1999; Li *et al.*, 2003; Kolakowska and Malina, 2005; López-Chau *et al.*, 2013); 3) algorithms that use heuristic approaches to find splits (Amasyah and Ersoy, 2008; Manwani and Sastry, 2012; Robertson *et al.*, 2013; Wickramarachchi *et al.*, 2015). An interesting exception to this classification are omnivariate decision trees (Yıldız and Alpaydın, 2001). With the latter, for each split different model types are considered, and the data are used to select the best model type. Yıldız and Alpaydın (2001) argue that it is likely beneficial if the first splits in the trees are more complex, whereas the splits closer to the leaf nodes can be simpler. This is because the samples in these nodes are more homogeneous, and because the sample sizes are smaller.

An important recent contribution, mentioned previously, are optimal classification trees (Bertsimas and Dunn, 2017). These trees differ from conventional trees in that the splits in these trees are not found in a sequential manner by local optimisation, but instead, using mixed-integer optimisation, the whole trees are constructed at once in such a way that the training error is minimised. The construction of such optimal trees had not been computationally tangible until recently and is still computationally demanding. Bertsimas and Dunn (2017) present algorithms both for finding optimal oblique classification trees and optimal classification trees that use univariable splitting. Interestingly, these optimal trees do not seem to overfit the training data. They compare both optimal tree versions with univariate and oblique trees that use local optimisation in a large-scale benchmark study. Interestingly, they find that the optimal oblique classification trees outperform the other variants. This observation that oblique trees particularly benefit from global optimisation of the splits might be explained by the fact that multivariable splits are more sophisticated than univariable splits. The great flexibility of multivariable splits makes it likely that the optimal tree is associated with a partition of the variable space that is close to the partition associated with optimal predictive performance. This flexibility may be less beneficial when recursively growing the trees using local optimisation. While the multivariable splits found using local optimisation are likely still better than univariable splits found using local optimisation because the search space associated with multivariable splitting is larger, locally optimal multivariable splits still suffer from the fact that a locally optimal split is rarely globally optimal.

A.2 Random forest-based approaches that use multivariate trees

The underlying concept of rotation forests (Rodríguez *et al.*, 2006) is to learn conventional univariate trees on different transformations of the data set. Each of the variables in these transformed data sets contains information from several of the original variables. More precisely, before training each tree, the available variables are randomly divided into K subsets and principal component analysis (PCA) is applied to each subset and the original data substituted by the principal components. In the latter procedure, the coefficients of the principal components are learned using random subsets of the observations, but the trees are learned using all observations. Rodríguez *et al.* (2006) state that the idea of this proceeding is to attain both accuracy of the tree predictions and diversity between the different trees. Gashler *et al.* (2008) present the Mean Margins Decision Tree Learning (MMDT) algorithm for finding splits that involve all variables in oblique trees and use bagging to form forests of such trees. They recommend using a mixed ensemble of trees that involves both univariate trees and oblique trees constructed using the MMDT algorithm. In the approach "oblique random forests", presented by Menze *et al.* (2011), multivariable splits are

learned using conventional regression methods. These splits do not involve all variables, but subsets of *mtry* variables, sampled randomly at each split as in conventional random forests. Possible choices for the regression method used to learn the multivariable splits are ridge regression, logistic regression, or partial least squares regression. Canonical correlation forests (Rainforth and Wood, 2015) transform the data using canonical correlation analysis before each split and subsequently choose the best axis-aligned splits in the transformed data, which are oblique splits in the original variable space.

A.3 Approaches to identifying interactions from tree ensembles

Ishwaran (2007) considers a new variable importance measure, similar to the classical permutation variable importance of random forests. As a first step towards measuring the importance of the interaction effect between two variables, this new variable importance measure is calculated with respect to perturbing the influence of both variables jointly and with respect to perturbing each of the two variables separately. This results in three values, one measuring the influence of both variables taken together and two measuring the separate influence of each variable. Subsequently, to measure the disparity between the joint influence of the variables and their separate influences, the sum of the two values measuring the individual influences of the variables is subtracted from the value measuring the joint influence of the variables. These differences are denoted "paired association" values in Ishwaran (2007). Both, strongly positive and strongly negative values of the paired association are assumed to be indicative of interaction effects, if the univariable importance values of both involved variables are reasonably large (Ishwaran, 2007; Ishwaran and Kogalur, 2020). A very similar approach is presented by Kelly and Okada (2012). Here they use the classical permutation variable importance and calculate the difference between the sum of the two importance values that measure the individual influences of the two variables and the importance value that measures the joint influence of the two variables. Kelly and Okada (2012) state that positive values of their measure would indicate "positive interactions" and negative values "negative interactions"; however, it is not stated what the terms "positive" and "negative" describe in this context. Bureau *et al.* (2005) had previously considered another approach related to that of Kelly and Okada (2012). Unlike the latter authors, Bureau *et al.* (2005) only used the value measuring the joint of influence of the two variables obtained using the permutation variable importance measure, but they did not adjust that value for the individual, or rather the marginal influences of the two variables. Therefore, the approach by Bureau *et al.* (2005) does not measure the strength of interaction between the two variables, but the strength of the joint effect.

Dazard *et al.* (2018) introduce a new interaction importance measure called Interaction Minimal Depth Maximal Subtree (IMDMS) based on second-order maximal subtrees (Ishwaran *et al.*, 2010). The latter build upon the concept of maximal subtrees (Ishwaran, 2007), which can be used for measuring variable importance. Generally, IMDMS focuses on the minimal distance between splits performed using each of the two members of variable pairs in the hierarchial structures of the trees. This procedure is based on the notion that interacting variables are more often used in quick succession for splitting in trees.

The methods presented above are based on conventional trees that use univariable splitting. However, as noted in the introduction, conventional trees do not sufficiently model interaction

effects between variables whose effects are only strong used simultaneously. Ng and Breiman (2005) and, more recently, Yoshida and Koike (2011), present two approaches that use multivariable splits in the trees. With these approaches, interaction effects are considered directly in the splitting, whereas in the cases of the methods presented above, these effects were only modeled if the corresponding variables were selected for univariable splitting. Ng and Breiman (2005) first form a synthetic variable from each pair with the goal of keeping a large part of the interaction effects information between the two variables intact. Subsequently, the variable importance values of these synthetic variables are calculated and set relative to the variable importance values of both variables used to form the respective synthetic variables. In this approach, it is necessary to categorise each metric variable. Yoshida and Koike (2011) present an approach SNPInterForest, which is a random forest-based interaction detection method specifically for SNP data. As in the case of Ng and Breiman (2005), SNPInterForest forms synthetic variables from all pairs of variables with the goal of keeping the information on the interaction effects between the variables in the synthetic variables intact. However, due to the categorical level of measurement of SNP data, with SNPInterForest it is possible to keep all information intact when forming the synthetic variables. The latter variables have nine categories, where each of these categories corresponds to a specific combination of categories of the two SNP variables, which have three categories. The measure for the degree of interaction between two variables of SNPInterForest is based on the frequencies with which these variables are present in the same branches of the trees. A threshold in the measure values for interaction detection is obtained via simulation results in the paper. A similar idea is considered in Chen and Zhang (2013) who investigate the variable pairs for potential interaction effects by applying Fisher's exact test to determine whether both members in the respective variable pair occur overproportionally often in the same trees.

Li *et al.* (2016), Basu *et al.* (2018), and Jiang *et al.* (2009) present approaches applicable only if the outcome is binary. The interaction importance measure for variable pairs associated with the permuted random forest method by Li *et al.* (2016) focuses on the difference between the prediction error of a random forest expected after removing the interaction effect between the two respective variables, and its prediction error expected when keeping intact both the interaction effect and the corresponding main effects. Here, removing the interaction effect, while keeping the main effects intact, is performed by permuting the values of both variables within each class. Basu *et al.* (2018) introduce a method called iterative random forests for determining high-order interactions among biomolecules. First, a random forest is grown, where the candidate split variables in the trees are drawn with probabilities proportional to weights optimised using an iterative scheme based on the Gini variable importance. Second, a procedure called generalized RIT is applied to the random forest obtained in the first step to determine tuples of variables that are more frequently used jointly to classify observations from one of the two classes rather than for the other. This process is repeated on a fixed number of bootstrap samples of the data set. Subsequently, the identified tuples from all bootstrap repetitions are collected as candidates for tuples featuring high-order interactions. For each identified tuple the frequency with which it occurs in the bootstrap samples is calculated in order to measure the stabilities of the corresponding high-order interactions. Basu *et al.* (2018) denote these frequencies as stability scores. Jiang *et al.* (2009) present an approach called epiForest that, similarly to SNPInterForest, is exclusively tailored to SNP data. With epiForest, a small number of promising variables is pre-selected using a forward selection procedure

based on the Gini importance of a random forest. Next, all possible pairs and triples of variables are tested for two-way and three-way interaction effects and the resulting p-values are adjusted for multiple testing using the Bonferroni correction. The latter adjustment does not account for the variables considered in the testing already being pre-selected, which could lead to an increased number of false positive results.

Methods for metric outcomes have been proposed by Sorokina *et al.* (2008) and Du and Linero (2019). To measure the importance of an interaction between two variables, Sorokina *et al.* (2008) compare the predictive performance of so-called Additive Groves of two trees (Sorokina *et al.*, 2007) that are unrestricted to that of Additive Groves of two trees that are restricted in such away that they do not model interactions between the two respective variables. Additive Groves of trees consist of regression trees where the predictions are summed up to obtain the final predictions. In the case of the restricted Additive Groves of trees, the first tree is restricted to exclude one of the two variables and the second tree the other. Du and Linero (2019) take a Bayesian perspective in their method, Dirichlet process forests. These consist of clusters of trees, where trees in the same cluster focus on detecting a specific interaction.

B Details on the interaction forest algorithm

B.1 Prediction algorithm

While the split selection and splitting is performed differently with interaction forests than with conventional random forests, both algorithms result in large numbers of decision trees the leaf nodes of which feature observations with similar values of the outcome variable. For this reason, prediction using interaction forests is performed in the same way as in the case of conventional random forests and their variants. For categorical outcomes, either the point predictions of the trees are summarized using majority voting to obtain point predictions, or the outcome class probabilities predicted by the trees are averaged to obtained class probability predictions (option `probability=TRUE` in the function `interactionfor()` of the `diversityForest` package that allows to construct an interaction forest). For continuous and survival outcomes, the predictions of the individual trees are averaged; see Ishwaran *et al.* (2008) for details on the survival case.

B.2 Procedure used for pre-selecting variable pairs that show indications of interaction effects

In cases with larger numbers of variables, the number of possible variable pairs becomes very large. Consequently, when considering all possible variable pairs, the individual variable pairs would be considered too rarely for splitting in the forests, or not at all. This could not only have a negative effect on the predictive performance, because strongly interacting variable pairs would be considered too rarely or not at all, but it would also make the ranking of the interaction effects impossible. For this reason, if the number of possible variable effects is too large, we perform a pre-selection of promising variable pairs. This pre-selection is performed if the number of possible variables pairs $\binom{p}{2}$ is larger than 5000, that is, for data sets with p larger than 100. For such data sets, we pre-select 5000 promising variable pairs. If the dimensionality of the data is not too large ($5000 < \binom{p}{2} < 10^5$), we can simply test each possible variable pair for interaction effect in the pre-selection. In cases of larger numbers of variables, we integrate the recently introduced interaction effect screening procedure BOLT-SSI (Zhou *et al.*, 2019) into the pre-selection. For data sets with more than 30000 variables, we pre-select 30000 variables in a univariate fashion before applying BOLT-SSI. Below we describe the pre-selection procedure in detail.

If the number of possible variable pairs $\binom{p}{2}$ is smaller than 10^5 (i.e., for $100 < p < 448$) we test each possible variable pair for interaction effect and select the variables pairs associated with the smallest p-values out of these tests. Independent of whether the outcome is binary or metric, for each variable pair (x_{j_1}, x_{j_2}) we perform a linear regression of the outcome on x_{j_1}, x_{j_2}, and $x_{j_1} \cdot x_{j_2}$. The latter product models the interaction between the two variables. Subsequently, for each variable pair, we record the p-value of the test of the coefficient of the interaction term $x_{j_1} \cdot x_{j_2}$ being equal to zero. Finally, we pre-select the 5000 variable pairs with the smallest p-values of the interaction effects. Performing such a large number of linear regressions becomes computationally possible through the `fastLmPure()` function from the R package `RcppEigen`. If $n > 500$, we select randomly a subset of 500 observations to speed up the calculations. In the latter random selection, we include all observations from the smaller class, if it is represented by less than or equal to 30 observations in the data set and otherwise condition the smaller class

to be represented by at least 30 sampled observations. The latter was performed to avoid the smaller class to be represented by too few observations in the subset data set. For multiclass responses with more than two classes, we only consider observations from the two largest classes and subsequently treat the target variable as a binary variable performing the same pre-selection algorithm described above. For survival outcomes we only consider observations with observed survival times and again perform linear regressions in the same way as described above.

If the number of possible variable pairs is larger than 10^5, that is, if p is larger than 447, testing each possible variable pair soon becomes too costly from a computational point of view. A simple procedure to obtain promising variable pairs would be to randomly sample 10^5 variable pairs, test each variable pair for interaction effect and keep the variable pairs with the smallest p-values. However, with this procedure it is likely that we would miss some of the most important interaction effects, if p is large. For this reason we employ the recently introduced interaction effect screening procedure BOLT-SSI (Zhou *et al.*, 2019), which aims at finding the most important two-way interaction effects in high-dimensional data. Using BOLT-SSI, a maximum of p variable pairs can be selected. Therefore, if $p < 5000$, the number of variable pairs found using BOLT-SSI will be less than the number 5000 of variable pairs we want to select. To obtain 5000 selected variable pairs for $p < 5000$, first, if $n > 200$, we select randomly a subset of 200 observations in the same way as in the case of $100 < p < 448$ described above for computational efficiency and, second, conduct the following procedure: 1) Apply BOLT-SSI to obtain the first p selected variable pairs. The variable pairs that interact the strongest will be likely among these p variable pairs; 2) For $l = 1, \ldots, 20$: a) Apply BOLT-SSI to the data, using a subset of $\lfloor p/3 \rfloor$ variables sampled randomly anew in each iteration; b) Add those variable pairs selected in a) that are not among the already selected variable pairs; c) Stop if the total number of selected variable pairs exceeds 5000, in which case only the first 5000 selected variable pairs are kept. Apart from increasing the number of selected variables, the random subsetting of the variable space in the latter procedure also avoids that the selected variable pairs are dominated too strongly by a few variables which interact with many other variables. If the number of selected variable pairs n_{found} is still smaller than 5000 after applying this procedure, we proceed as follows: 1) Draw randomly $20 \cdot (5000 - n_{found})$ from all possible variable pairs not among the already selected variable pairs; 2) Test each of the variable pairs drawn in 1) for interaction effect (using `fastLmPure()` as described in the previous paragraph) and add the $5000 - n_{found}$ variable pairs associated with the smallest p-values from these tests to the set of selected variable pairs.

If $5000 \leq p \leq 30000$, we simply select p variable pairs using BOLT-SSI, again using a random subset of 200 observations if $n > 200$. For data sets with $p > 30000$, applying BOLT-SSI to all variables would be computationally too challenging. Therefore, if $p > 30000$, we pre-select 30000 variables by regressing the outcome on each variable using univariable linear regressions and keep the 30000 variables with the smallest p-values from the tests of the slopes being equal to zero. In these linear regressions, categorical and survival outcomes are handled in the same way as in the case of the linear regressions performed for pre-selecting the variable pairs (see above). If $n > 500$, these univariable linear regressions are performed using a random subset of 500 observations. Subsequently, we apply BOLT-SSI to select 5000 promising variable pairs, where this selection is based on a random subset of 200 observations if $n > 200$.

Note that the above procedure for pre-selecting the promising variable pairs is of ad hoc

nature. It cannot be excluded that some relevant interaction effects are missed with this procedure. However, given the fact that the number 5000 of variable pairs to pre-select is large, most of the variable pairs with reasonably strong interactions are likely included in the set of pre-selected variable pairs.

B.3 Handling of unordered categorical covariate variables

The split types presented in Section 3.2.4 of the main paper do not apply directly to unordered categorical variables. This issue is dealt with in the interaction forest algorithm by converting unordered categorical variables into ordered variables. The ordering is performed in the same way as when using the option `respect.unordered.factors="order"` implemented in the R package `ranger` (version 0.12.1) (Wright and Ziegler, 2017). The idea of this option is to take the outcome into account for ordering the categories in such a way that when moving along the ordering of the categories, the outcome tends to change in a consistent direction. For example, in the case of two-class classification the categories are ordered according to the proportion of the corresponding observations falling into the second class. For multi-class classification, the categories are ordered according to the first principal component of the weighted covariance matrix. For further details, see Wright and König (2019) who describe these approaches in detail and evaluate them empirically.

B.4 Procedure used for drawing $p_b^{(j_2)}$

When drawing the $p_b^{(j_2)}$ values, that is, the second points in the split point pairs for the bivariable splits, it has to be made sure that all resulting five bivariate splits are valid. The latter is the case, if each quadrant in the two-dimensional coordinate system with origin $(p_b^{(j_1)}, p_b^{(j_2)})$ contains at least one observation.

Based on the latter notion, we use the following procedure for drawing $p_b^{(j_2)}$:

1. Be $x_{j_2|x_{j_1}<p_b^{(j_1)}}$ and $x_{j_2|x_{j_1}>p_b^{(j_1)}}$ the subsets of the x_{j_2} values for which the corresponding observations have x_{j_1} values smaller and larger than $p_b^{(j_1)}$, respectively. For $p_b^{(j_1)}$, see step 1.(c)i. of the algorithm for split selection shown in Section 3.2.5 of the main paper.

   Moreover, define the following: $a_l := \max(\min(x_{j_2|x_{j_1}<p_b^{(j_1)}}), \min(x_{j_2|x_{j_1}>p_b^{(j_1)}}))$ and $a_u := \min(\max(x_{j_2|x_{j_1}<p_b^{(j_1)}}), \max(x_{j_2|x_{j_1}>p_b^{(j_1)}}))$. If $a_l \geq a_u$, draw a new $p_b^{(j_1)}$ value (by going back to step 1.(c)i. of the algorithm for split selection). In the current implementation, a maximum of 20 new $p_b^{(j_1)}$ values are drawn. If a_l is still greater or equal to a_u by then, we only use the univariable splits for the drawn variable pair x_{j_1} and x_{j_2} from step 1.(b) of the algorithm for split selection and continue by drawing a new variable pair. However, in practice this case can be expected to be very rare.

2. Draw $p_b^{(j_2)}$ by taking the average of two randomly drawn values from the subset of the unique x_{j_2} values contained in the interval $[a_l, a_u]$.

B.5 Hyperparameter values used by default

For the number of variable pairs to sample for each split *npairs* we use $\min\{\sqrt{p}/2, 10\}$. The quantity $\sqrt{p}/2$ follows the common rule of thumb of random forests to sample $\sqrt{p}$ candidate variables for each split. We divide $\sqrt{p}$ by two, because we sample pairs of variables. The reason for limiting *npairs* by ten is computational efficiency. In the large-scale empirical study performed in Hornung (2020) the performance of diversity forests did not generally improve beyond using only a few candidate splits. The latter result was independent of the number of variables in the data sets, where, however, no data sets with more than 240 variables had been included. Nevertheless, it seems unlikely that for very high-dimensional data sets a larger number of candidate splits would be beneficial for the following two reasons: First, there was no relevant improvement for larger numbers of candidate splits in the case of any of the data sets studied in Hornung (2020). Second, even for very high-dimensional data sets, a maximum of 5000 pre-selected variable pairs is considered (cf. Section B.2). The latter restriction has the effect that the number of sampled candidate splits does not vanish in relation to the number of possible splits for very high-dimensional data sets.

The number of trees to construct for each forest is set to 20000 if EIM values should be calculated and to 2000 otherwise. The reason for using such a large number of trees is that this ensures that stable rankings of the important interaction effects are obtained, given that the number of all possible variable pairs can be quite large and that several lists of EIM values must be obtained, see Section 3.3 of the main paper. Tree construction is performed fully in C++ in `diversityForest` and using all cores available on the system in parallel by default, which is why so many trees does not pose an issue computationally.

Subsampling of the subsets used for tree construction is performed without replacement, where sample fractions of 0.7 are used since Probst *et al.* (2019) found these defaults to be optimal on average in the case of random forests in a large real data study.

The diversity forest algorithm presented in Hornung (2020) involves a parameter *proptry*, which, if set to values smaller than one, limits the number of candidate splits tried in the case of small nodes to a fixed proportion *proptry* of all possible splits, in order to avoid risking overfitting in small nodes. It was seen in Hornung (2020) that this parameter does not influence the performance strongly and values of one were often appropriate. Therefore, in the current version of the interaction forest algorithm, we did not include this parameter. It is implicitly set to the value one, because the number of sampled candidate splits is not restricted for small nodes. If this parameter would be included, this would have the effect that for small nodes, smaller numbers than *npairs* variable pairs would be sampled in the split selection.

B.6 Procedure for adjusting the raw quantitative EIM values to make them specific for quantitative interaction effects

As described in Section 3.3 of the main paper, two variables that both have strong univariable effects will have large raw quantitative EIM values for two of the four quantitative split types, where the corners of these two split types are opposing in Figure 1 of the main paper. Therefore, using the raw quantitative EIM values in the case of the quantitative interaction effects would make it impossible to discern between variable pairs with quantitative interaction effects and variable

pairs, for which both members have univariable effects only.

In the case of quantitative interaction effects, however, the raw quantitative EIM value will be large for only one of the four quantitative split types. Based on the latter notion we apply the following procedure in an effort to make the raw quantitative EIM values for each pair distinctive for quantitative interaction effects: We adjust each raw quantitative EIM value by subtracting the maximum of zero and the corresponding raw quantitative EIM value of that quantitative split type the corner of which (Figure 1 of the main paper) opposes that of the targeted quantitative split type. These adjusted raw quantitative EIM values will be small for pairs of variables that both have a univariable effect, but no quantitative interaction effect. Taking the maximum of zero and the respective raw quantitative EIM values before subtracting the latter is equivalent to subtracting them only if they are non-negative. This choice was made to prevent meaningless results in cases in which there are no important quantitative interaction effects in the data: In such situations, the negative quantitative EIM values will be of similar sizes in absolute terms as the positive quantitative EIM values. Here, subtracting negative quantitative EIM values can influence adjusted quantitative EIM values to become large in comparison to the others by chance, if the subtracted negative quantitative EIM values are large in absolute terms.

C Real data based exemplary interaction forest analyses

In the following several real data analyses, we will illustrate how interaction forests can be applied for interaction detection in practice. Note that we are not closely familiar with the subject matters studied with the investigated data sets. Therefore, we are not able to judge how meaningful our interpretations of the shown relations are on a context level. These analyses are meant for illustrative purposes only.

C.1 'stock' data – continuous outcome

This data set contains 950 daily stock prices from January 1988 through October 1991, for ten aerospace companies. The data were obtained from the open science online platform, OpenML (Vanschoren *et al.*, 2013), by downloading it under the data set ID 223 using the R package of the same name (version 1.10) (Casalicchio *et al.*, 2017). The names of the companies were anonymized and the stock prices for one of these companies ("`company10`") were flagged as the outcome. Thus, for this data set, both the outcome and the covariate variables were continuous.

As a first step, we construct an interaction forest and calculate the univariable, the quantitative, and the qualitative EIM values:

```
library("diversityForest")
set.seed(1234)
model <- interactionfor(dependent.variable.name = "company10", data = datastock,
                        importance="both")
```

The argument `importance="both"` (default) specifies that both quantitative and qualitative EIM values should be calculated. As there are only nine variables, we can have a look at the univariable EIM values printed to the console. The object `eim.univ.sorted` contained in the object `model` produced by the above code contains the univariable EIM values sorted in descending order:

```
model$eim.univ.sorted
```

```
  company1   company6   company9   company8   company2   company7   company5
11.8405795  5.1407022  3.9815370  2.7472936  2.4069568  1.3922671  1.3836012
company3   company4
1.3246459  0.9373223
```

Among all nine companies, the stock prices of `company1` seem to be associated the strongest with those of `company10`.

The output of `interactionfor()` is an object of class `interactionfor`. There is a `plot()` function for `interactionfor` objects. We apply this function as follows:

```
plot(model)
```

By default, this function produces three plots, which are shown in Supplementary Figures S1 to S3. When executing `plot(model)`, the first of these plots will be shown and the remaining plots are shown by pressing ENTER repeatedly. The first plot shows the univariable, quantitative, and qualitative EIM values in decreasing order. Looking at the distribution of the univariable EIM values helps judge if some variables are particularly important. These would have much larger

univariable EIM values than all other variables. If the largest quantitative or qualitative EIM values set themselves apart strongly for all remaining values, it is likely that the corresponding variable pairs feature particularly strong quantitative or qualitative interaction effects. It is, nevertheless, still important to visualise the bivariable influences of these variables to prevent false positive results and to learn about the forms of the interaction effects.

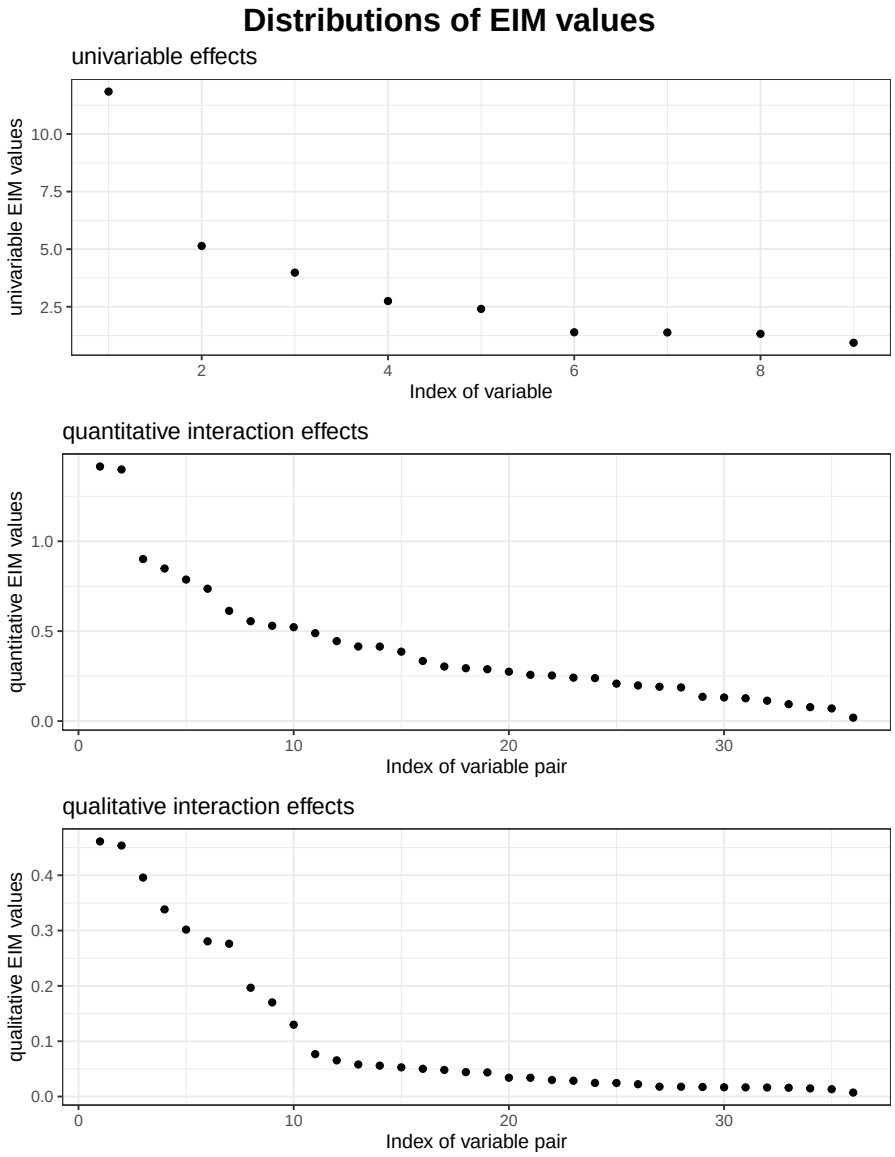

Fig. S1: Result of function `plot.interactionfor()`: EIM values ('stock' data set). The values are sorted in decreasing order.

Supplementary Figure S2 shows the estimated bivariable influences of the two variable pairs with the largest quantitative EIM values. The sub-captions of the upper and lower panels provide the information on which types the quantitative interaction effects were classified to by the interaction forest algorithm (cf. Section 3.3 of the main paper). For example, the sub-caption of

the upper panels reads "company6 small AND company9 large". This means that if the stock prices are low for `company6` and at the same time high for `company9`, this will have an effect on the stock prices of `company10` (i.e., the outcome). However, it is not clear yet from this caption "company6 small AND company9 large", *how* this affects the stock price of `company10`. The latter is revealed by the subplots in the upper panels: The LOESS fits deliver considerably larger values in the upper left corner of the plot and the data points are considerably darker in this region, where darker values correspond to larger outcome values. The subplot in the upper-right panel confirms that the LOESS fits are largest if the values of `company6` are small and at the same time those of `company9` are large. Therefore, we can conclude that the stock prices of `company10` are particularly high if those of `company6` are low and at the same time those of `company9` are high. Analogous interpretations can be made for the lower panels.

The plots also contain the results of tests on interaction effects obtained using classical linear regression. If the p-values of these tests are large, this can indicate that the variable pair does not feature a true interaction effect. However, it is also possible that the interaction effect is not detected using classical regression. For example, consider a binary variable A, a continuous variable B, and a binary outcome: Suppose that for $A = 0$ all observations are of outcome class 1 and for $A = 1$ the observations are of outcome class 1 if $B < 5$ and of outcome class 2 if $B \geq 5$. This would correspond to a clear quantitative interaction effect associated with split type 6 in Figure 1 of the main paper. However, a test for interaction effect using logistic regression would not deliver a significant result in this situation. This artificial example illustrates that classical parameter approaches are not always successful in interaction detection.

Supplementary Figure S3 shows the estimated bivariable influences of the two variable pairs with the largest qualitative EIM values. The upper panels of this plot suggests that if the stock price of `company2` is low, `company7` seems to have a negative influence on the stock price of `company10`, but if `company2` has a high stock price, `company7` seems to have a positive influence on the stock price of `company10`. The lower panels suggests that the qualitative interaction effect between `company1` and `company7` is quite different than that between `company2` and `company7`: The stock price of `company7` seems to have a positive influence on the stock price of `company10` if the stock price of `company1` is low, but a negative influence if the stock price of `company1` is high.

By default, the `plot()` function for `interactionfor` objects produces plots for the *two* variable pairs with the largest quantitative and qualitative interaction forests, but using the function arguments `numpairsquant` and `numpairsqual` different numbers of top variable pairs can be shown.

In our analysis of Supplementary Figure S3 we saw that both variable pairs with largest qualitative EIM values involved `company7`. Looking at the ordered list of qualitative EIM values, `model$eim.qual.sorted` reveals that four of the five variable pairs with largest qualitative EIM values involve `company7`. Suppose we want to visualise all bivariable influence of variable pairs that involve `company7` in descending order of the qualitative EIM values. This can be realised with the function `plotEffects()` using the argument `allwith`:

```
plotEffects(model, allwith="company7", type="qual", numpairs=8)
```

Here, `type="qual"` specifies that the variables pairs should be sorted according to the qualitative EIM values in descending order. Moreover, `numpairs=8` specifies that we want to visualise the first eight of these pairs which corresponds to all possible pairs that involve `company7`, since there

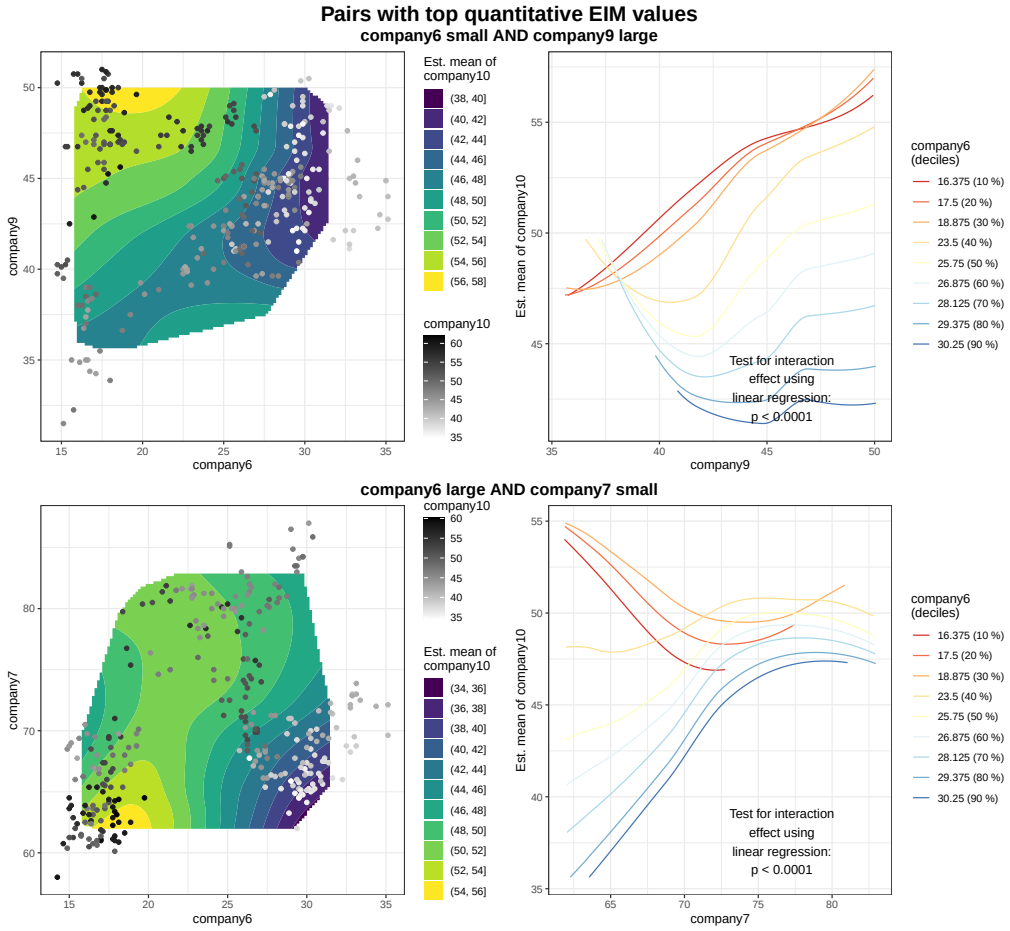

Fig. S2: Result of function `plot.interactionfor()`: Estimated bivariable influences of the two variable pairs with the largest quantitative EIM values ('stock' data set). The contour plots in the left panels show two-dimensional LOESS fits. For reasons of clarity, the points in the left panels do not show all observations, but random subsets of 300 observations. The lines in the right panels show cross sections of the two-dimensional LOESS fits in the left panels.

are nine companies as covariate variables. Per default, only plots for the first `numpairs=5` pairs are shown. The argument `allwith` is particularly useful in situations in which a specific variable (e.g., a treatment variable in medical studies) is of main interest.

The first plot produced by the above command is identical to Supplementary Figure S3 (as the two variable pairs with largest qualitative EIM value both contained `company7`). The remaining three plots are shown in Supplementary Figures S4 to S6. Only the first of these figures, Supplementary Figure S4, suggests strong qualitative interaction effects. Thus, for the four variable pairs involving `company7` that featured the largest qualitative EIM values, we observed strong qualitative interaction effects, but not for the others. This is in line with the qualitative EIM values obtained for the variable pairs that involve `company7`: The largest four of these are much larger than the remaining four, suggesting that only the largest four are associated with relevant qualitative interaction effects.

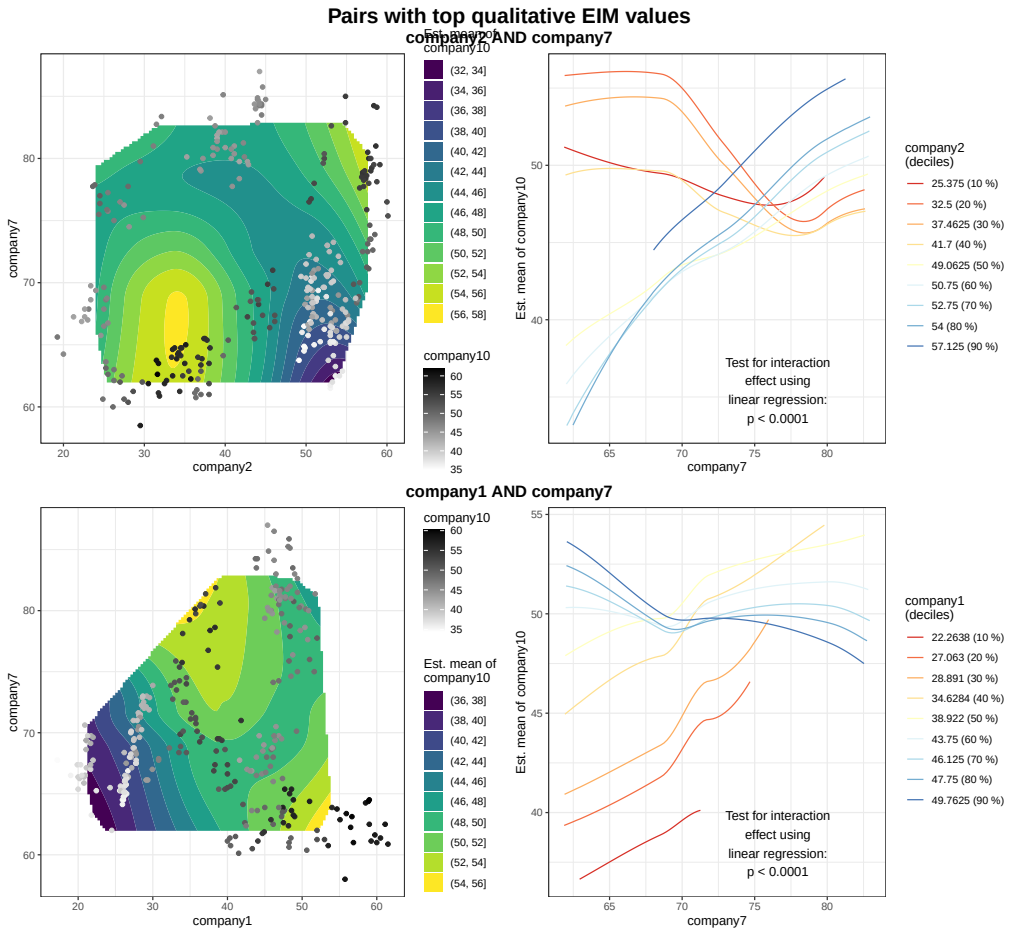

Fig. S3: Result of function `plot.interactionfor()`: Estimated bivariable influences of the two variable pairs with the largest qualitative EIM values ('stock' data set). The contour plots in the left panels show two-dimensional LOESS fits. For reasons of clarity, the points in the left panels do not show all observations, but random subsets of 300 observations. The lines in the right panels show cross sections of the two-dimensional LOESS fits in the left panels.

The function `plotEffects()` can also be used for different purposes than the one shown above. We will use this function again in the analyses of further data sets in the next sections.

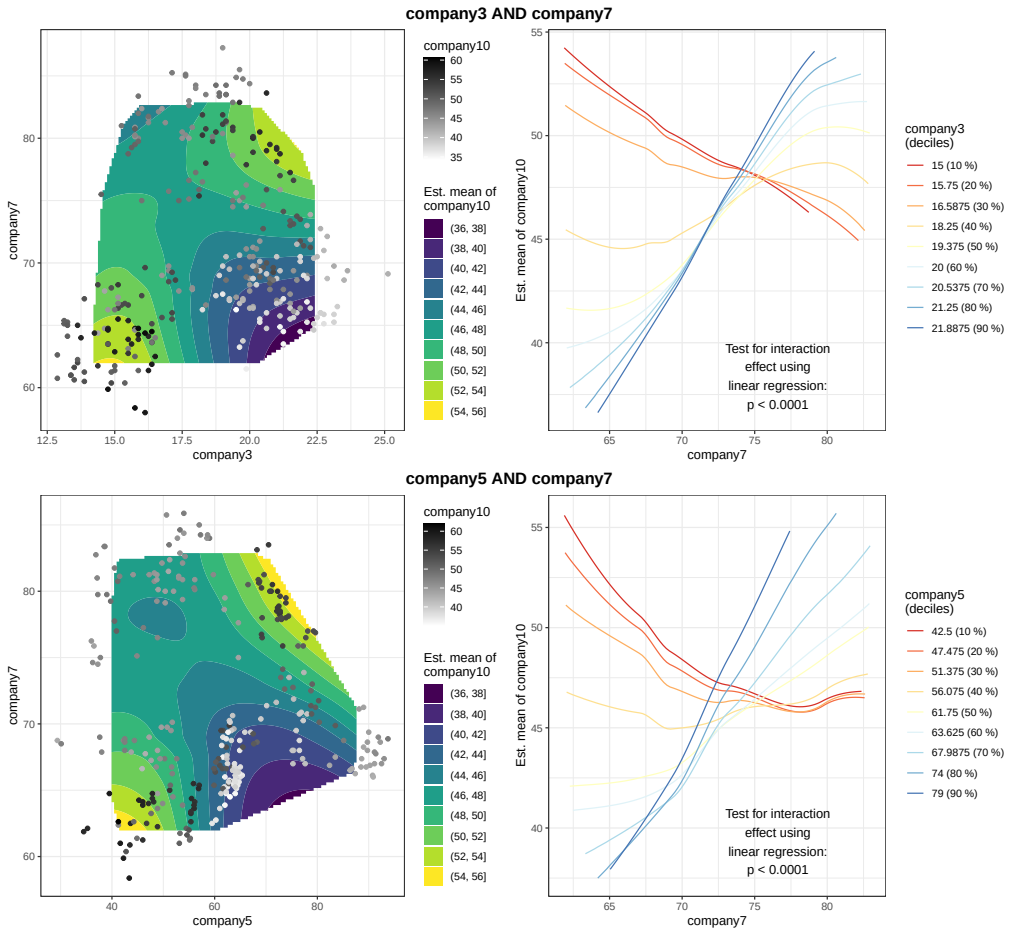

Fig. S4: Result of function `plotEffects()`: Estimated bivariable influences of the two variable pairs involving `company7` with the third and fourth largest qualitative EIM values ('stock' data set). The contour plots in the left panels show two-dimensional LOESS fits. For reasons of clarity, the points in the left panels do not show all observations, but random subsets of 300 observations. The lines in the right panels show cross sections of the two-dimensional LOESS fits in the left panels.

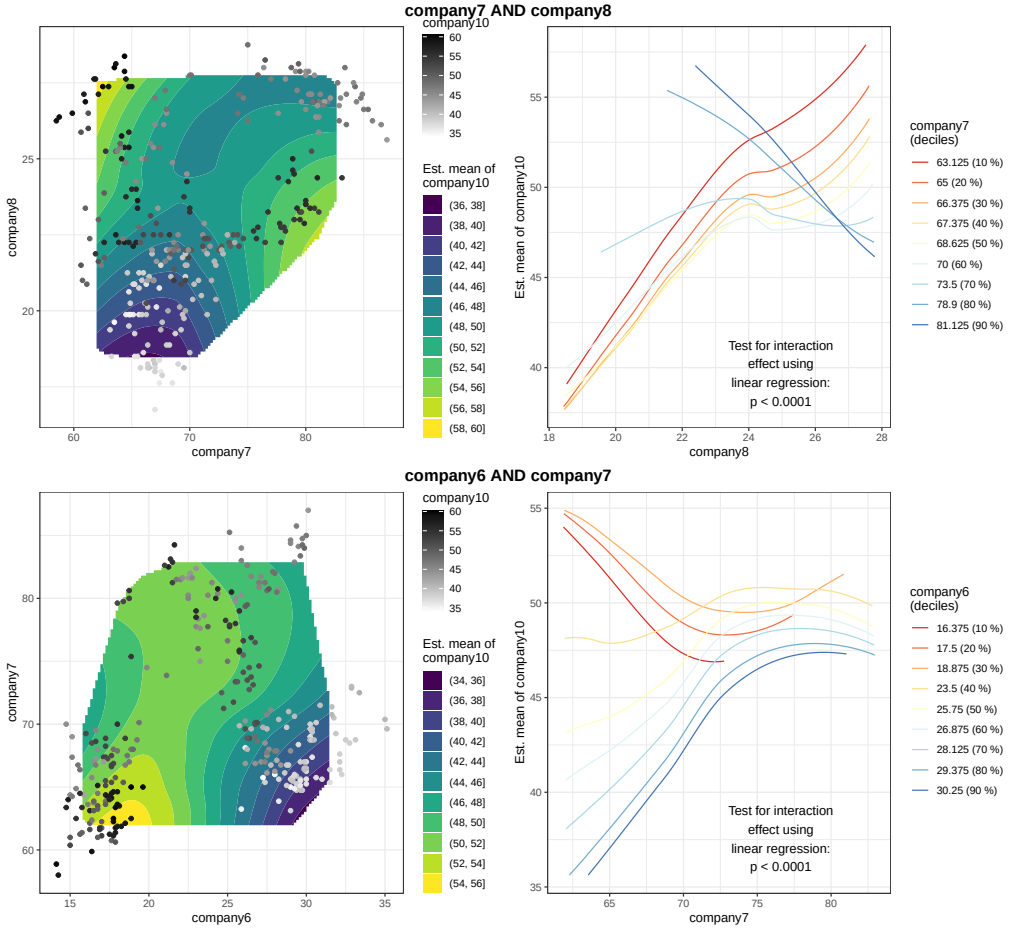

Fig. S5: Result of function `plotEffects()`: Estimated bivariable influences of the two variable pairs involving `company7` with the fifth and sixth largest qualitative EIM values ('stock' data set). The contour plots in the left panels show two-dimensional LOESS fits. For reasons of clarity, the points in the left panels do not show all observations, but random subsets of 300 observations. The lines in the right panels show cross sections of the two-dimensional LOESS fits in the left panels.

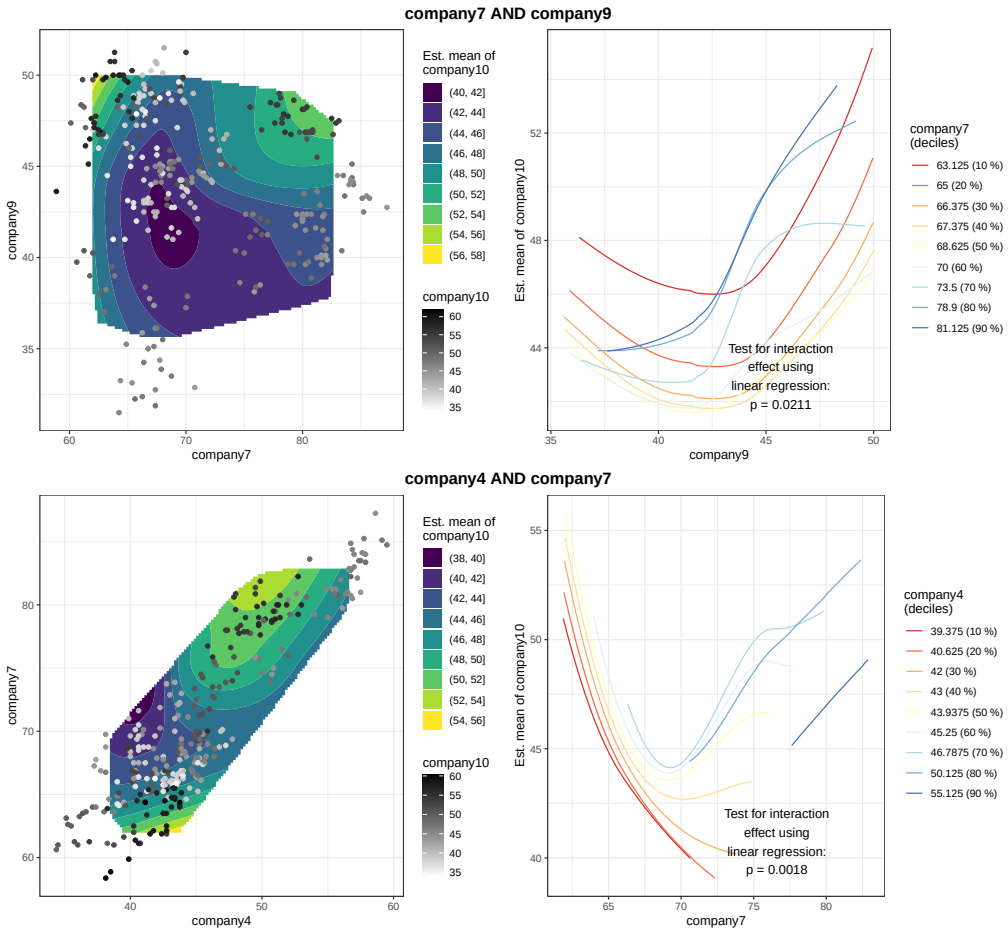

Fig. S6: Result of function `plotEffects()`: Estimated bivariable influences of the two variable pairs involving `company7` with the seventh and eighth largest qualitative EIM values ('stock' data set). The contour plots in the left panels show two-dimensional LOESS fits. For reasons of clarity, the points in the left panels do not show all observations, but random subsets of 300 observations. The lines in the right panels show cross sections of the two-dimensional LOESS fits in the left panels.

C.2 'zoo' data – binary covariate variables

This data set describes 101 different biological species using 16 simple attributes, where 15 of these are binary and one is metric (the number of legs). The outcome "mammal vs. other" is binary. The data set was also downloaded from the open science online platform OpenML (data set ID: 965).

As a first step we again construct an interaction forest and calculate the EIM values:

```
set.seed(1234)
model <- interactionfor(dependent.variable.name = "type", data = datazoo)
```

To get a first overview we apply the function `plot()` to `model`:

```
plot(model)
```

The first plot produced by this command is shown in Supplementary Figure S7. The univariable EIM values in this plot suggest that one of the variables has a particularly strong influence. We can identify this variable by consulting the univariable EIM values sorted in decreasing order:

```
model$eim.univ.sorted[1]
```

```
     milk
0.1748339
```

This variable indicates whether the species produces milk or not. Using this variable alone, it would be possible to classify all species correctly, because all 41 mammals in the data set do give milk and all 60 other species do not give milk.

The estimated bivariable influences of the two variable pairs with largest quantitative EIM values are shown in Supplementary Figure S8. Both variable pairs seem to feature a strong quantitative interaction effect associated with split type two in Figure 1 of the main paper. The ordering of the two categories `true` and `false` in the plots corresponds to the ordering found using the procedure described in Section B.3. The sub-captions of the upper and lower panels that describe which types the quantitative interaction effects are of, also refer to this ordering.

The upper panels of Supplementary Figure S8 reveals that all species in the data set that had both hair and a backbone, were mammals. Only two of the 41 mammals in the data set did not have both, and these were dolphins and porpoises who do not have hair. Therefore, if we would classify the species in the data set only by whether they have both hair and a backbone, we would classify 99 of 101 species correctly. Only four of the species that had hair were not mammals (these were: honey bee, housefly, moth, wasp).

All species in the data set that do not lay eggs and are not venomous are mammals (lower panels of Supplementary Figure S8). Except for one mammal, all mammals in the data set do not lay eggs and are not venomous. The exception was the platypus, which is not venomous, but does lay eggs. Thus, if we would classify the species based on whether they do not lay eggs and are not venomous at the same time, we would classify 100 of the 101 species correctly.

The estimated bivariable influences of the two pairs with the largest qualitative EIM values are shown in Supplementary Figure S9. The observed interaction effect between the number of legs and the presence or absence of a tail (upper panels of Supplementary Figure S9) is mostly

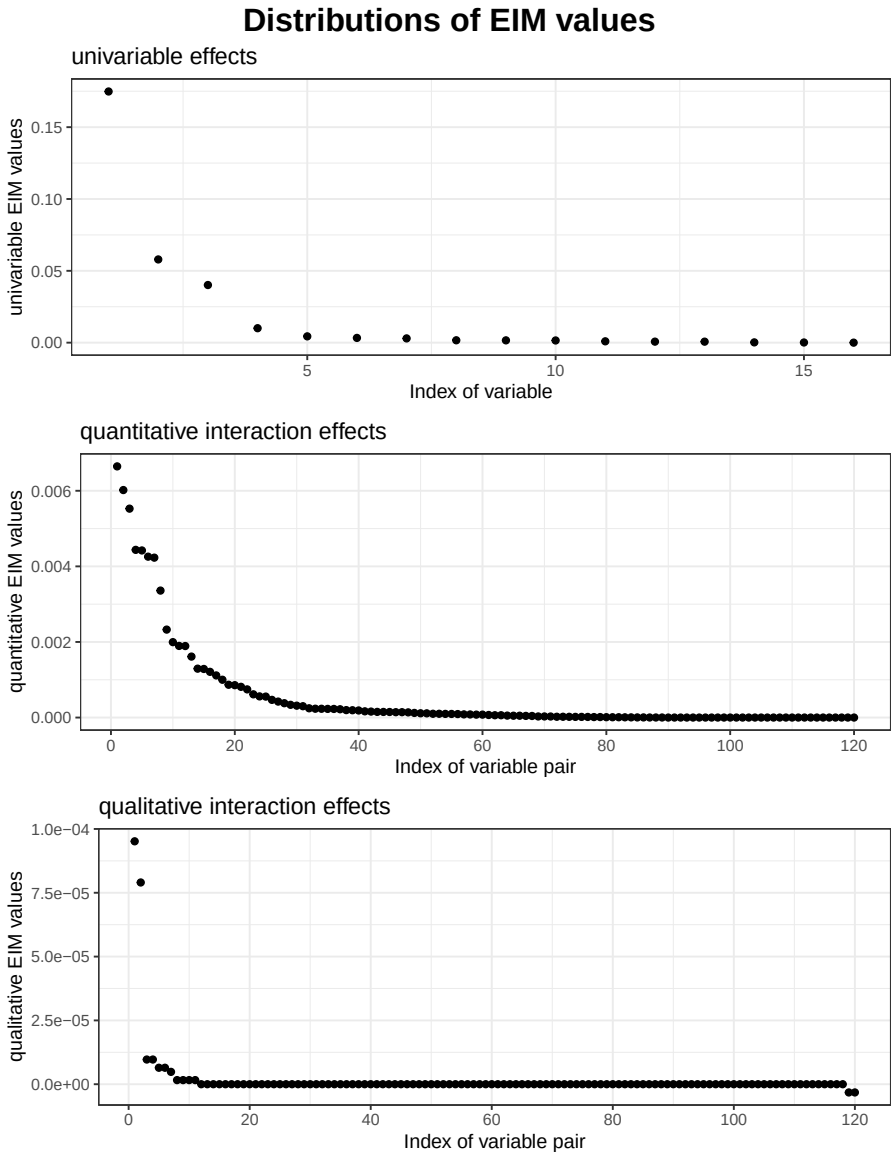

Fig. S7: Result of function `plot.interactionfor()`: EIM values ('zoo' data set). The values are sorted in decreasing order.

due to species with two and four legs in this data set: While species that have a tail are much more frequently mammals if they have four legs compared to if they have two legs, species that do not have a tail are much more likely to be mammals if they have two legs compared to if they have four legs. This can be explained as follows: 1) species that have a tail and two legs are most often birds; 2) species that have a tail and four legs are most often mammals; 3) there were only two species without a tail and two legs, which were both mammals (human and gorilla); 4) about half of the species without a tail were mammals. The observed qualitative interaction effect between the presence or absence of a tail and the presence or absence of fins (lower panels of Supplementary Figure S9) is due solely to one species which is a seal. The latter is registered

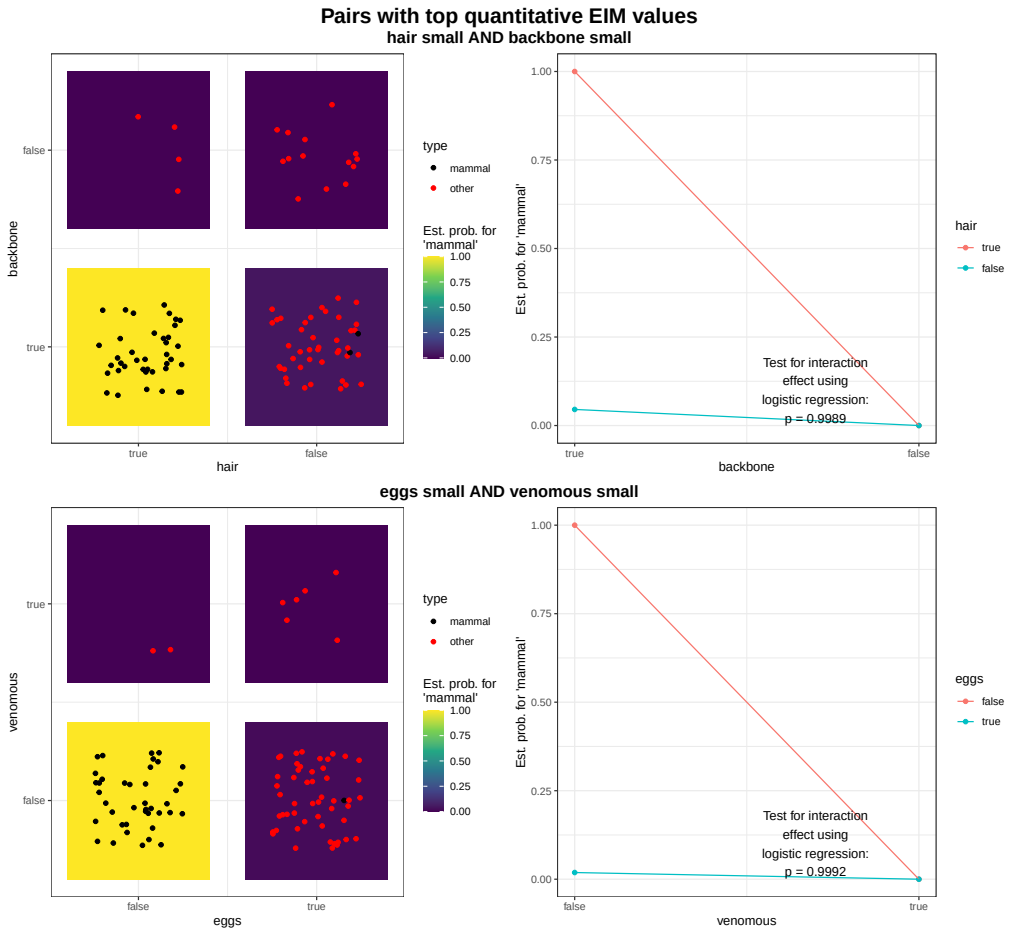

Fig. S8: Result of function `plot.interactionfor()`: Estimated bivariable influences of the two variable pairs with the largest quantitative EIM values ('zoo' data set). The heat maps in the left panels and the dots in the right panels show the frequencies of mammals in the respective combinations of the categories.

in the data as having fins, but no tail (the latter specification may not be correct, because sea lions are by contrast specified as having a tail in this data set). The remaining results are clear here: Species with fins are rarely mammals and species without fins are more often mammals if they have a tail than if they do not.

In Supplementary Figure S7, the two largest qualitative EIM values set themselves apart very strongly from the remaining qualitative EIM values. This suggests that, apart from the two variable pairs with largest qualitative EIM values (Supplementary Figure S9), none of the remaining variable pairs show indications of qualitative interaction effects. We check this presumption by plotting the estimated bivariable influences of the variable pairs with the largest five qualitative EIM values using the following command:

```
plotEffects(model, type="qual")
```

For the plots obtained for the two variable pairs with the largest qualitative EIM values see

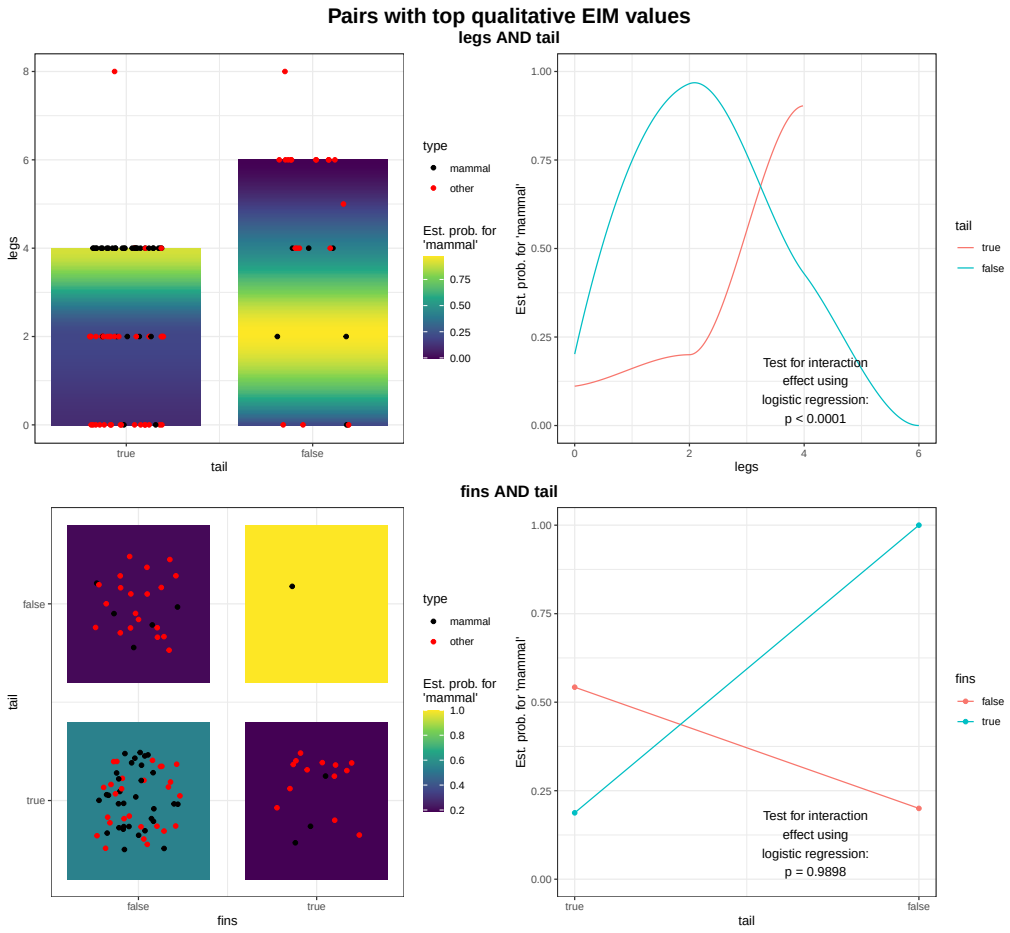

Fig. S9: Result of function `plot.interactionfor()`: Estimated bivariable influences of the two variable pairs with the largest qualitative EIM values ('zoo' data set). The heat maps in the upper left panel and the lines in the upper right panel show (one-dimensional) LOESS fits. The outcome was coded as '1' for 'mammal' and '0' for 'other' when performing the LOESS regression. The heat maps in the lower-left panels and the dots in the lower-right panels show the frequencies of mammals in the respective combinations of the categories.

Supplementary Figure S9. The plots obtained for the variables pairs with third to fifth largest qualitative EIM values are shown in Supplementary Figures S10 and S11. The latter two plots do not suggest qualitative interaction effects for the respective variable pairs, which confirms our presumption that only the two variable pairs with largest qualitative EIM values show indications of qualitative interaction effects.

As a last step of the analysis, we want to study the bivariable influence of the presence or absence of teeth and the presence or absence of feathers. Plotting the estimated bivariable influence of a specific pair of choice can be performed using the function `plotPair()` (also possible with the function `plotEffects()` using the argument `pairs`):

```
plotPair(pair=c("toothed", "feathers"), yvarname="type", data=datazoo)
```

Note that this function does not require an `interactionfor` object and therefore can be used

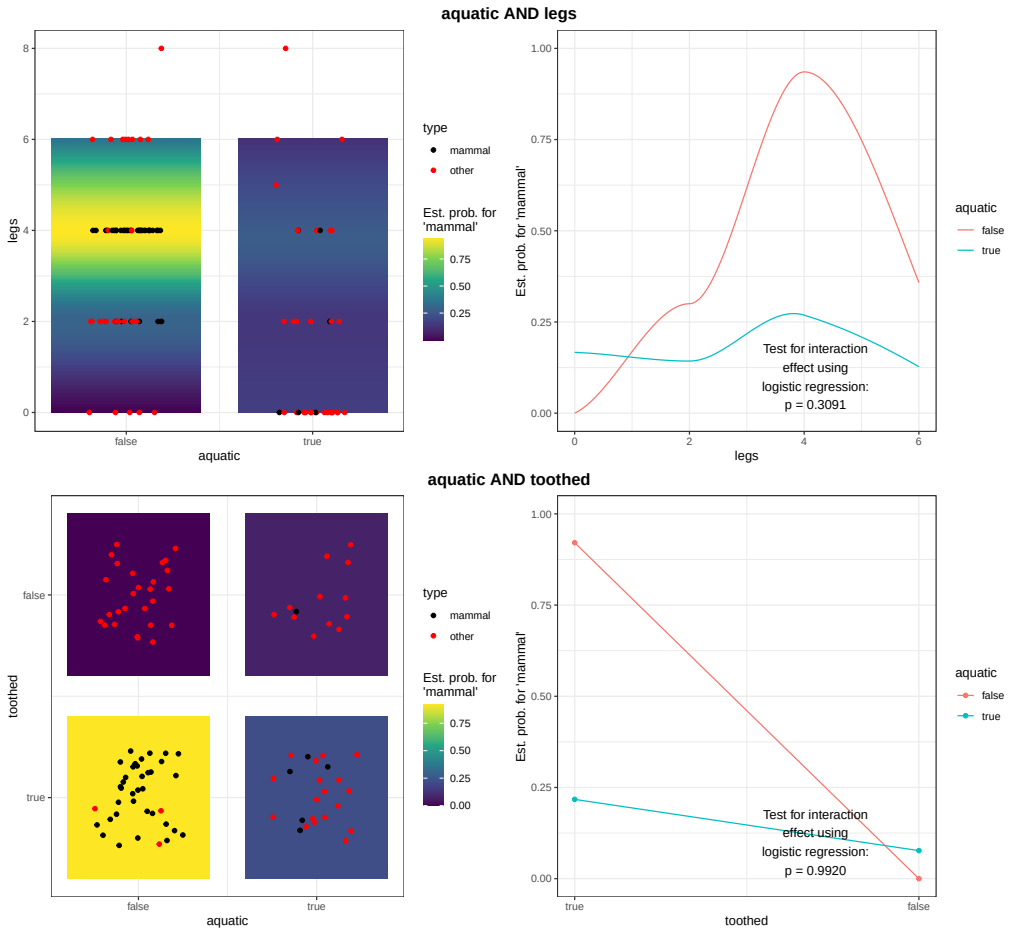

Fig. S10: Result of function `plotEffects()`: Estimated bivariable influences of the two variable pairs with the third and fourth largest qualitative EIM values ('zoo' data set). The heat map in the upper left panel and the lines in the upper right panel show (one-dimensional) LOESS fits. The outcome was coded as '1' for 'mammal' and '0' for 'other' when performing the LOESS regression. The heat map in the lower-left panel and the dots in the lower-right panel show the frequencies of mammals in the respective combinations of the categories.

without constructing an interaction forest beforehand. The resulting plot is shown in Supplementary Figure S12. There are no species in the data set that both have teeth and feathers. None of the species in the data set that have feathers is a mammal (these are all birds). Only one of the species that has neither feathers nor teeth is a mammal, which is the platypus. Most of the teethed species that do not have feathers are mammals. The other teethed species without feathers are mostly fish.

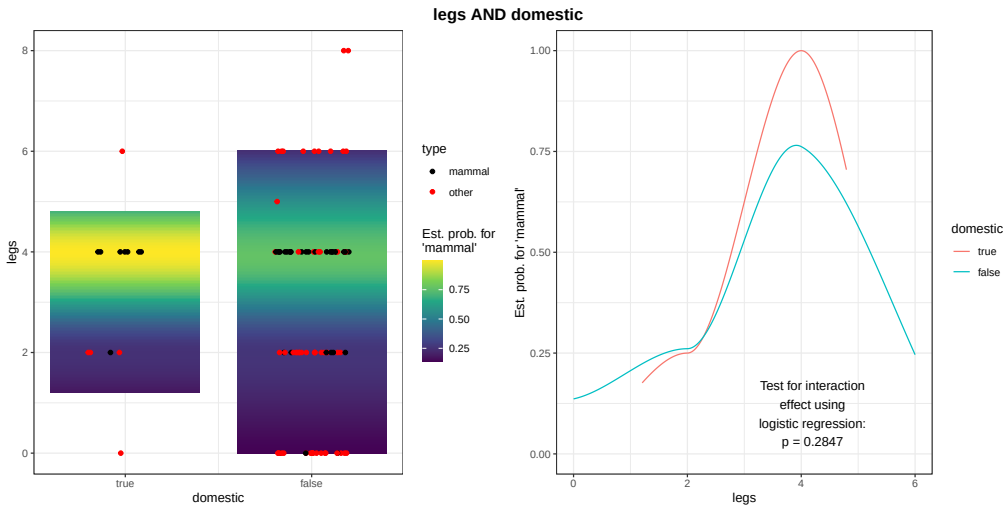

Fig. S11: Result of function `plotEffects()`: Estimated bivariable influences of the variable pair with the fifth largest quantitative EIM values ('zoo' data set). The heat map in the left panel and the lines in the right panel show (one-dimensional) LOESS fits. The outcome was coded as '1' for 'mammal' and '0' for 'other' when performing the LOESS regression.

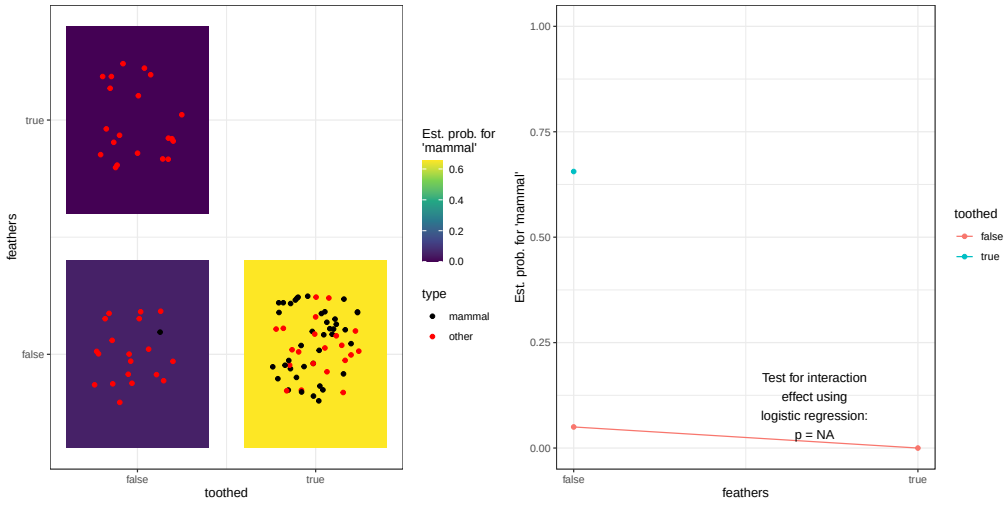

Fig. S12: Result of function `plotPair()`: Estimated bivariable influence of the variable pair `toothed` and `feathers` ('zoo' data set). The heat map in the left panel and the dots in the right panel show the frequencies of mammals in the respective combinations of the categories.

C.3 'white-clover' data – small sample size

The objective of these data was to predict the persistence of 63 white clover populations in summer dry hill land in 1994 using 31 variables that quantify the amount of white clover and other species in the years 1991 to 1994 and provide information on the strata where the white clover was being grown. These data were again downloaded from OpenML (data set ID: 1009). The binary outcome "[0, 8.82[vs. [8.82, 35.29]" quantifies the amount of white clover in the populations in 1994.

Note that there are 31 covariate variables in this data set, but only 63 observations. In the simulation study shown in Section 4.2 of the main paper, in the case of the smallest considered sample size $n = 100$, interaction forests were able to identify only strong qualitative interactions. Weaker qualitative interactions and quantitative interactions were not identifiable in this setting. The analysis presented in this section will, however, illustrate that also for small data sets it is possible to identify both qualitative and quantitative interaction effects using interaction forests.

As in the previous subsections, in the first step we construct the interaction forest and calculate the EIM values:

```
set.seed(1234)
model <- interactionfor(dependent.variable.name = "amount", data = dataclover)
```

Second, we obtain a first overview using the `plot()` function:

```
plot(model)
```

The resulting plots are shown in Supplementary Figures S13 to S15. None of the variables seem to have a particularly strong (univariable) influence (upper panel of Supplementary Figure S13). Analogous statements can be made with respect to the quantitative and qualitative interaction effects.

Both variable pairs with largest quantitative EIM values (Supplementary Figure S14) feature the variable `strata`, which is a variable with many unordered categories. The categories of `strata` are again ordered using the procedure described in Section B.3 in the figures. As seen in Supplementary Figure S14, observed quantitative and qualitative interaction effects determined with interaction forests that involve unordered categorical variables with many categories can be difficult to interpret. Nevertheless, someone with profound knowledge on the studied subject matter may be able to make meaningful interpretations here.

The estimated bivariable influences of both variable pairs with largest qualitative EIM values (Supplementary Figure S15) suggest qualitative interaction effects. However, given the small sample size, the observed relations should not be overinterpreted.

It is strongly advisable to investigate also variable pairs with smaller quantitative and qualitative EIM values for interaction effects beyond the two variable pairs with largest quantitative and qualitative EIM values that are considered by default in the `plot()` function. As we saw already in the previous subsections, the function `plotEffects()` can be used for this purpose. When applying `plotEffects()` without specifying any further function arguments than the `interactionfor` object, the bivariable influences of the variable pairs with the five largest quantitative EIM values are visualised:

```
plotEffects(model)
```

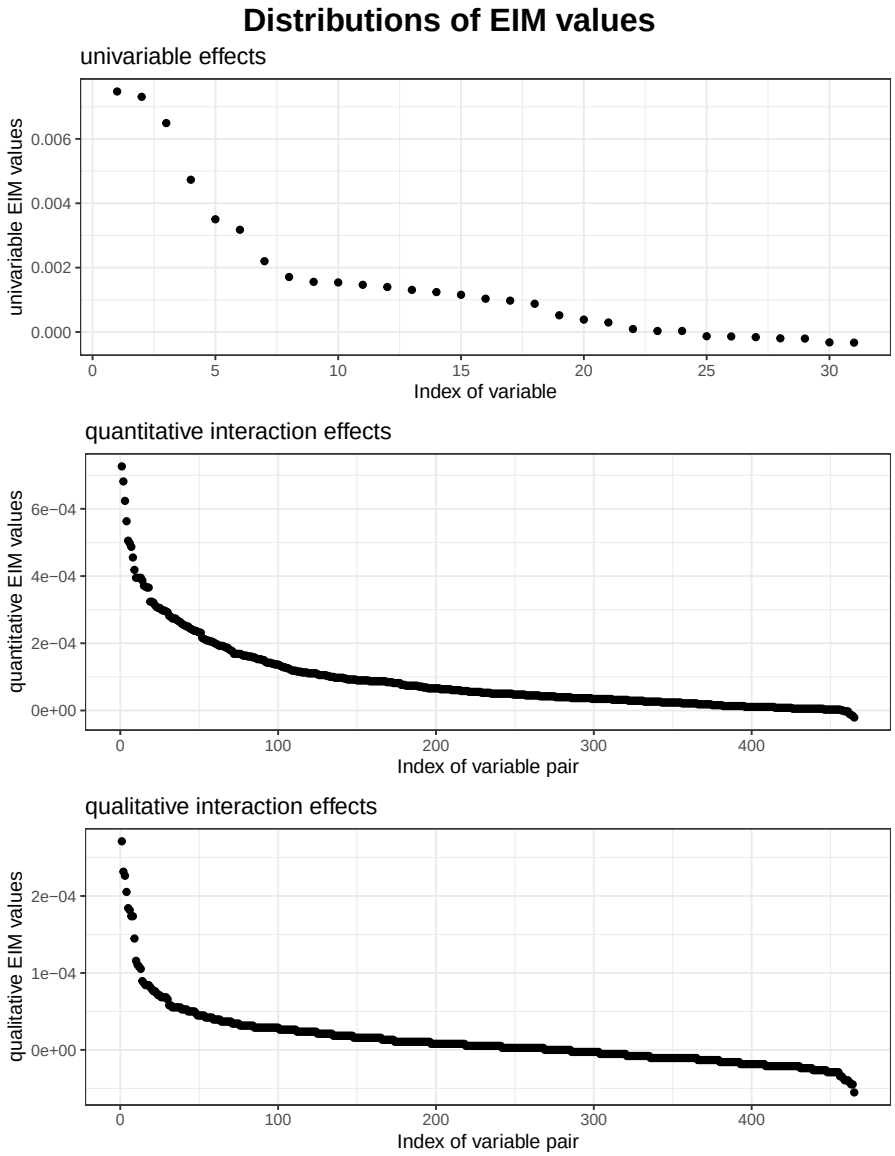

Fig. S13: Result of function `plot.interactionfor()`: EIM values ('white-clover' data set). The values are sorted in decreasing order.

We already investigated the variable pairs with the two largest quantitative EIM values in Supplementary Figure S14. The estimated bivariable influences of the variable pairs with third to fifth largest quantitative EIM values are visualised in Supplementary Figures S16 and S17. The variable pairs in Supplementary Figure S16 again involve the variable `strata` that has many categories, which is why these results are difficult to interpret. The variable pair with fifth largest quantitative EIM value involves two continuous variables (Supplementary Figure S17). The estimated bivariable influence of this variable pair is clearly associated with a quantitative interaction effect: If both `Cocksfoot.93` and `OtherGrasses.94` are small, the predicted amount of white clover is large, but not if only one of them is small or both of them are large.

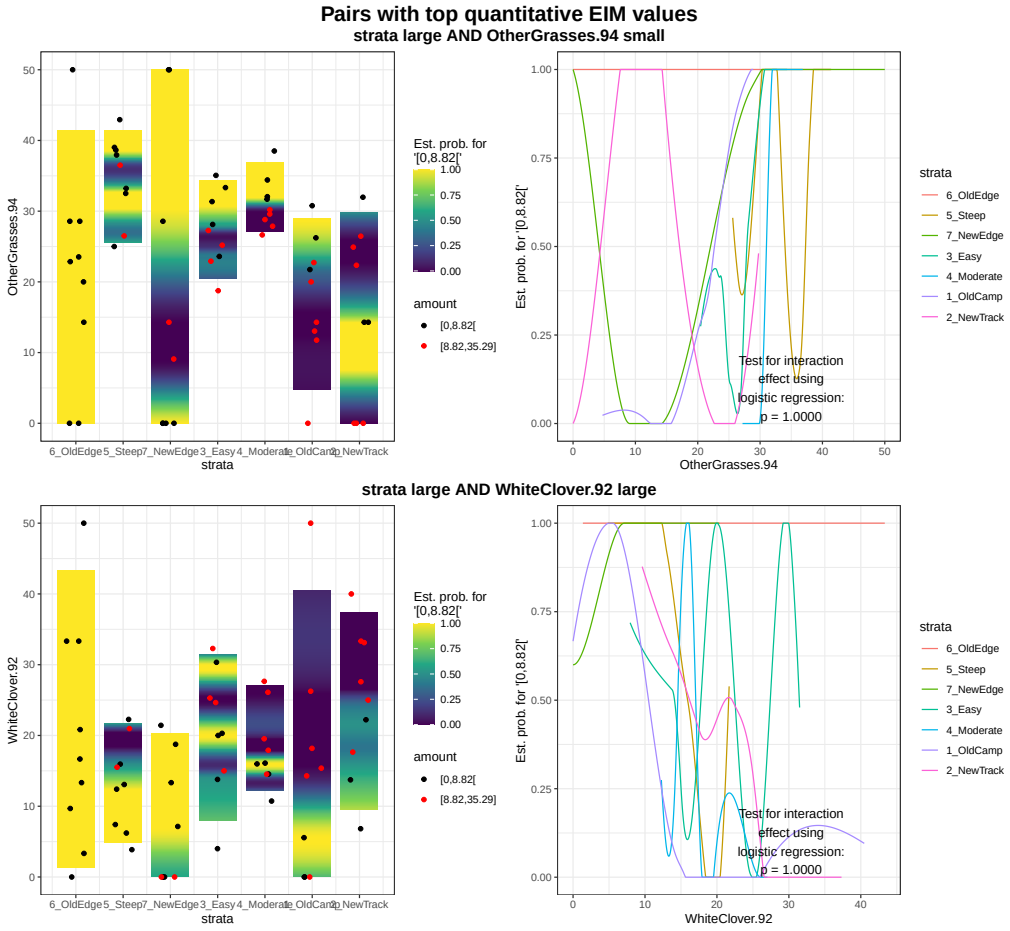

Fig. S14: Result of function `plot.interactionfor()`: Estimated bivariable influences of the two variable pairs with the largest quantitative EIM values ('white-clover' data set).The heat maps in the left panels and the lines in the right panels show (one-dimensional) LOESS fits. The outcome was coded as '1' for '[0, 8.82[' and '0' for '[8.82, 35.29]' when performing the LOESS regression.

Next, we visualise the estimated bivariable influences of the five variable pairs with the largest qualitative EIM values using the following command:

```
plotEffects(model, type="qual")
```

The variable pair with the third largest qualitative EIM value involves the multi-categorical variable `strata` and the corresponding heat map, in the upper-left panel of Supplementary Figure S18, does not suggest a qualitative interaction effect. While the estimated bivariable influence of `plot` and `Weeds.94` visualised in the lower panels of the figure also does not suggest an interaction effect of qualitative type, the two variables do seem to interact: The estimated influence of `Weeds.94` is quite different for `plot="tahora"` or `plot="prop"` than for `plot="hula"`. The estimated bivariable influence of the variable pair with fifth largest qualitative EIM value (Supplementary Figure S19), however, does suggest a qualitative interaction effect. Again, all these results must be interpreted cautiously, on account of the small sample size.

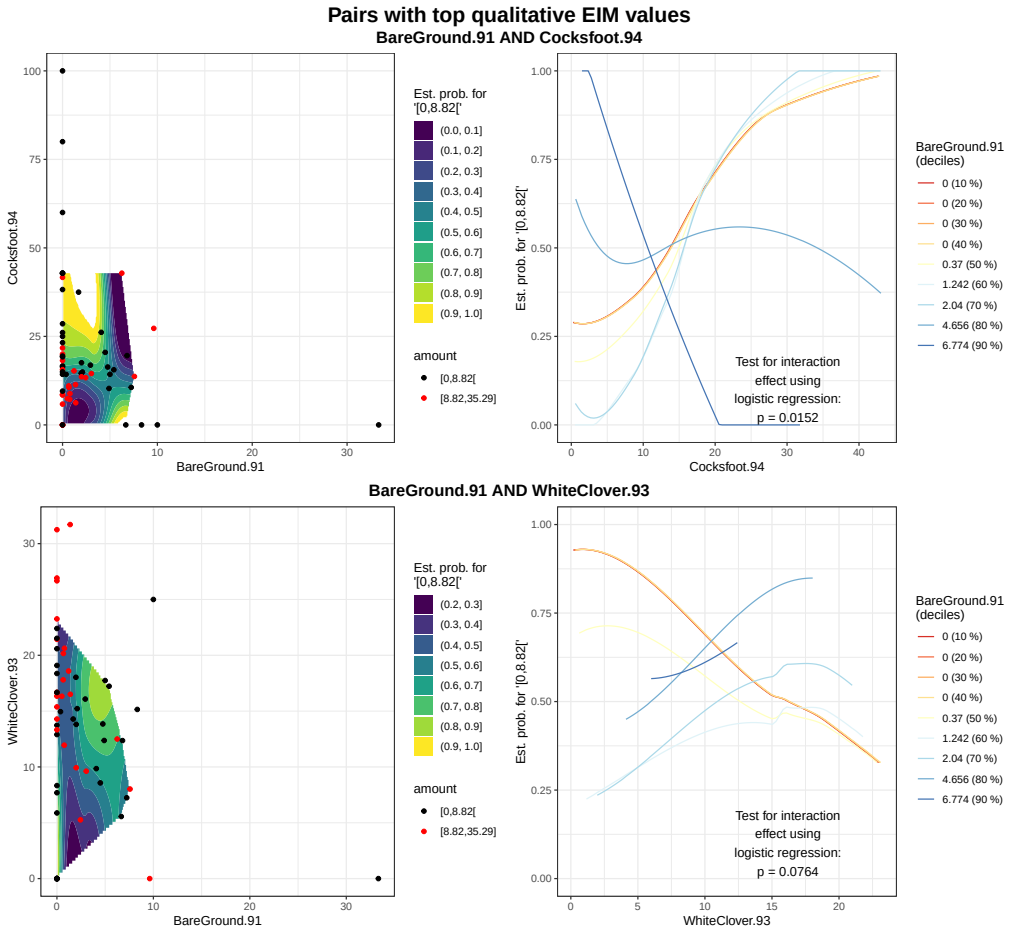

Fig. S15: Result of function `plot.interactionfor()`: Estimated bivariable influences of the two variable pairs with the largest qualitative EIM values ('white-clover' data set). The contour plots in the left panels show two-dimensional LOESS fits. The lines in the right panels show cross-sections of the two-dimensional LOESS fits in the left panels. The outcome was coded as '1' for '[0, 8.82[' and '0' for '[8.82, 35.29]' when performing the LOESS regression.

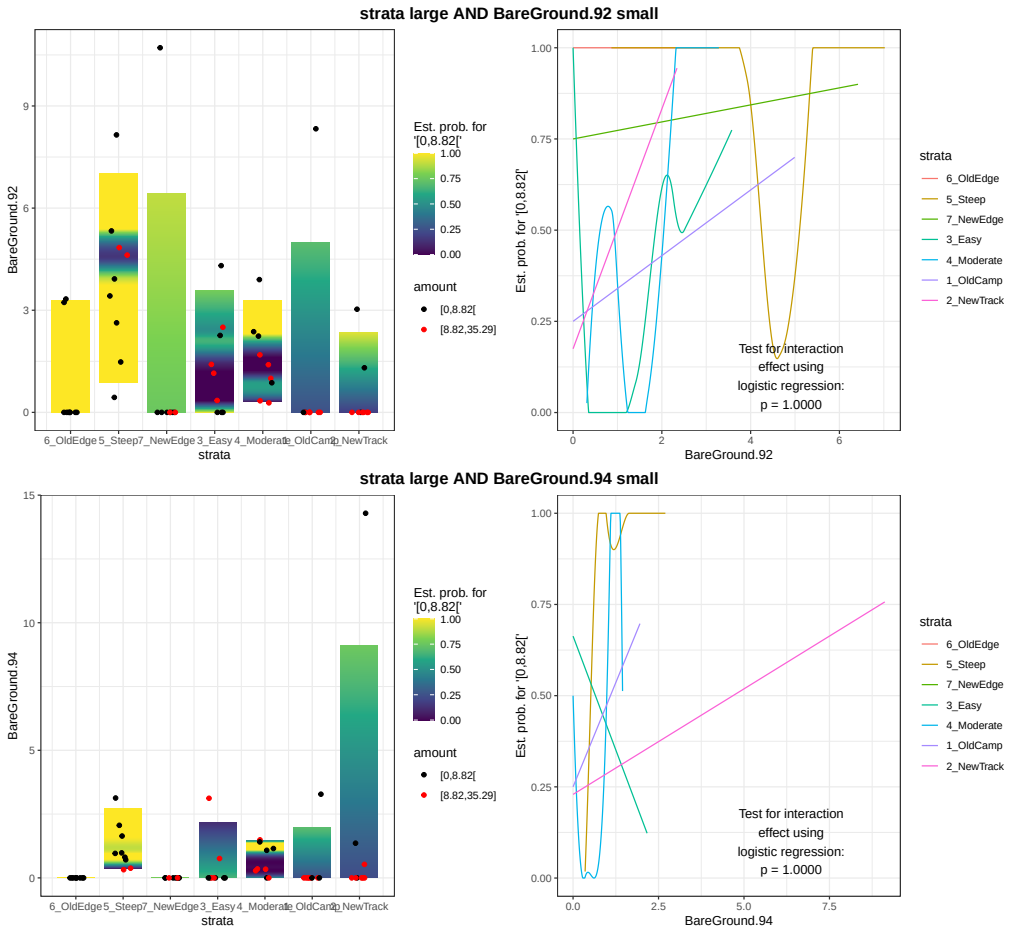

Fig. S16: Result of function `plotEffects()`: Estimated bivariable influences of the two variable pairs with the third and fourth largest quantitative EIM values ('white-clover' data set). The heat maps in the left panels and the lines in the right panels show (one-dimensional) LOESS fits. The outcome was coded as '1' for '[0, 8.82[' and '0' for '[8.82, 35.29]' when performing the LOESS regression.

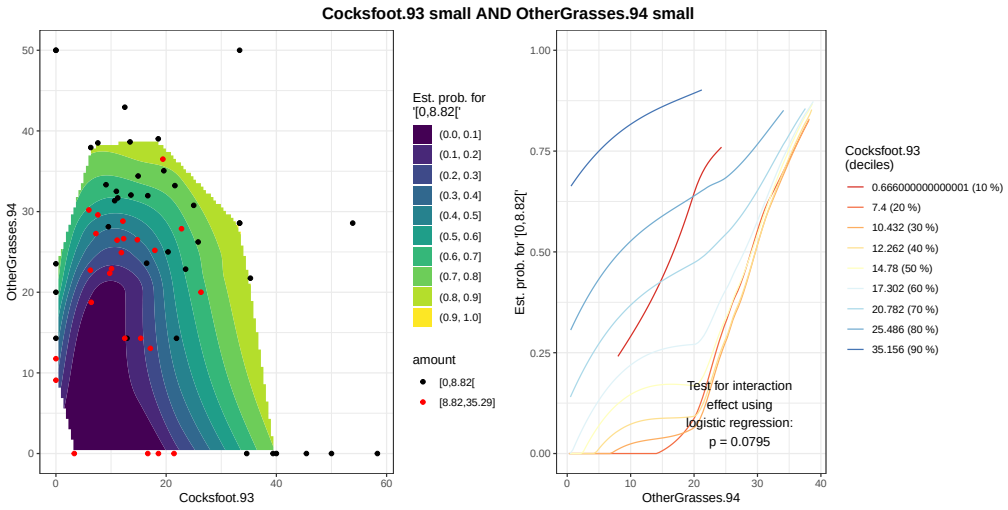

Fig. S17: Result of function `plotEffects()`: Estimated bivariable influence of the variable pair with the fifth largest quantitative EIM value ('white-clover' data set). The contour plot in the left panel shows a two-dimensional LOESS fit. The lines in the right panel show cross-sections of the two-dimensional LOESS fit in the left panel. The outcome was coded as '1' for '[0, 8.82[' and '0' for '[8.82, 35.29]' when performing the LOESS regression.

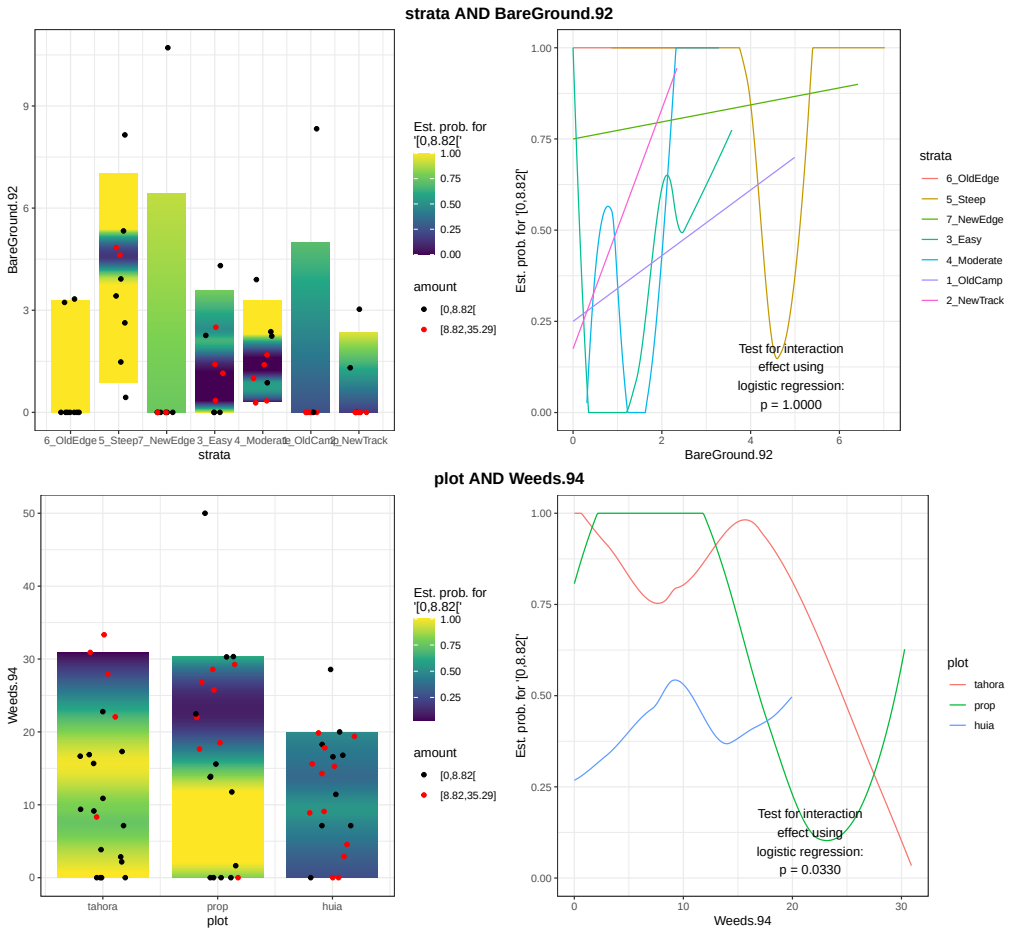

Fig. S18: Result of function `plotEffects()`: Estimated bivariable influences of the two variable pairs with the third and fourth largest qualitative EIM values ('white-clover' data set). The heat maps in the left panels and the lines in the right panels show (one-dimensional) LOESS fits. The outcome was coded as '1' for '[0, 8.82[' and '0' for '[8.82, 35.29]' when performing the LOESS regression.

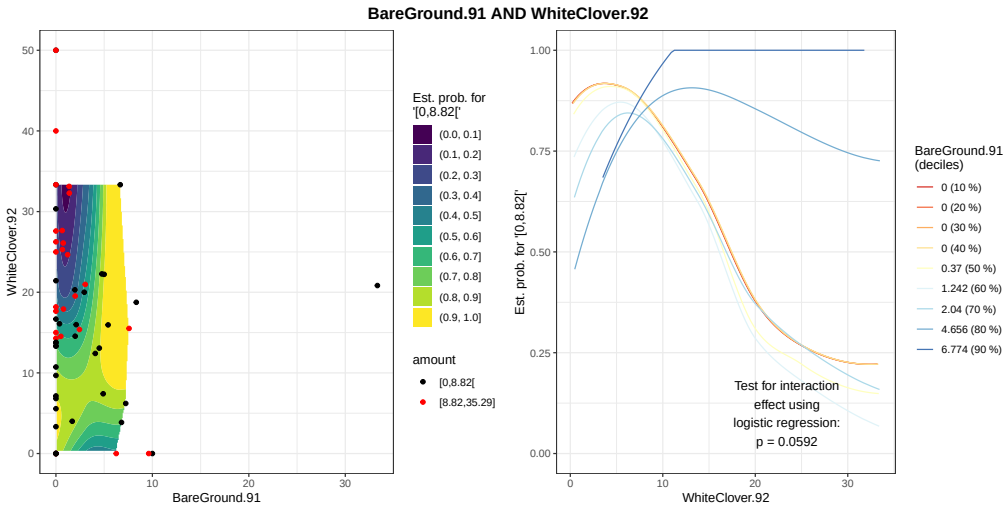

Fig. S19: Result of function `plotEffects()`: Estimated bivariable influence of the variable pair with the fifth largest qualitative EIM value ('white-clover' data set). The contour plot in the left panel shows a two-dimensional LOESS fit. The lines in the right panels show cross-sections of the two-dimensional LOESS fit in the left panel. The outcome was coded as '1' for '[0, 8.82[' and '0' for '[8.82, 35.29]' when performing the LOESS regression.

C.4 'colon-rna' data – high-dimensional data, survival and binary outcome

This data set features 22210 RNA measurements in 350 patients suffering from colon adenocarcinoma. These data were originally downloaded from the database The Cancer Genome Atlas Project (TCGA) (Ley *et al.*, 2013). We used these data previously in Hornung and Wright (2019). Two outcomes were considered: 1) survival (273 (78%) of the 350 patients had censored survival times); 2) the presence or absence of the TP53 mutation as a surrogate for a clinically meaningful binary outcome.

We first consider the survival outcome. Here, in contrast to the previous subsections, we do not only have to specify the function argument `dependent.variable.name`, but also `status.variable.name`, where the latter has to be set to the name of the survival status variable "status":

```
set.seed(1234)
modelsurv <- interactionfor(dependent.variable.name = "time",
                            status.variable.name = "status", data = datarnasurv)
```

In principle, it is also possible to use the `formula` interface with interaction forests (i.e., here: `formula = Surv(time, status) ~ .`). However, for high-dimensional data the `formula` interface can lead to problems in R (stack overflows).

After having constructed the interaction forest, we first apply the `plot()` function:

```
plot(model)
```

The results are shown in Supplementary Figures S20 to S22.

When interpreting the univariable EIM values, it is important to consider that the univariable EIM values of all those variables that did not occur in any of the 5000 pre-selected variable pairs are set to zero in the interaction forest algorithm (cf. Section 3.3 of the main paper). Therefore, variables that have a strong influence on prediction can only be among the variables with the largest univariable EIM values if they were involved in the pre-selected pairs. Another issue associated with univariable EIM values for higher dimensional data is that variables that are featured in a larger number of pre-selected variable pairs can receive too large univariable EIM values; for details see Section 3.3 of the main paper. We will illustrate these issues in the analysis of the binary outcome "TP53 yes vs. TP53 no" shown further below. Note that these issues with the univariable EIM values exist only for higher dimensional data, because for data with at most 100 variables, all possible variable pairs are considered for splitting. However, if the interest lies in measuring the univariable importance of the variables for prediction in high-dimensional data, a conventional random forest should be used and an interaction forest only for ranking the variable pairs with respect to the strengths of their interaction effects.

Two of the variable pairs have distinctively higher qualitative EIM values than the remaining variable pairs (lower panel of Supplementary Figure S20).

The variable pair with the largest quantitative EIM value is not associated with a clear quantitative interaction effect (upper panel of Supplementary Figure S21). Nevertheless, there are some indications of a quantitative interaction effect, as patients with long survival times (i.e., those associated with bright points) are not contained in the upper left region of the heat map. This might

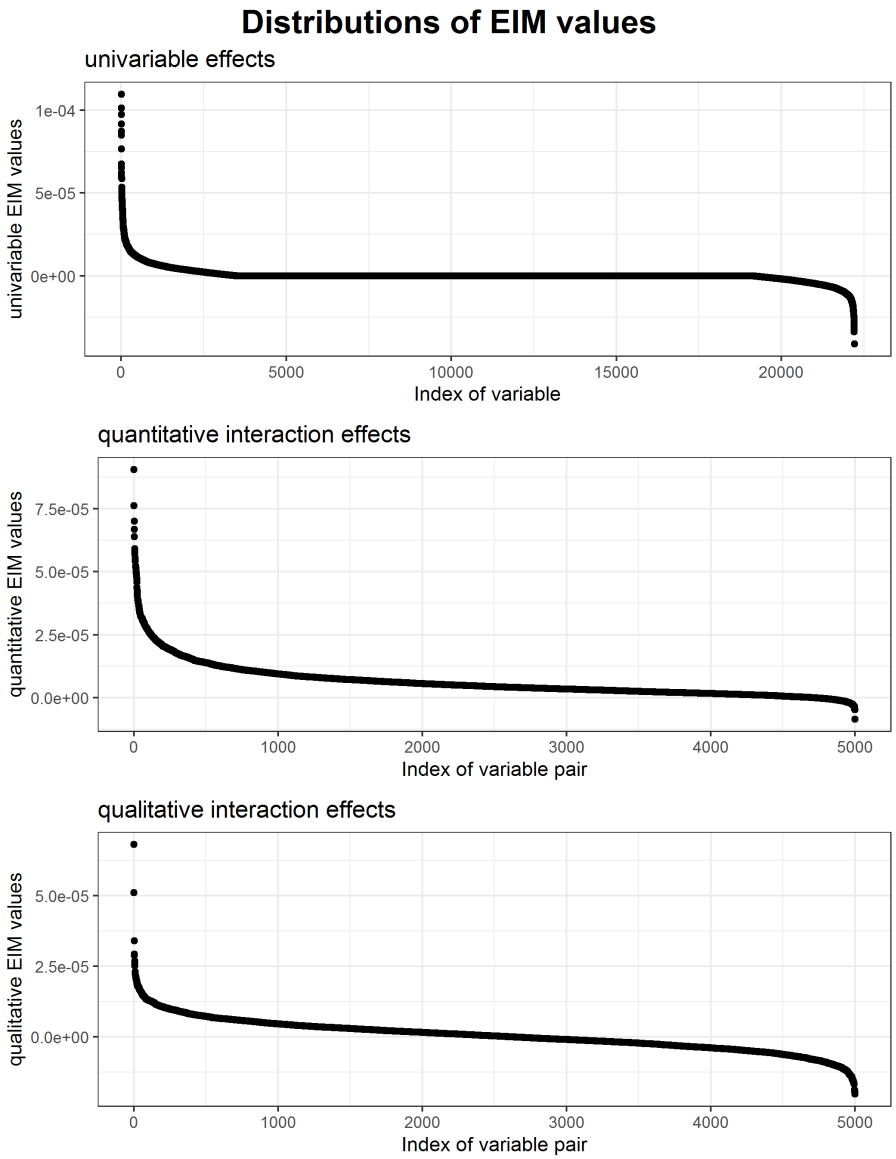

Fig. S20: Result of function `plot.interactionfor()`: EIM values ('colon-rna' data set, survival outcome). The values are sorted in decreasing order.

indicate that patients with small `ENSG00000125970` measurements, but high `ENSG00000229320` measurements are unlikely to live long. For the variable pair with second-largest quantitative EIM value (lower panel of Supplementary Figure S21), there are stronger indications of a quantitative interaction effect: For patients with large `ENSG00000214290` measurements the influence of `ENSG00000215447` on the risk seems to be considerably stronger than for patients with small `ENSG00000214290` measurements. Nevertheless, the test for interaction effect using classical Cox regression was not significant for both variable pairs (with "significant" we mean $p < 0.05$).

The estimated bivariable influences of the two variable pairs with the largest qualitative EIM values are shown in Supplementary Figure S22. The first of these variable pairs shows clear indica-

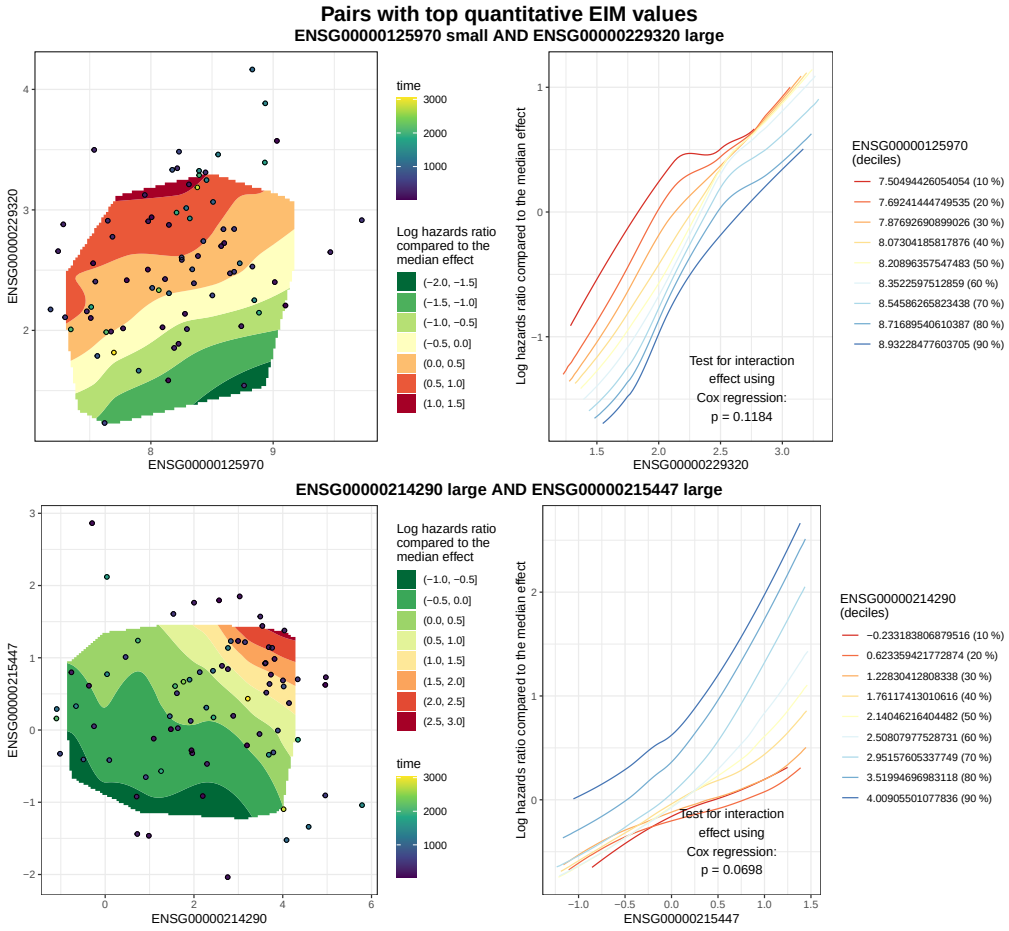

Fig. S21: Result of function `plot.interactionfor()`: Estimated bivariable influences of the two variable pairs with the largest quantitative EIM values ('colon-rna' data set, survival outcome). The contour plots in the left panels show two-dimensional LOESS fits of the log hazards ratio in relation to the median effect, where these LOESS fits were obtained using a Cox proportional hazard additive model. The colored points show the uncensored observations only. The lines in the right panels show cross-sections of the two-dimensional LOESS fits in the left panels.

tions of a qualitative interaction effect: `ENSG00000174516` seems to have a negative influence on the risk for small `ENSG00000133640` measurements, but a positive influence for large `ENSG00000133640` measurements. For the second variable pair (lower panel of Supplementary Figure S22), there are also indications of a qualitative interaction effect, albeit less strong, as in the case of the first variable pair: For small `ENSG00000134318` measurements, `ENSG00000171827` seems to have a positive influence on the risk and for large `ENSG00000134318` measurements, `ENSG00000171827` seems to have a slightly negative influence on the risk. The test for interaction effect using classical Cox regression was significant in both cases.

Again, we also study variable pairs with smaller quantitative and qualitative EIM values than the ones above:

```
plotEffects(modelsurv)
```

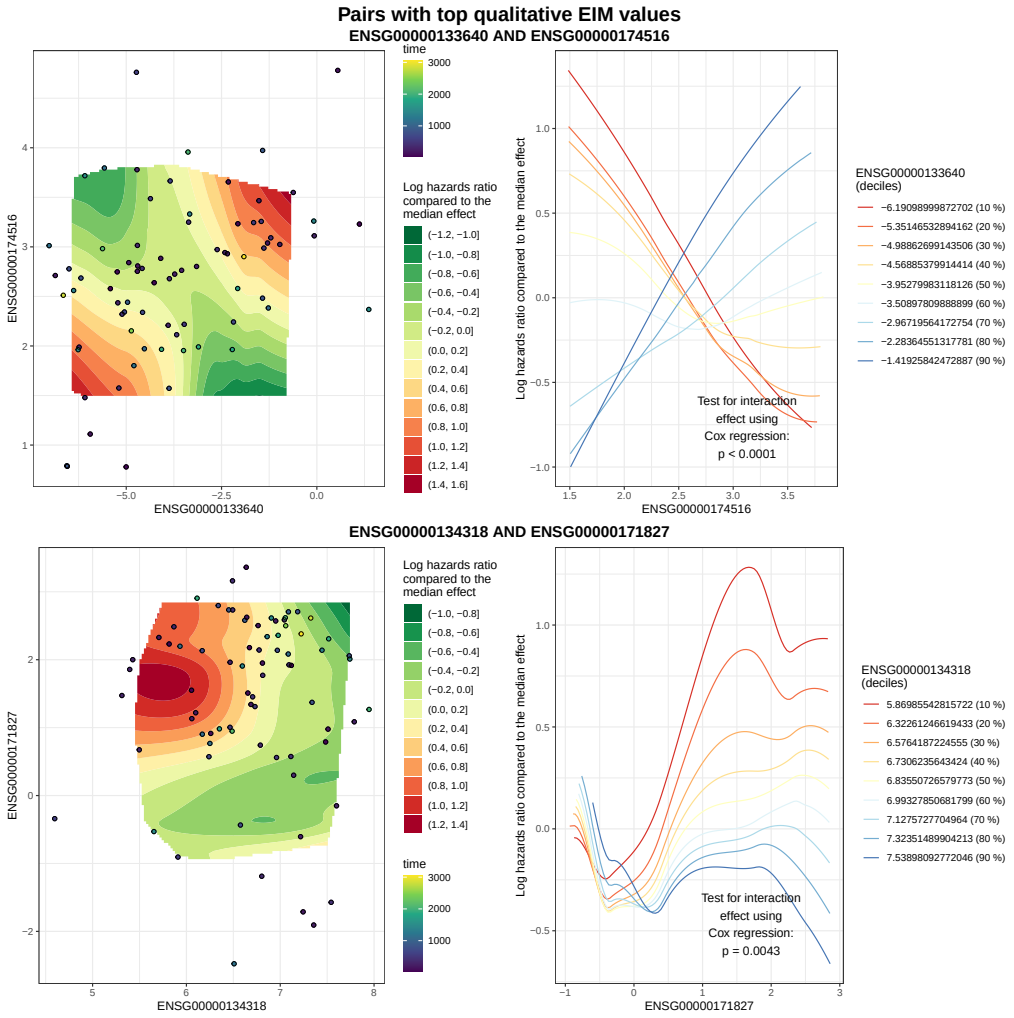

Fig. S22: Result of function `plot.interactionfor()`: Estimated bivariable influences of the two variable pairs with the largest qualitative EIM values ('colon-rna' data set, survival outcome). The contour plots in the left panels show two-dimensional LOESS fits of the log hazards ratio in relation to the median effect, where these LOESS fits were obtained using a Cox proportional hazard additive model. The colored points show the uncensored observations only. The lines in the right panels show cross-sections of the two-dimensional LOESS fits in the left panels.

```
plotEffects(modelsurv, type="qual")
```

We have already looked at the variable pairs with the two largest quantitative EIM values and those with the two largest qualitative EIM values. The remaining plots resulting from the above commands are shown in Supplementary Figures S23 to S26. While the estimated bivariable influences of the variable pairs with third and fourth largest quantitative EIM values (Supplementary Figure S23) do show some indications of quantitative interaction effects, no such effect can be seen in the case of the variable pair with the fifth largest quantitative EIM value (Supplementary Figure S24). The test for interaction effect using classical Cox regression was not significant in all three cases. The plots for the variable pairs with third and fifth largest qualitative EIM val-

ues (upper panel of Supplementary Figure S25 and Supplementary Figure S26, respectively) both show indications of qualitative interaction effects. In addition, the p-values from Cox regression are small here. The plot for the variable pair with the fourth largest qualitative EIM value (lower panel of Supplementary Figure S25) does not show clear indications of a qualitative interaction effect and the result of Cox regression is not significant here.

In practice, it would likely make sense to study even more pairs with large quantitative and qualitative EIM values to avoid failing to identify important effects.

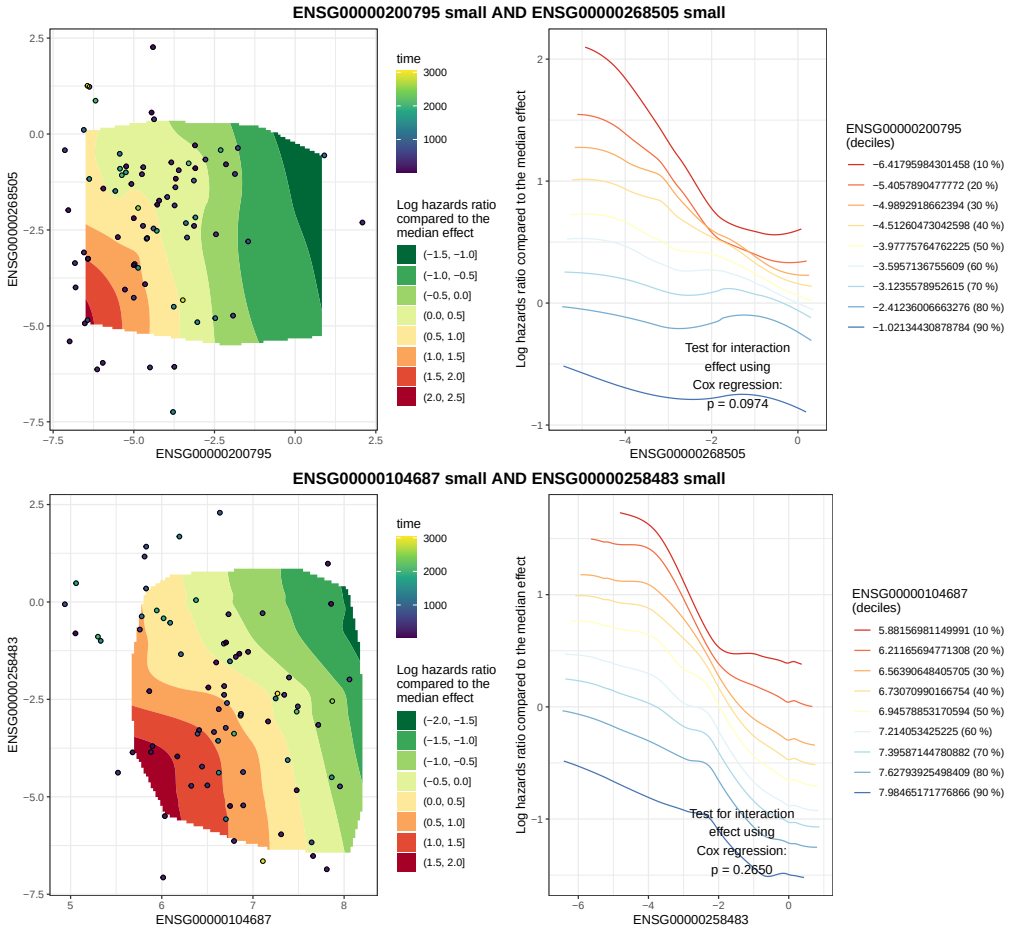

Fig. S23: Result of function `plot.interactionfor()`: Estimated bivariable influences of the two variable pairs with the third and fourth largest quantitative EIM values ('colon-rna' data set, survival outcome). The contour plots in the left panels show two-dimensional LOESS fits of the log hazards ratio in relation to the median effect, where these LOESS fits were obtained using a Cox proportional hazard additive model. The colored points show the uncensored observations only. The lines in the right panels show cross-sections of the two-dimensional LOESS fits in the left panels.

The estimated bivariable influences of the five variable pairs with the largest quantitative EIM values did not suggest strong quantitative interaction effects (in particular, the tests for interaction using classical Cox regression were not significant). It was not clear whether the quantitative EIM values failed in detecting stronger quantitative interaction effects of the types considered with

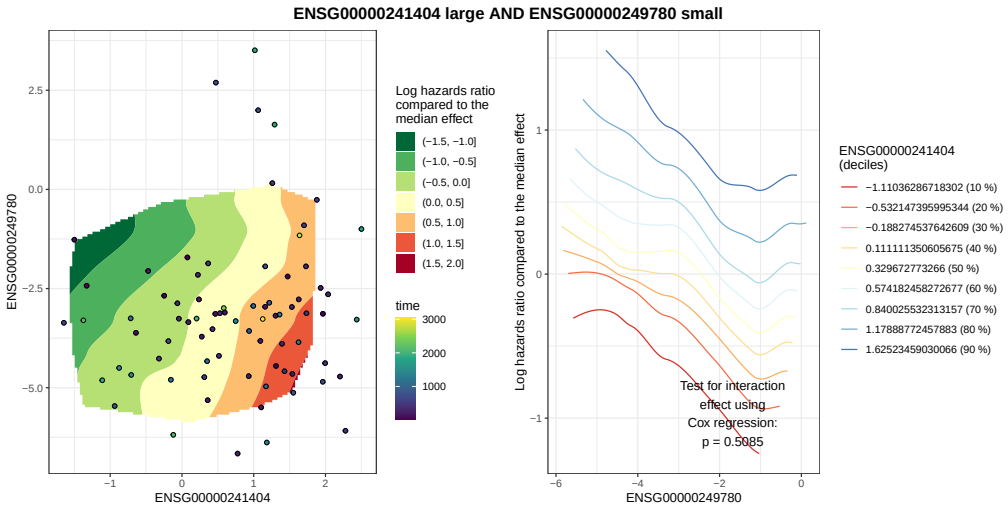

Fig. S24: Result of function `plot.interactionfor()`: Estimated bivariable influence of the two variable pair with the fifth largest quantitative EIM values ('colon-rna' data set, survival outcome). The contour plot in the left panel shows a two-dimensional LOESS fit of the log hazards ratio in relation to the median effect, where this LOESS fit was obtained using a Cox proportional hazard additive model. The colored points show the uncensored observations only. The lines in the right panel show cross-sections of the two-dimensional LOESS fits in the left panel.

interaction forests, or if there are simply no indications of variable pairs with stronger effects of these types. To investigate this issue, we looked at the estimated bivariable influences of various variable pairs with quantitative EIM values ranking much worse. Here, we saw much less indication of quantitative interaction effects, and in particular, the estimates of the log hazard ratio varied much less in the plots compared to in the cases of the variable pairs with top quantitative EIM values. As an example, in Supplementary Figure S27, we show the estimated bivariable influences of the variable pairs with the 100th and 2500th largest quantitative EIM values. These plots can be obtained using the following command:

```
plotEffects(modelsurv, indpairs=c(100, 2500))
```

For both variable pairs, there are no notable indications of quantitative interaction effects. Moreover, the ranges of the estimated log hazard ratios compared to the median effect are much smaller than for the five variable pairs with top quantitative EIM values (Supplementary Figures S21, S23, and S24). This suggests that the importance for prediction is smaller for these two variable pairs with worse ranking quantitative EIM values than for the variable pairs with top-ranking quantitative EIM values.

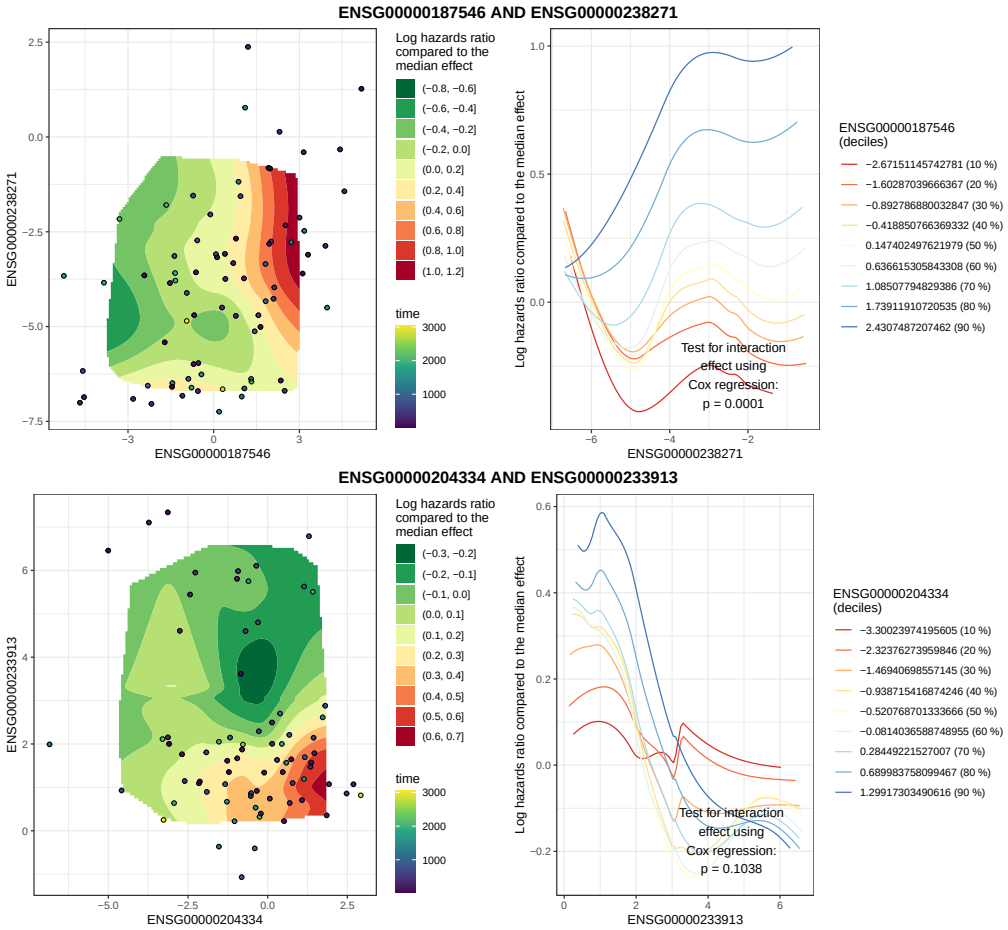

Fig. S25: Result of function `plot.interactionfor()`: Estimated bivariable influences of the two variable pairs with the third and fourth largest qualitative EIM values ('colon-rna' data set, survival outcome). The contour plots in the left panels show two-dimensional LOESS fits of the log hazards ratio in relation to the median effect, where these LOESS fits were obtained using a Cox proportional hazard additive model. The colored points show the uncensored observations only. The lines in the right panels show cross-sections of the two-dimensional LOESS fits in the left panels.

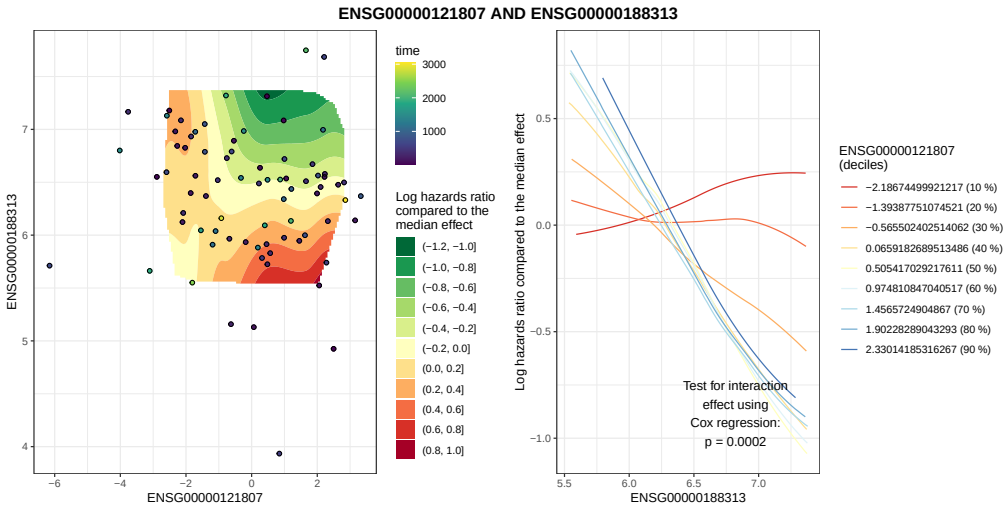

Fig. S26: Result of function `plot.interactionfor()`: Estimated bivariable influence of the variable pair with the fifth largest qualitative EIM values ('colon-rna' data set, survival outcome). The contour plot in the left panel shows a two-dimensional LOESS fit of the log hazards ratio in relation to the median effect, where this LOESS fit was obtained using a Cox proportional hazard additive model. The colored points show the uncensored observations only. The lines in the right panel show cross-sections of the two-dimensional LOESS fits in the left panel.

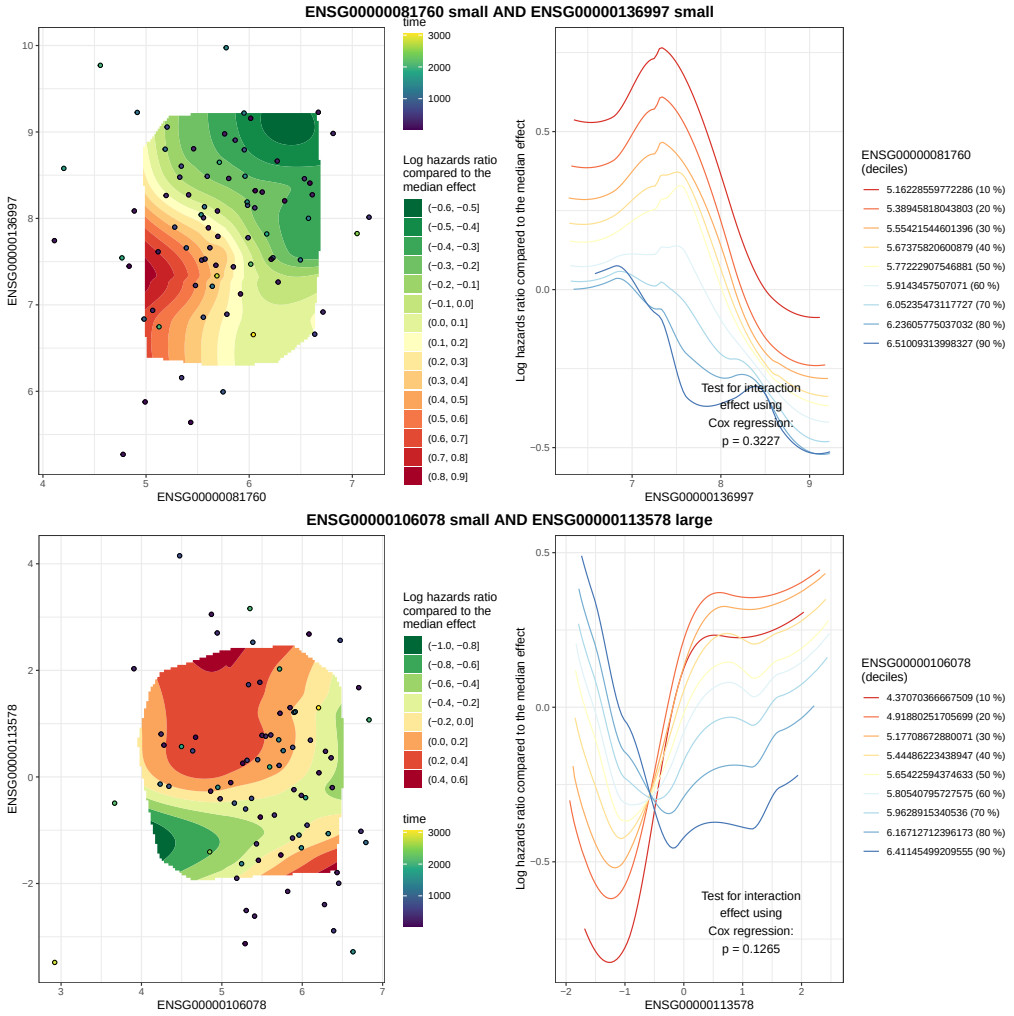

Fig. S27: Result of function `plot.interactionfor()`: Estimated bivariable influences of the two variable pairs with the 100th and 2500th largest quantitative EIM values ('colon-rna' data set, survival outcome). The contour plots in the left panels show two-dimensional LOESS fits of the log hazards ratio in relation to the median effect, where these LOESS fits were obtained using a Cox proportional hazard additive model. The colored points show the uncensored observations only. The lines in the right panels show cross-sections of the two-dimensional LOESS fits in the left panels.

Now we consider the binary outcome "TP53 yes vs. TP53 no", again using the 22210 RNA measurements as covariate variables:

```
set.seed(1234)
modeltp53 <- interactionfor(dependent.variable.name = "TP53", data = datatp53)
```

Next we apply the function `plot()`:

```
plot(modeltp53)
```

Two of the variables have much larger univariable EIM values than all other variables (upper panel of Supplementary Figure S28). These are `ENSG00000174307` and `ENSG00000087088`:

```
head(modeltp53$eim.univ.sorted)
ENSG00000174307 ENSG00000087088 ENSG00000135679 ENSG00000253878 ENSG00000251095
   0.0050254717    0.0044273585    0.0008707547    0.0007452830    0.0005306604
ENSG00000249825
   0.0004778302
```

As already mentioned when describing the results obtained for the survival outcome, one issue of the univariable EIM values in the case of high-dimensional data is that variables that are featured often in the pre-selected pairs are assigned too large univariable EIM values. This issue seems to be the case for the two variables with the largest univariable EIM values: `ENSG00000174307` was featured the second most frequently in the pre-selected variable pairs (63 times) and `ENSG00000087088` the fourth most frequently (36 times). We computed the classical permutation variable importance values of conventional random forests (using the R package `ranger` and 20000 trees) for comparison. Here, `ENSG00000174307` had the 27th largest permutation VIM value among all variables, and `ENSG00000087088` had the 11th largest permutation VIM value. Thus, these variables seem important, but not as important as the univariable EIM values would suggest. The variable, `ENSG00000135679`, that had the third largest univariable EIM value had the second largest permutation VIM value. A second issue for high-dimensional data also mentioned above is that the univariable EIM values miss important variables if these are not featured in the pre-selected pairs. Three of the ten variables with largest permutation VIM values were not featured in the pre-selected pairs and thus received an univariable EIM value of zero. Given the above issues, we again strongly recommend that, if it is of interest to measure the univariable importance of the variables for high-dimensional data sets, a conventional random forest should be constructed for this purpose in addition to the interaction forest used for ranking the interaction effects.

The estimated bivariable influences of both variable pairs with largest quantitative EIM values (Supplementary Figure S29) clearly suggest a quantitative interaction effect: For the first variable pair there is a high concentration of observations with TP53 mutation in the lower-right corner of the heat map and for the second variable pair observations without TP53 mutation prevail in the upper-right corner of the corresponding heat map.

While the variable pair with largest qualitative EIM value seems to be associated with a comparably weak qualitative interaction effect, the picture is clearer in the case of the variable pair with second-largest qualitative EIM value (Supplementary Figure S30).

We investigate variable pairs with smaller quantitative and qualitative EIM values using the commands:

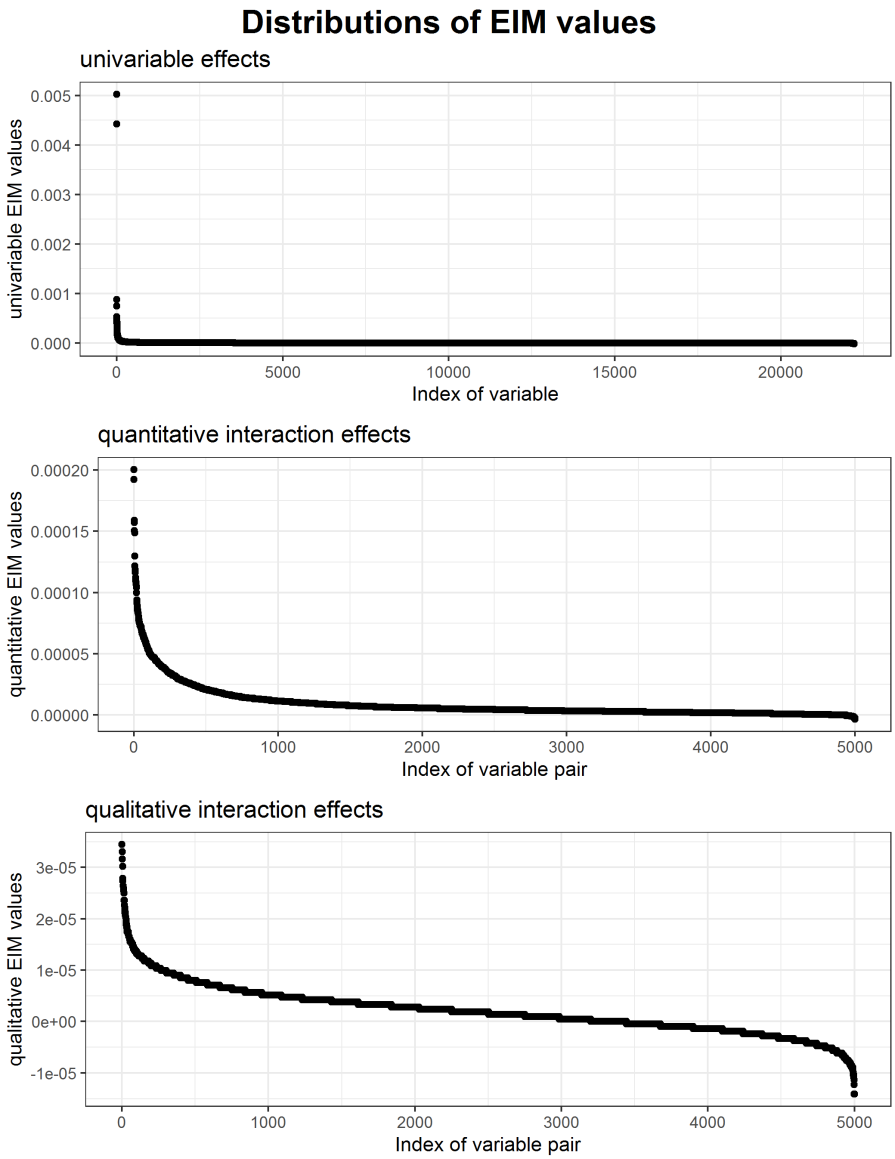

Fig. S28: Result of function `plot.interactionfor()`: EIM values ('colon-rna' data set, binary outcome 'TP53'). The values are sorted in decreasing order.

```
plotEffects(modeltp53)
plotEffects(modeltp53, type="qual")
```

The plots for the variable pairs with third to fifth largest quantitative EIM values all suggest quantitative interaction effects (Supplementary Figures S31 and S32).

However, in the case of the plots for the qualitative EIM values (Supplementary Figures S33 and S34), only the plot of the variable pair with fifth largest qualitative EIM value suggests a qualitative interaction effect. To see whether there are more variable pairs with qualitative interaction effects among those with large qualitative EIM values, we also investigated the pairs with sixth to fifteenth largest qualitative EIM values. Here, we found more variable pairs that

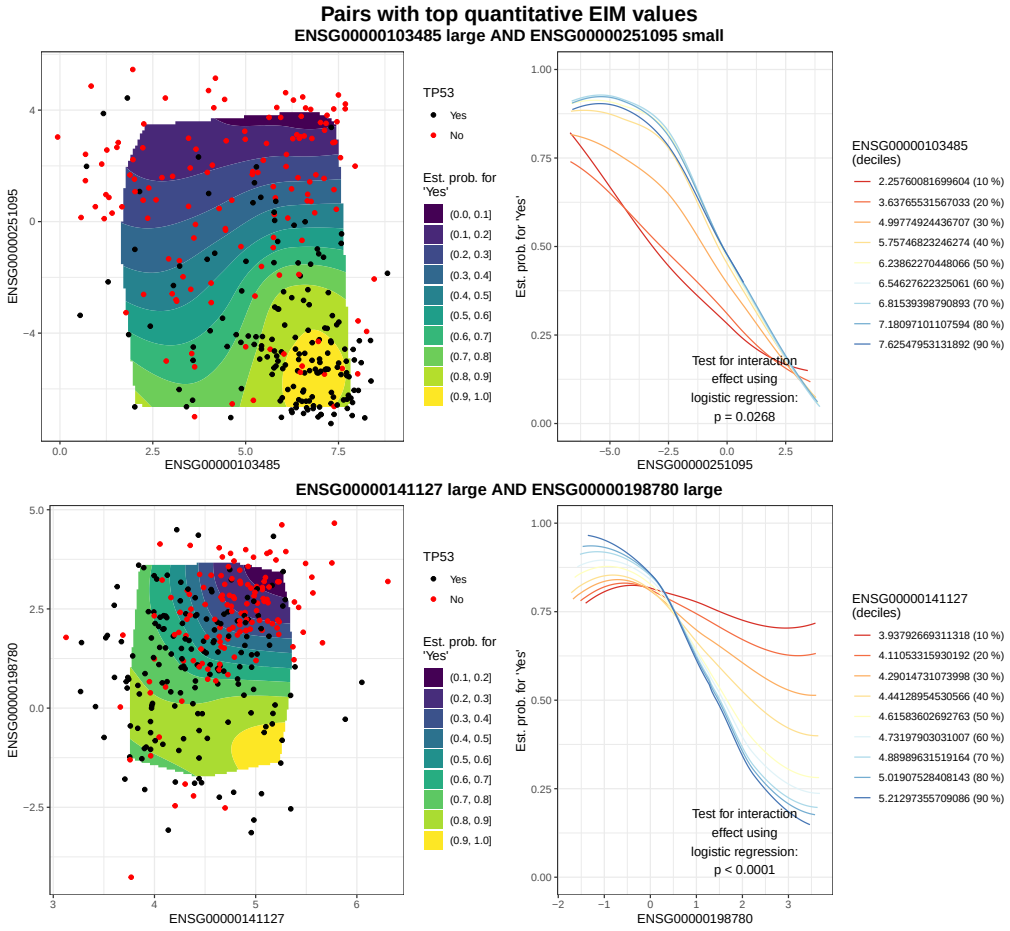

Fig. S29: Result of function `plot.interactionfor()`: Estimated bivariable influences of the two variable pairs with the largest quantitative EIM values ('colon-rna' data set, binary outcome 'TP53'). The contour plots in the left panels show two-dimensional LOESS fits. The lines in the right panels show cross-sections of the two-dimensional LOESS fits in the left panels. The outcome was coded as '1' for 'Yes' and '0' for 'No' when performing the LOESS regression.

showed clear indications of qualitative interaction effects. The clearest indications were seen for variable pairs numbers 7, 9, 10, and 11, the estimated bivariable influence of which we visualise using:

```
plotEffects(modeltp53, type="qual", indpairs=c(7,9,10,11))
```

The resulting plots are shown in Supplementary Figure S35 and S36. These results illustrate that it can be worthwhile to also investigate variable pairs beyond those with largest quantitative or qualitative EIM values.

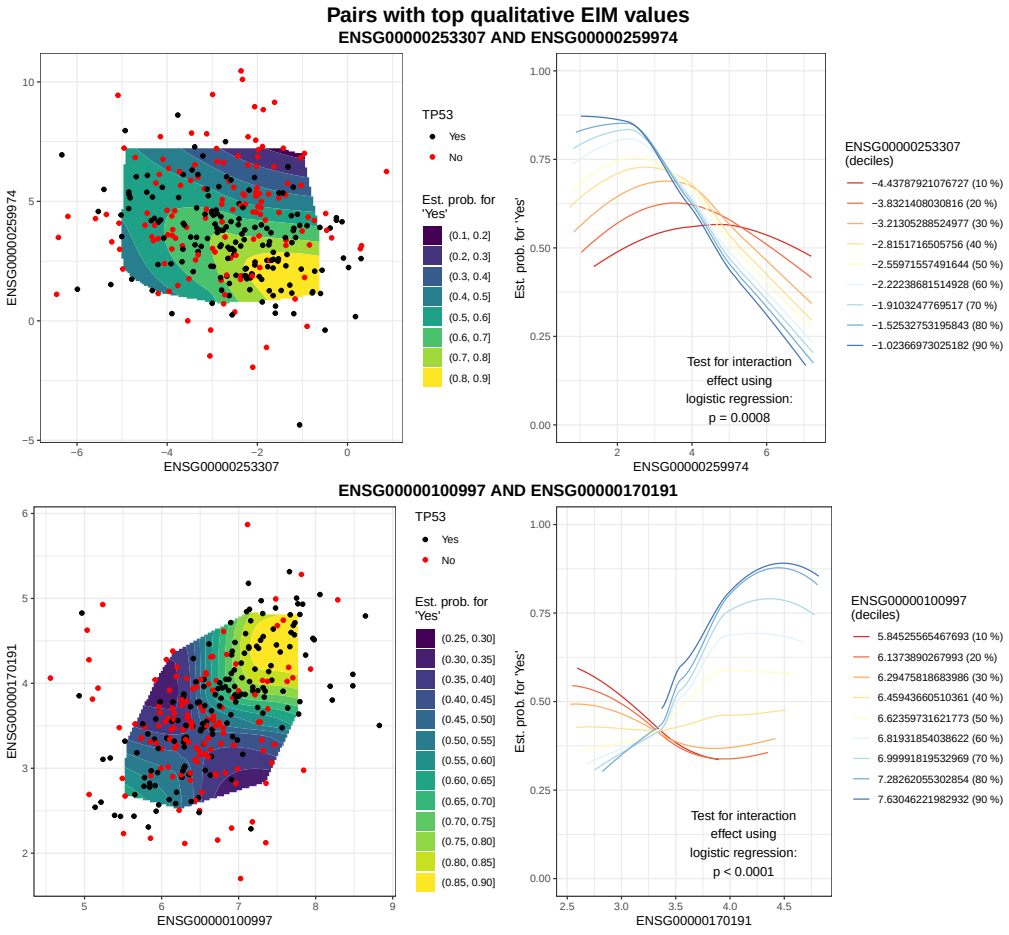

Fig. S30: Result of function `plot.interactionfor()`: Estimated bivariable influences of the two variable pairs with the largest quantitative EIM values ('colon-rna' data set, binary outcome 'TP53'). The contour plots in the left panels show two-dimensional LOESS fits. The lines in the right panels show cross-sections of the two-dimensional LOESS fits in the left panels. The outcome was coded as '1' for 'Yes' and '0' for 'No' when performing the LOESS regression.

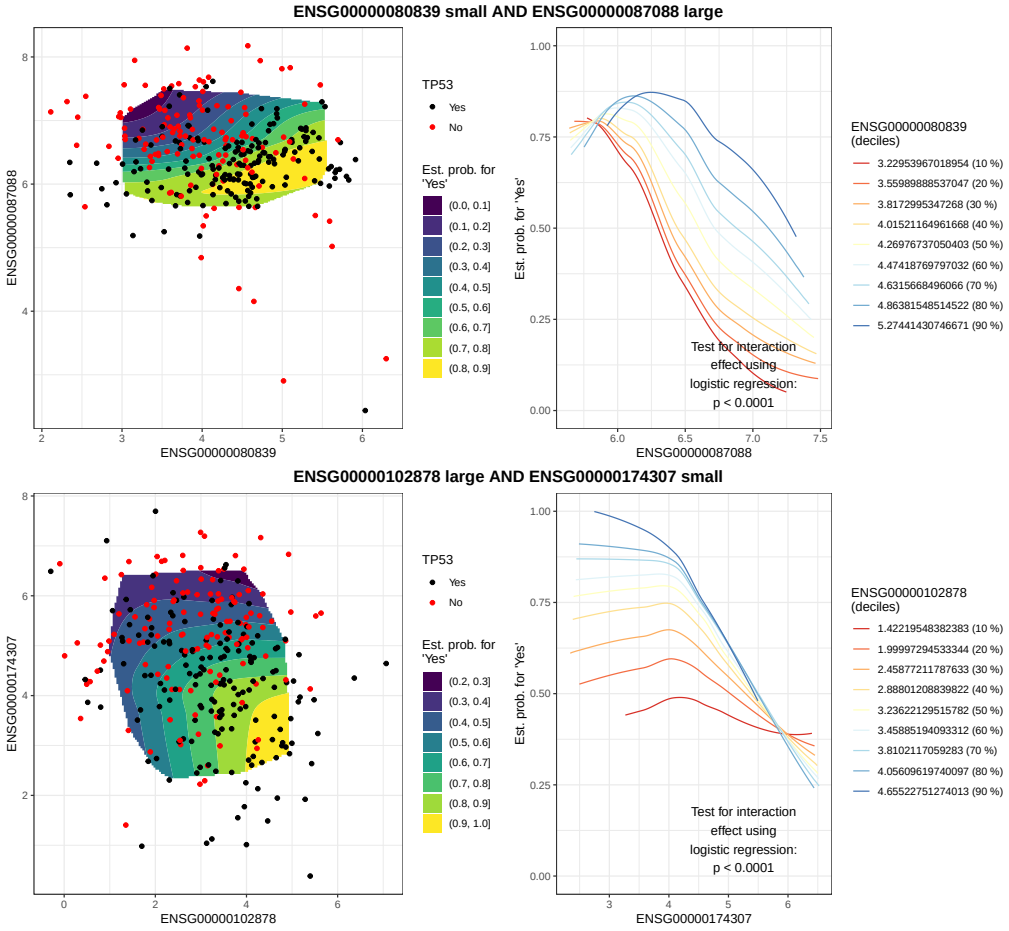

Fig. S31: Result of function `plotEffects()`: Estimated bivariable influences of the two variable pairs with the third and fourth largest quantitative EIM values ('colon-rna' data set, binary outcome 'TP53'). The contour plots in the left panels show two-dimensional LOESS fits. The lines in the right panels show cross-sections of the two-dimensional LOESS fits in the left panels. The outcome was coded as '1' for 'Yes' and '0' for 'No' when performing the LOESS regression.

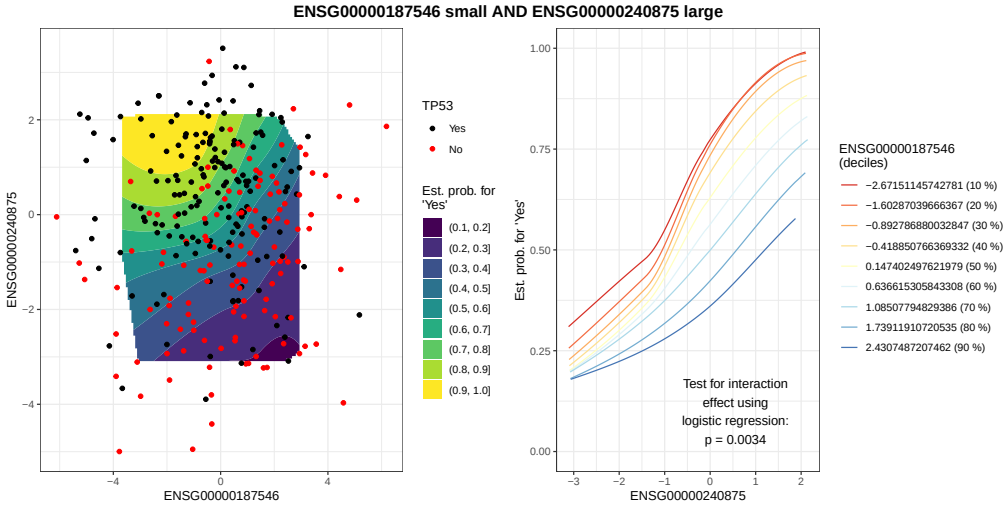

Fig. S32: Result of function `plotEffects()`: Estimated bivariable influences of the variable pair with the fifth largest quantitative EIM value ('colon-rna' data set, binary outcome 'TP53'). The contour plot in the left panel shows a two-dimensional LOESS fit. The lines in the right panel show cross-sections of the two-dimensional LOESS fit in the left panel. The outcome was coded as '1' for 'Yes' and '0' for 'No' when performing the LOESS regression.

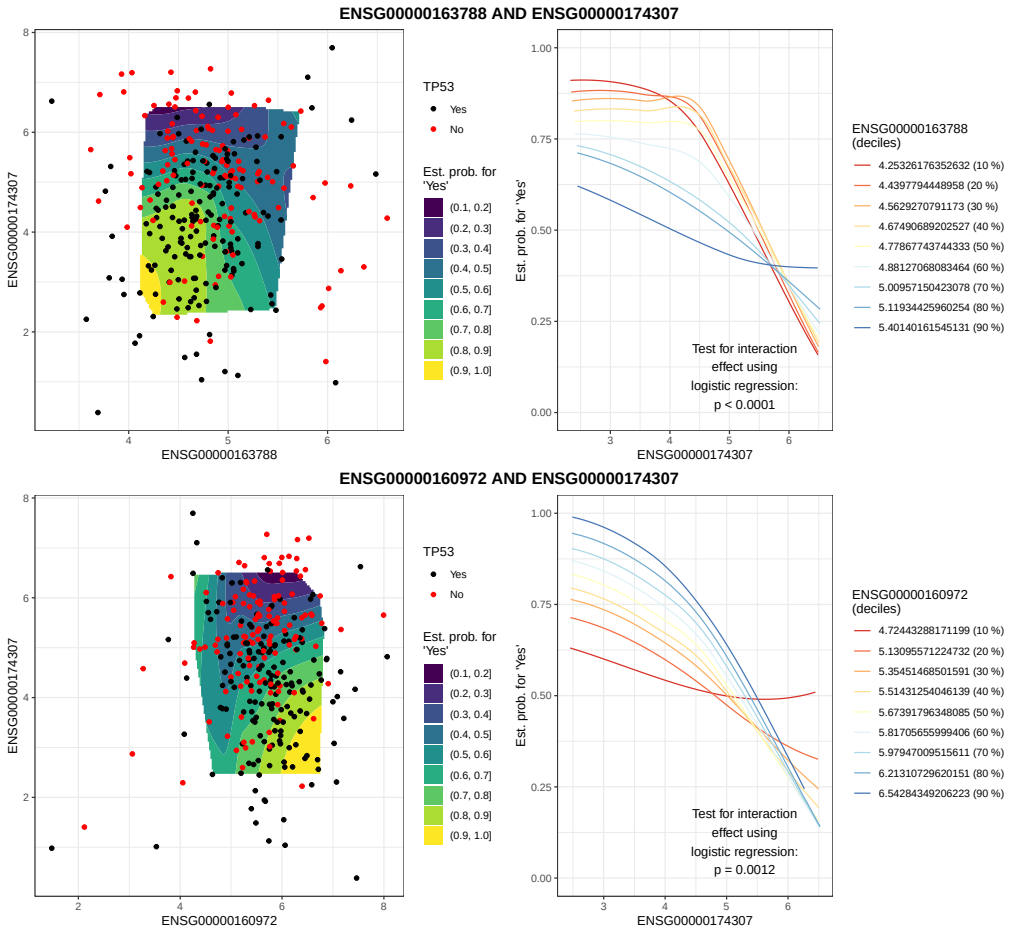

Fig. S33: Result of function `plotEffects()`: Estimated bivariable influences of the two variable pairs with the third and fourth largest qualitative EIM values ('colon-rna' data set, binary outcome 'TP53'). The contour plots in the left panels show two-dimensional LOESS fits. The lines in the right panels show cross-sections of the two-dimensional LOESS fits in the left panels. The outcome was coded as '1' for 'Yes' and '0' for 'No' when performing the LOESS regression.

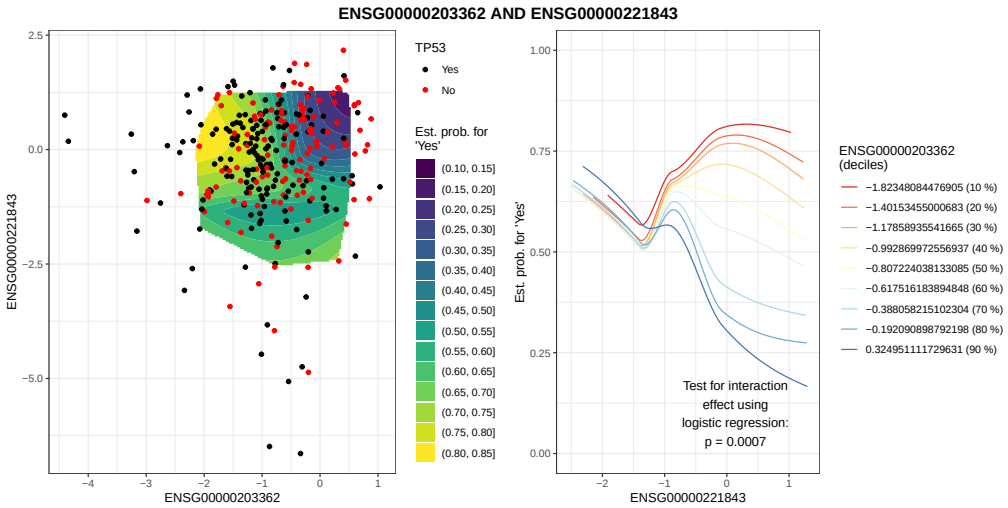

Fig. S34: Result of function `plotEffects()`: Estimated bivariable influences of the variable pair with the fifth largest qualitative EIM value ('colon-rna' data set, binary outcome 'TP53'). The contour plot in the left panel shows a two-dimensional LOESS fit. The lines in the right panel show cross-sections of the two-dimensional LOESS fit in the left panel. The outcome was coded as '1' for 'Yes' and '0' for 'No' when performing the LOESS regression.

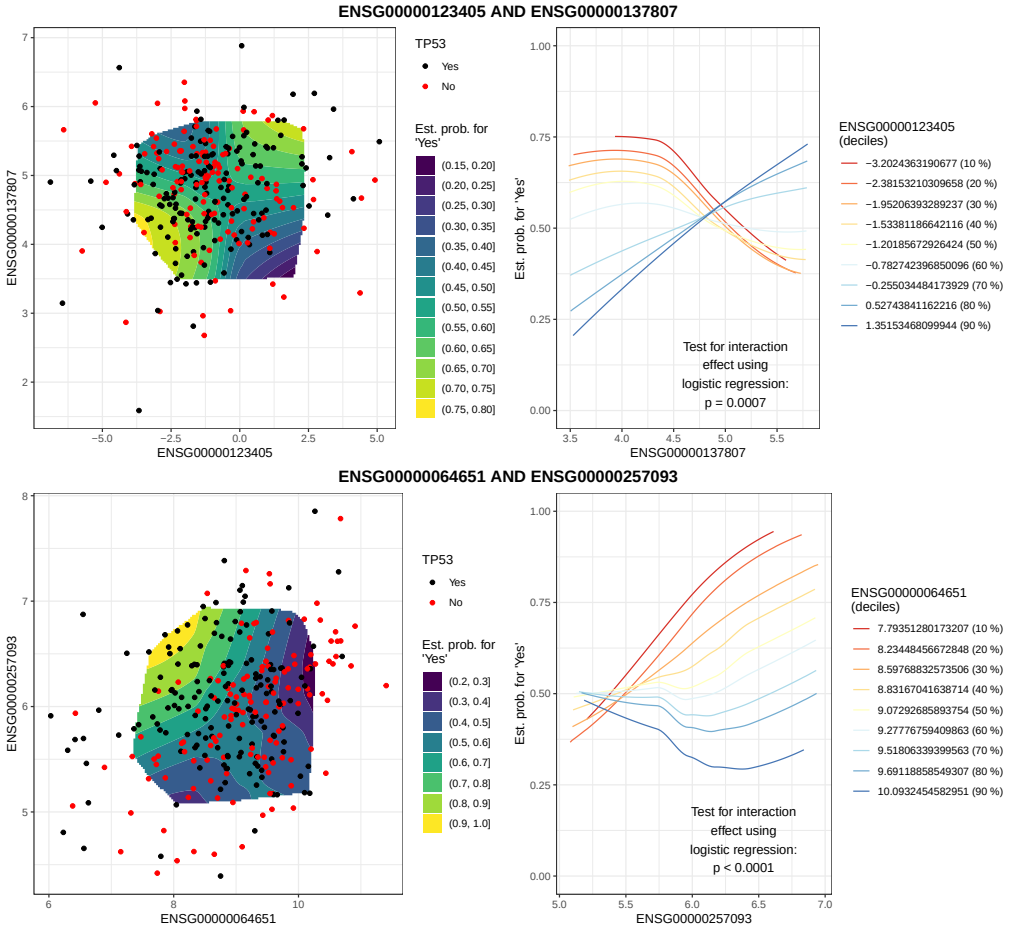

Fig. S35: Result of function `plotEffects()`: Estimated bivariable influences of the two variable pairs with the seventh and ninth largest qualitative EIM values ('colon-rna' data set, binary outcome 'TP53'). The contour plots in the left panels show two-dimensional LOESS fits. The lines in the right panels show cross-sections of the two-dimensional LOESS fits in the left panels. The outcome was coded as '1' for 'Yes' and '0' for 'No' when performing the LOESS regression.

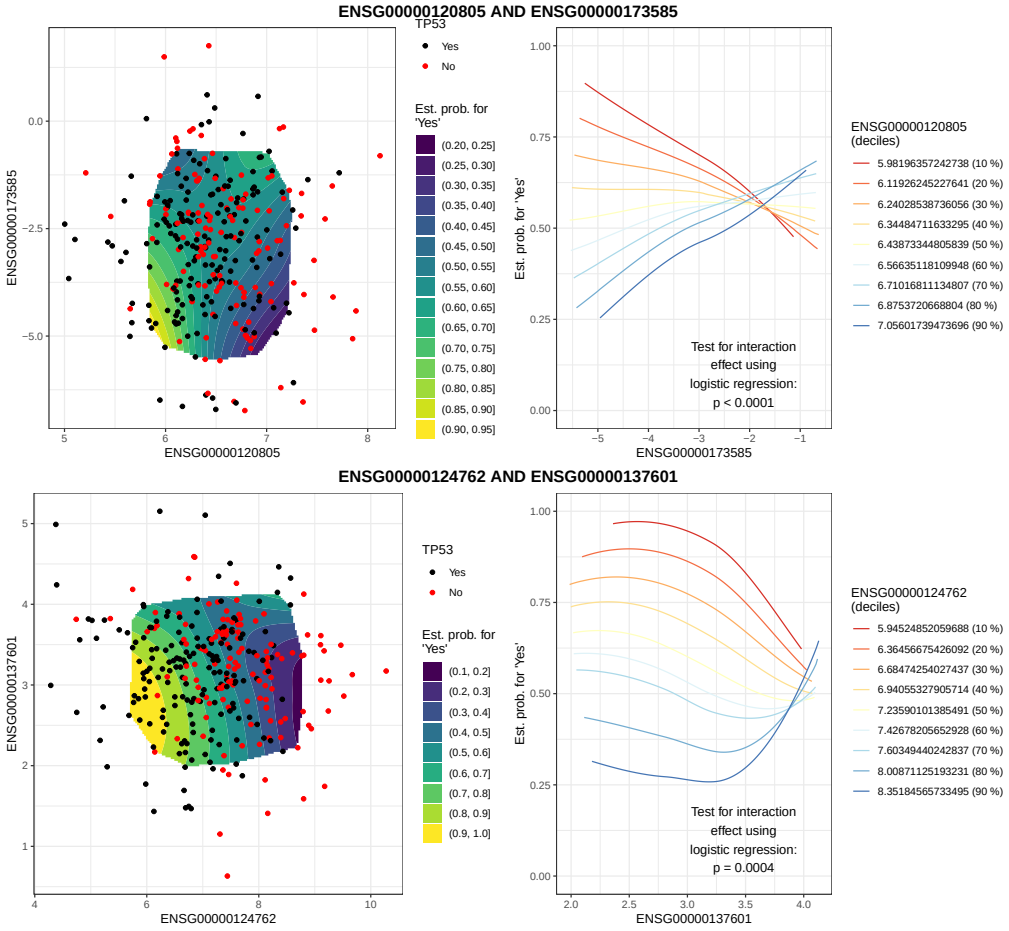

Fig. S36: Result of function `plotEffects()`: Estimated bivariable influences of the two variable pairs with the tenth and eleventh largest qualitative EIM values ('colon-rna' data set, binary outcome 'TP53'). The contour plots in the left panels show two-dimensional LOESS fits. The lines in the right panels show cross-sections of the two-dimensional LOESS fits in the left panels. The outcome was coded as '1' for 'Yes' and '0' for 'No' when performing the LOESS regression.

D Real data study: Further details and results

D.1 Further details on the study design

For both RF and IF, we used the option `probability=TRUE`, which obtains class probability estimates by averaging across the class frequencies in the leaf nodes of the trees in the prediction. Class point predictions were obtained from the class probability estimates using the cutoff 0.5 (i.e., the more likely class was chosen). The R package `ccf` (version 0.1.0) implementing CaF does not feature an option for obtaining class probability estimates. Here, we simply used the proportions of the trees predicting either class as class probability estimates.

When applying the R implementations of CoF, ObF, and RoF to the 220 data sets in the comparison, the computations stopped with errors for some data sets. We were able to fix the great majority of these issues by performing respective changes in the R codes underlying the R packages implementing these methods. For details, see Supplementary Material 2 accompanying the paper, where all R codes used to obtain the results shown in the paper are available. After fixing issues in the R implementations of CaF, ObF, and RoF, four of the 5500 applications of the stratified cross-validation still resulted in errors: There were three errors in the case of ObF and one error in the case of CaF. All three concerned data sets were strongly imbalanced. Because of these four errors, both for CaF and ObF, there was one data set for which results were not available for one of the five repetitions of the cross-validation. For ObF there was one data set in addition for which results were not available for two of the five repetitions. When calculating the averages across the five repetitions of the cross-validation for each combination of data set and method, we simply ignored these few missing results.

D.2 Dependencies of the ranks the methods achieved with respect to the different metrics on the numbers of variables and the sample sizes

The dependencies of the ranks of the methods with respect to the ACC on the number of variables in the data sets are visualised in Supplementary Figure S37. The corresponding results with respect to the AUC and the Brier are shown in Supplementary Figures S38 and S39. Following Couronné *et al.* (2018), we will focus on the ACC. The results obtained for the AUC and the Brier will be compared to those obtained for the ACC. The ranks of IF and RF hardly seem to be influenced by the number of variables in Supplementary Figure S37. As described in Section 4.1.3 of the main paper, CaF achieved the best rank frequently for data sets with small numbers of variables. However, CaF was also often among the worst methods for such data sets. As a result, CaF achieved only slightly better mean ranks for data sets with small numbers of variables. ObF achieved remarkably better ranks for data sets with large numbers of variables, for which the mean performance of ObF was comparable to that of IF and slightly better than that of RF. Interestingly, RoF achieved better ranks for very small numbers of variables. We did not observe this for the AUC (Supplementary Figure S38). Apart from the latter observation, the dependency structures observed for the AUC and the Brier are quite consistent with that observed for the ACC. For the Brier, the mean ranks of IF were slightly better for data sets with small numbers of variables. However, the confidence intervals are very broad for large number of variables, which

is why this result should not be overinterpreted.

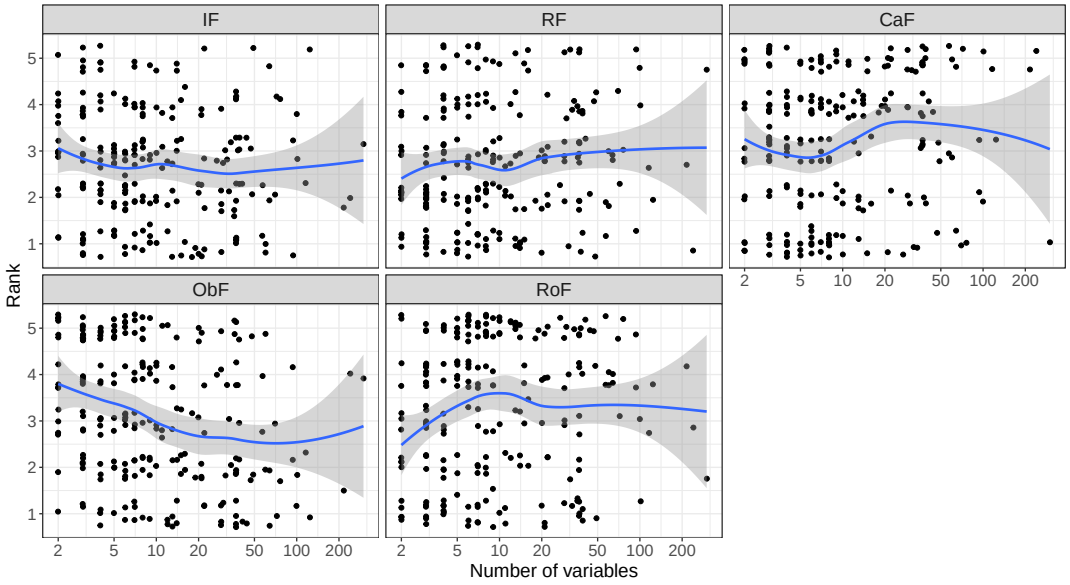

Fig. S37: Dependencies of the method ranks with respect to the ACC on the numbers of variables in the data sets. The black dots show the ranks the methods achieved for the individual data sets. We added random noise to the ranks to make it possible to discern tuples of points at the same positions. The blue lines show LOESS fits and the gray bands 95% pointwise confidence intervals. The x-axes are shown on log scale.

In general, the dependencies of the mean ranks of the methods on the sample size seem to be weak (Supplementary Figures S40, S41, and S42). A notable exception for each performance metric was that the ranks of RoF were considerably worse for very large sample sizes, where this method almost always took the last place for the studied data sets. This suggests that RoF is less capable of exploiting large sample sizes than the other compared methods. For the AUC, IF achieved slightly better mean ranks for large sample sizes.

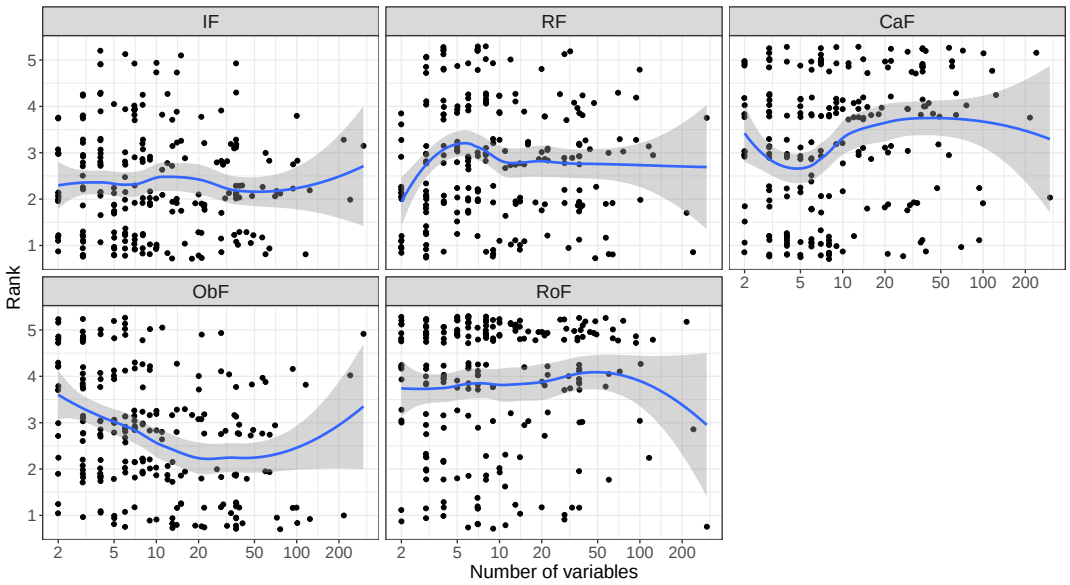

Fig. S38: Dependencies of the method ranks with respect to the AUC on the numbers of variables in the data sets. The black dots show the ranks the methods achieved for the individual data sets. We added random noise to the ranks to make it possible to discern tuples of points at the same positions. The blue lines show LOESS fits, and the gray bands 95% pointwise confidence intervals. The x-axes are shown on log scale.

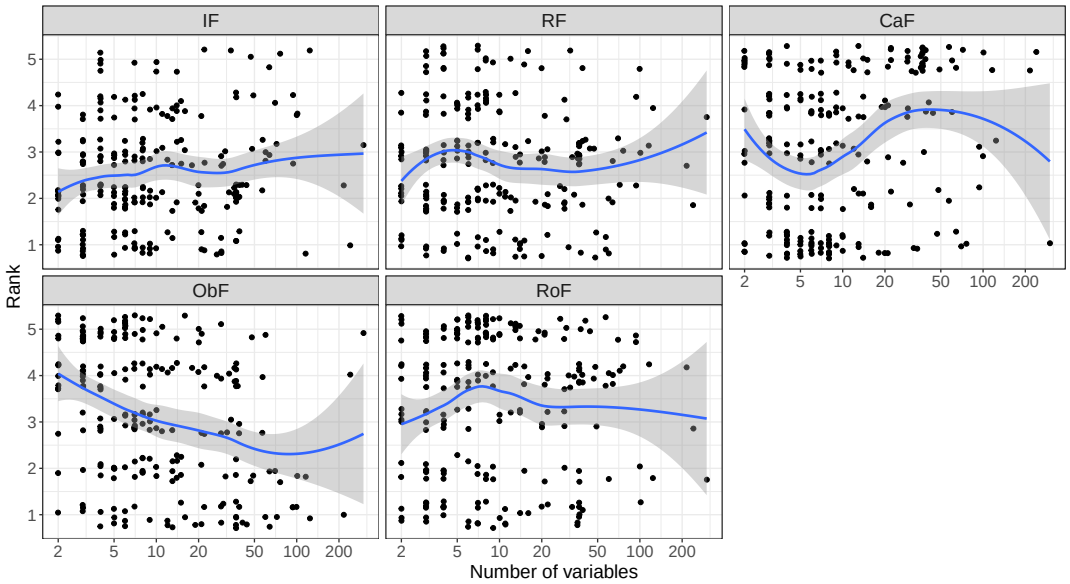

Fig. S39: Dependencies of the method ranks with respect to the Brier on the numbers of variables in the data sets. The black dots show the ranks the methods achieved for the individual data sets. We added random noise to the ranks to make it possible to discern tuples of points at the same positions. The blue lines show LOESS fits, and the gray bands 95% pointwise confidence intervals. The x-axes are shown on log scale.

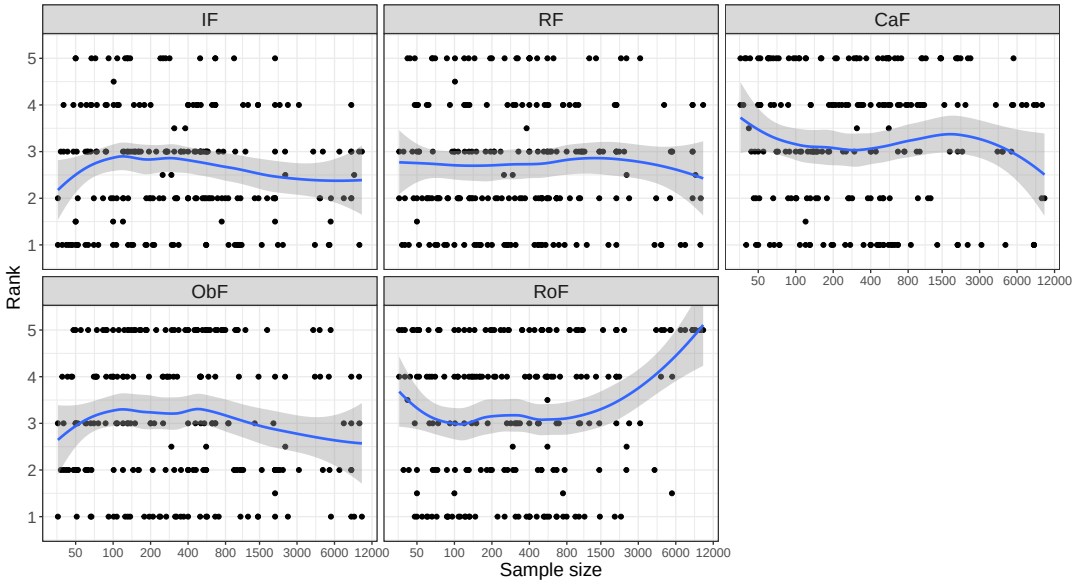

Fig. S40: Dependencies of the method ranks with respect to the ACC on the sample sizes. The black dots show the ranks the methods achieved for the individual data sets. The blue lines show LOESS fits, and the gray bands 95% pointwise confidence intervals. The x-axes are shown on log scale.

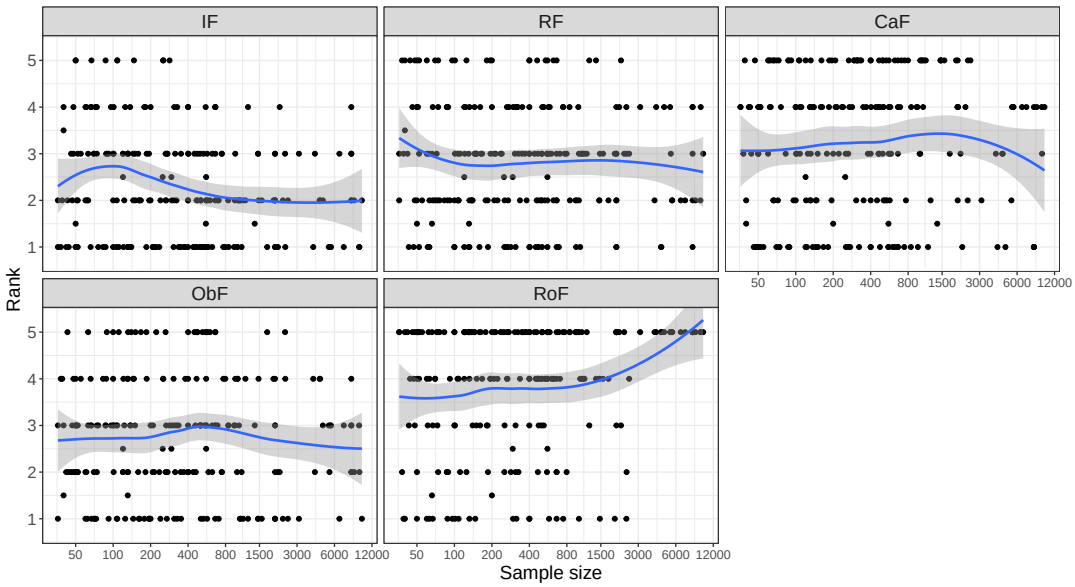

Fig. S41: Dependencies of the method ranks with respect to the AUC on the sample sizes. The black dots show the ranks the methods achieved for the individual data sets. The blue lines show LOESS fits, and the gray bands 95% pointwise confidence intervals. The x-axes are shown on log scale.

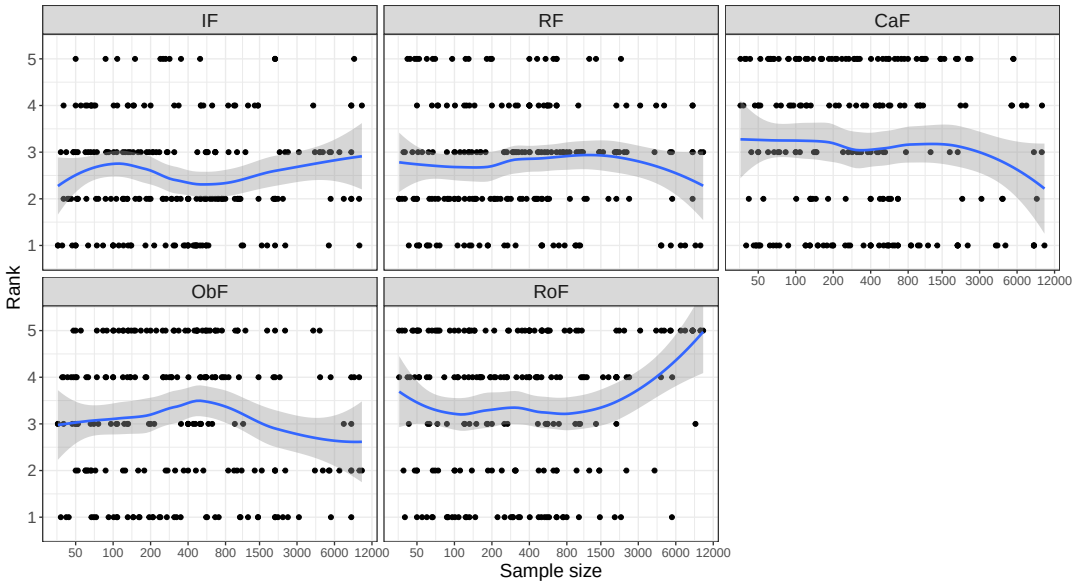

Fig. S42: Dependencies of the method ranks with respect to the Brier on the sample sizes. The black dots show the ranks the methods achieved for the individual data sets. The blue lines show LOESS fits, and the gray bands 95% pointwise confidence intervals. The x-axes are shown on log scale.

E Simulation study: Further details on the study design and the simulation setting

E.1 Further details on the study design

As described in Section 4.2.1 of the main paper, in the cases of iRF and – in particular – PA it was not feasible computationally to use 20000 trees per forest for the largest considered sample size ($n = 1000$). A single simulation iteration did not finish in 40 hours when using 20000 trees in the case of the largest sample size. Therefore, for data sets with $n = 1000$, we used the numbers of trees 1000 and 500 for PA and iRF, respectively, which correspond to the default values in the R packages `randomForestSRC` (version 2.9.3) and `iRF` (version 2.0.0) implementing these approaches.

While EIM, PA, and IMDMS return interaction importance scores for each pair of variables, iRF returns comparably short lists of tuples that are candidates for tuples featuring high-order interactions. Therefore, the results obtained using iRF are not directly comparable to those obtained using EIM, PA, and IMDMS. Be S the list of the tuples identified by iRF sorted according to the stability score in decreasing order. We defined the rank attributed to a variable pair using iRF as the index of the first tuple in S that contained both members of the variable pair. If the list of identified tuples did not contain tuples that featured both variables at all, we simply set the rank of the interacting pair that is attributed by iRF to missing and skip the corresponding simulation iteration when evaluating the results for iRF. Note that iRF is clearly given an advantage here: First, the lists of tuples identified by iRF are much shorter than the lists of all possible variable pairs, which is why the ranks obtained using iRF tend to be much lower. Second, for many of the simulation iterations the resulting lists of tuples identified by iRF do not contain tuples that contain both variables of interest. In these cases, iRF clearly did not identify the respective variable pair as an interacting pair, which is why just leaving these simulation iterations out gives iRF an advantage. Third, many of the tuples containing both variables of interest also contain other variables. In these cases, the order of the interaction effect attributed by iRF is higher than that of the actual interaction effect, which is only two. Nevertheless, as also described in Section 4.2.1 of the main paper, we decided to follow this overoptimistic procedure of attributing ranks using iRF, because we wanted to avoid putting iRF at a disadvantage. Our goal was to study, whether IF tends to outperform the competing approaches or not. If IF would still perform better than iRF, even if the latter is put at an advantage, we would have more certainty that IF truly performs better than iRF in the investigated context.

E.2 Exemplary pairs of variables in a simulated data set

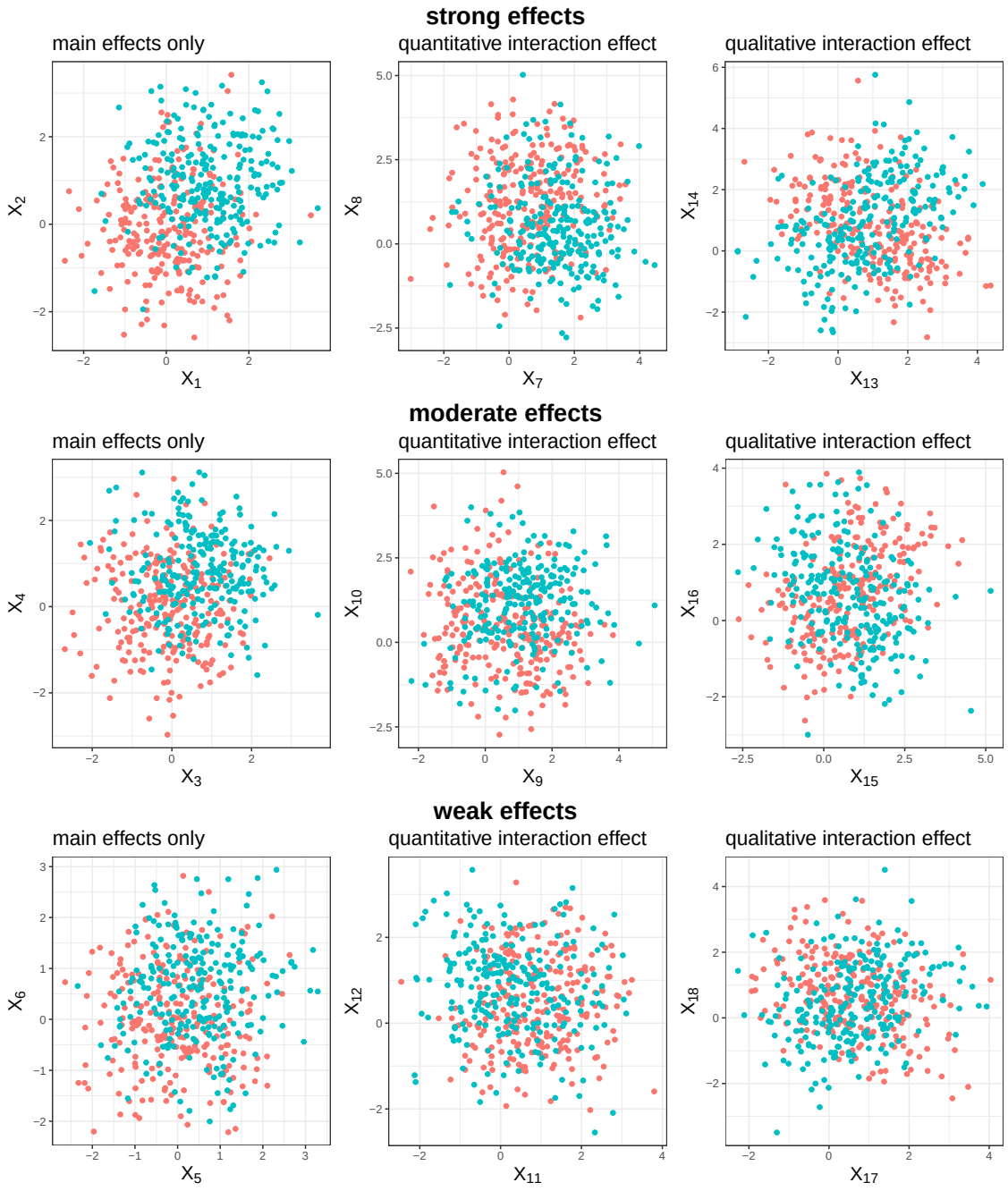

Fig. S43: Exemplary pairs of variables in a simulated data set (sample size: 500). Each point corresponds to an observation in the data set. The two colors distinguish the two outcome classes.

E.3 Detailed description of the simulation setting

For each simulated data set the first $n/2$ observations are from the first class and the second $n/2$ observations are from the second class. The distributions from which the values of the informative variables were drawn differ between the two classes.

The values of the uniformative variables $X_{19}, \ldots, X_{68}$ were drawn from the standard normal distribution $\mathcal{N}(0, 1)$.

The values of the variables $X_1, \ldots, X_6$ with only main effects, but no interaction effects were drawn from $\mathcal{N}(0, 1)$ for observations from the first class and from $\mathcal{N}(d, 1)$ for observations from the second class, where $d = 1$ for variables with strong effects, $d = 0.75$ for variables with moderate effects, and $d = 0.5$ for variables with weak effects (compare also Janitza *et al.* (2013)).

Be V the covariate space spanned by two continuous variables X_1^* and X_2^*, where X_1^* is on the x-axis and X_2^* is on the y-axis. In the following, when referring to V, the variable named first corresponds to X_1^* and the variable name second corresponds to X_2^*. The values of the variable pairs with qualitative interactions $\{X_{13}, X_{14}\}$, $\{X_{15}, X_{16}\}$, $\{X_{17}, X_{18}\}$ were drawn in such a way that the values from one class concentrate in the lower-left and upper-right corner of V and the values from the other class concentrate in the upper-left and lower-right corner of V. More precisely, the values of the variable pairs were drawn from the following mixtures of multivariate normal distributions:
For $\{X_{15}, X_{16}\}$ the values in the first class and for $\{X_{13}, X_{14}\}$ and $\{X_{17}, X_{18}\}$ the values in the second class were drawn from the following distribution:

$$\frac{1}{2} \cdot \mathcal{N}\left[\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\right] \;+\; \frac{1}{2} \cdot \mathcal{N}\left[\begin{pmatrix}a\\a\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\right] \tag{1}$$

For $\{X_{15}, X_{16}\}$ the values in the second class and for $\{X_{13}, X_{14}\}$ and $\{X_{17}, X_{18}\}$ the values in the first class were drawn from the following distribution:

$$\frac{1}{2} \cdot \mathcal{N}\left[\begin{pmatrix}a\\0\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\right] \;+\; \frac{1}{2} \cdot \mathcal{N}\left[\begin{pmatrix}0\\a\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\right] \tag{2}$$

As will be described in the following, the values of the parameter a for strong, moderate, and weak qualitative interaction effects were specified in such a way that the resulting variables featured a comparable predictive power as the variables with only main effects of the same effect strength levels. In a general context, the predictive importance of a variable is not only determined by how decisive the variable is on its own, but also by how predictive the variable is in conjunction with other variables. For example, two predictive variables, when considered together, may deliver a very accurate prediction rule, but when considered separately the two variables may not be nearly as predictive. Thus, the predictive importance of one of these variables depends on the presence of the other variable. If we want to measure the predictive importance of a variable, it is necessary to study how important this variable is in conjunction with other variables in the mean. The predictive importance of two variables can be measured in terms of the similarity or dissimilarity of the joint distributions of these variables in the two classes. The more similar these joint distributions are between the two classes, the less predictive power is featured in these variables. The similarity between two probability density functions $f_A(\boldsymbol{x})$ and $f_B(\boldsymbol{x})$ can be

measured using the overlapping index (Pastore and Calcagnì, 2019): $\int \min[f_A(\boldsymbol{x}), f_B(\boldsymbol{x})]d\boldsymbol{x} \in [0,1]$. We used this overlapping index for making the predictive powers of the variables with qualitative interaction effects comparable to that of the variables with only main effects of the same effect strength levels. This is realized by choosing values for parameter a that lead to degrees of overlap between the joint distributions of the interacting variables in the two classes that are comparable to the corresponding overlaps for pairs of variables for which one or both of them have a univariable effect. Denote $\theta \in \{strong, moderate, weak\}$ the effect strength, $\{X_{\theta,q,1}, X_{\theta,q,2}\}$ a variable pair with qualitative interaction effect of strength θ, $X_{\theta,u}$ a variable with univariable effect of strength θ, $X_{\theta,Un\backslash u,1}, \ldots, X_{\theta,Un\backslash u,5}$ all variables with univariable effect with the exception of $X_{\theta,u}$, and $X_{\theta,No,1}, \ldots, X_{\theta,No,50}$ all variables without effect (i.e., the variables $X_{19}, \ldots, X_{68}$ in Table 2 of the main paper). Moreover, let $O(X_{j_1}, X_{j_2})$ denote the overlapping index between the joint distributions of a variable pair $\{X_{j_1}, X_{j_2}\}$ in the two classes. Then for each effect strength level $\theta \in \{strong, moderate, weak\}$, the parameter a in formulas (1) and (2) was fixed to the value for which the following equation holds: $O(X_{\theta,q,1}, X_{\theta,q,2}) = \frac{1}{55}\sum_{X_j \in S_{.\backslash u}} O(X_{\theta,u}, X_j)$, where $S_{.\backslash u} = \{X_{\theta,Un\backslash u,1}, \ldots, X_{\theta,Un\backslash u,5}, X_{\theta,No,1}, \ldots, X_{\theta,No,50}\}$. The reason why we did not include the variables that feature interaction effects, $X_7, \ldots, X_{18}$, in these averages was simplicity: If we had included these variables, the parameter controlling the overlapping index for pairs of variables with quantitative interactions (see below) would have depended on the value of a and vice versa. This would have made it difficult to find the correct values of these parameters. We calculated each overlapping index value numerically with an exactness of three decimal places. The following a values resulted for strong, moderate, and weak effects: 1.772, 1.51, and 1.225.

The values of the variable pairs with quantitative interactions $\{X_7, X_8\}$, $\{X_9, X_{10}\}$, $\{X_{11}, X_{12}\}$ were drawn in such a way that the values from the second class concentrate in one corner of V and the values of the first class are distributed across the remaining three corners of V. For $\{X_7, X_8\}$ the values of the second class concentrate in the lower-right corner, for $\{X_9, X_{10}\}$ in the upper-right corner, and for $\{X_{11}, X_{12}\}$ in the upper-left corner. Below we will demonstrate how the values of $\{X_7, X_8\}$ were drawn. The values of $\{X_9, X_{10}\}$ and $\{X_{11}, X_{12}\}$ were drawn analogously with the difference that for these the values in the second class concentrate in a different corner of V. The values of $\{X_7, X_8\}$ in the first class were drawn from the following distribution:

$$\frac{1}{3}\cdot\mathcal{N}\left[\begin{pmatrix}0\\a\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right] + \frac{1}{3}\cdot\mathcal{N}\left[\begin{pmatrix}a\\a\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right] + \frac{1}{3}\cdot\mathcal{N}\left[\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right] \tag{3}$$

The values of $\{X_7, X_8\}$ in the second class were drawn from the following distribution:

$$\begin{aligned}&\pi_a\cdot\mathcal{N}\left[\begin{pmatrix}a\\0\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right] + \frac{1-\pi_a}{3}\cdot\mathcal{N}\left[\begin{pmatrix}0\\a\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right] + \\ &\frac{1-\pi_a}{3}\cdot\mathcal{N}\left[\begin{pmatrix}a\\a\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right] + \frac{1-\pi_a}{3}\cdot\mathcal{N}\left[\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right]\end{aligned} \tag{4}$$

Here, a is fixed to the same value 1.772 that was also used in the case of $\{X_{13}, X_{14}\}$, that is, the variable pair with strong qualitative interaction effect. The larger the value of π_a is, the greater is the concentration of the second class in the lower-right corner. We fixed π_a to that value for which

the overlapping index between the two class-specific joint distributions of $\{X_7, X_8\}$ took the same value as that for $\{X_{13}, X_{14}\}$. An obvious alternative to this proceeding would have been to fix π_a to the value one and alter the value of a for achieving the desired overlapping index instead of diminishing the concentration of observations in the lower-right corner until the desired value of the overlapping index was achieved. However, this procedure would have led to a too small a value, because for quantitative interaction effects the variable value pairs in the second class vary less than in the case of qualitative interaction effects. As a consequence, the means of the normal mixture components in equations (3) and (4) associated with the same overlapping index value as in the case of the strong qualitative interaction effect would have been close to each other. The following π_a values resulted for strong, moderate, and weak effects: 0.649, 0.574, and 0.485. The same a values (1.772, 1.51, and 1.225) as in the case of the qualitative interaction effects were used for the three effect strength levels.

F Ranks the variables and variable pairs obtained for the individual data sets using the different methods

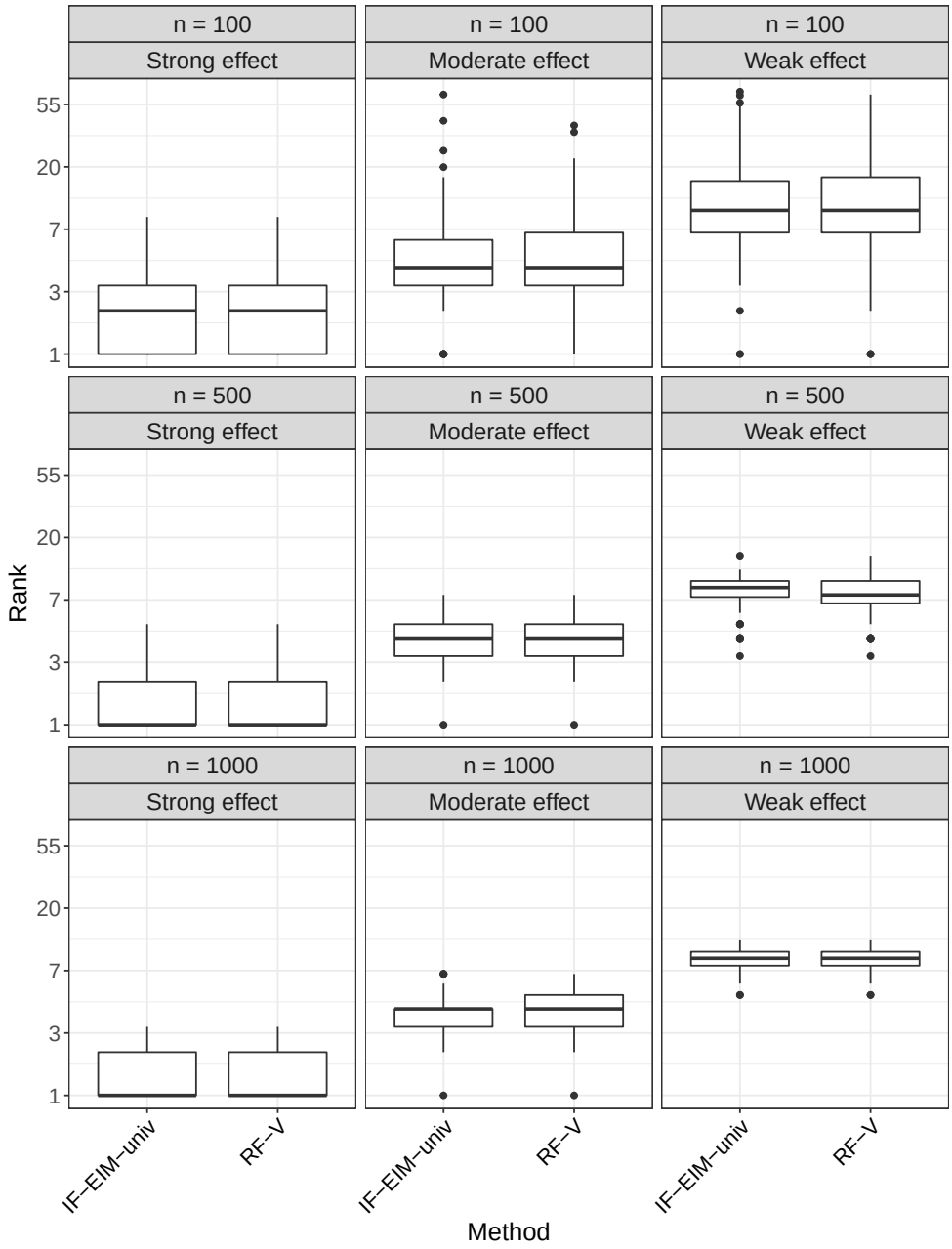

Fig. S44: Simulation results – univariable effects. The boxplots show the ranks the respective variables obtained using each simulated data set. Note that for the variables with main effects only, each effect strength was represented by two variables in the simulation design. The boxplots show the pooled ranks obtained for both variables of each effect strength. The y-axes are shown on log scale.

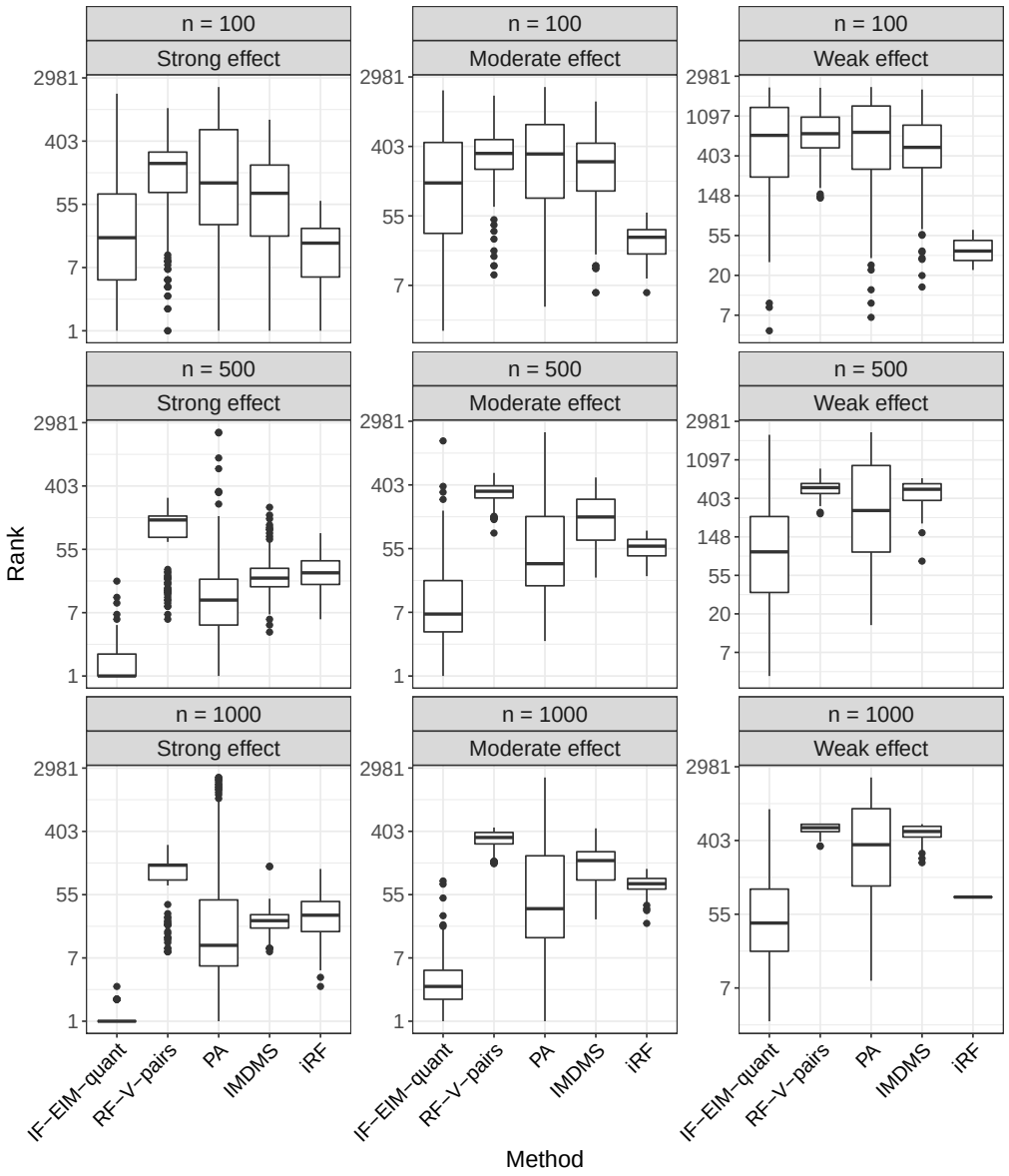

Fig. S45: Simulation results – quantitative interaction effects. The boxplots show the ranks the respective variable pairs obtained using each simulated data set. The y-axes are shown on log scale.

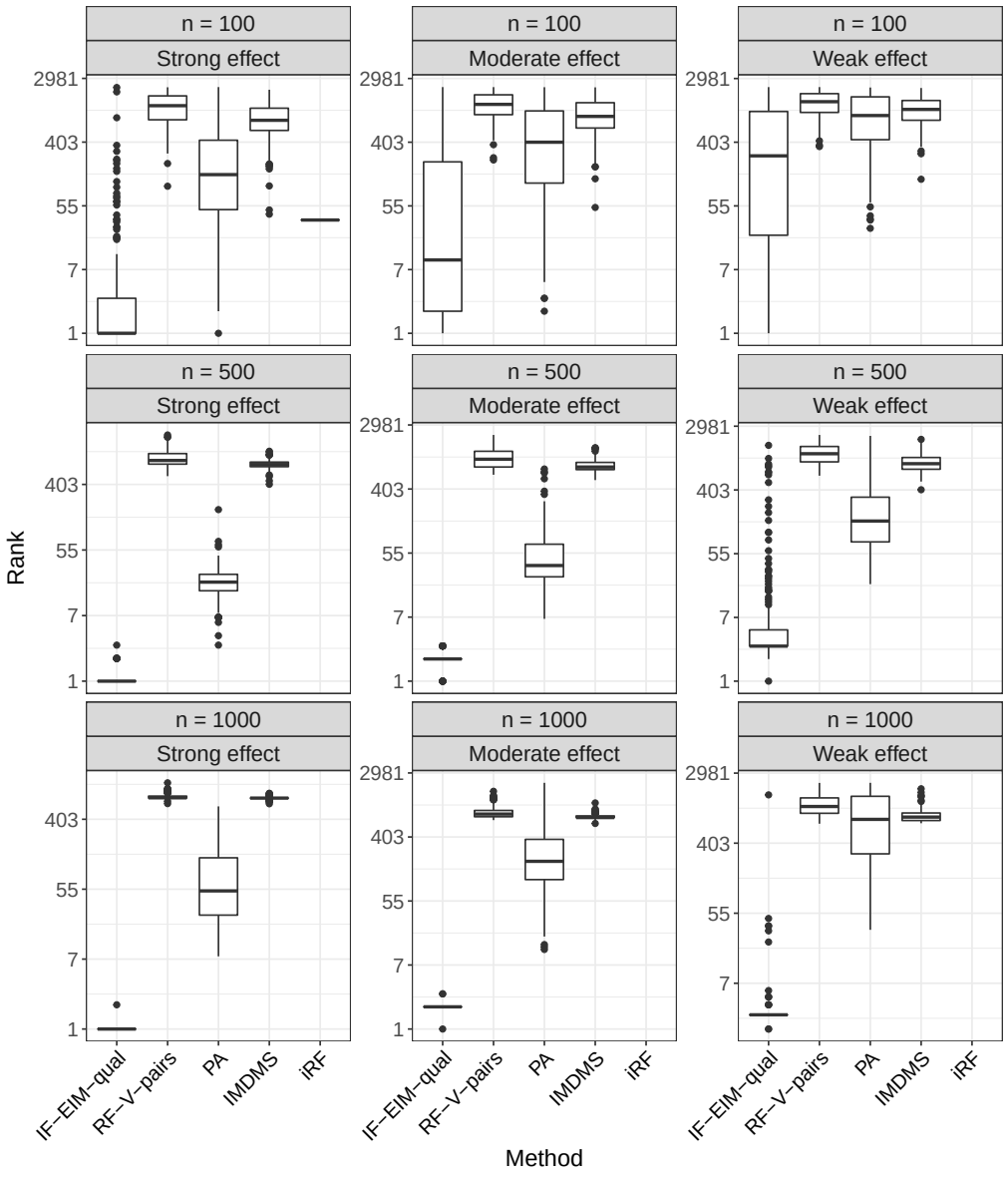

Fig. S46: Simulation results – qualitative interaction effects. The boxplots show the ranks the respective variable pairs obtained using each simulated data set. The y-axes are shown on log scale.

G Median ranks variable pairs with main effects, but without interaction effects, obtained using the different methods

Effect:	Strong	Moderate	Weak
	n = 100		
IF-EIM-qual	992.5 [591.0, 1414.2]	1032.5 [606.2, 1460.5]	1004.5 [571.2, 1589.0]
IF-EIM-quant	28.0 [6.0, 114.8]	93.5 [20.0, 329.8]	338.0 [154.8, 1028.2]
RF-V-pairs	2.0 [1.0, 6.0]	93.0 [15.8, 163.2]	336.5 [222.5, 475.5]
PA	5.0 [2.0, 33.8]	56.0 [13.8, 246.2]	497.5 [136.8, 1433.5]
IMDMS	2.0 [1.0, 6.0]	30.5 [11.0, 79.0]	297.5 [140.5, 491.2]
iRF	3.0 [1.0, 6.0] (99%)	15.0 [7.0, 24.0] (72%)	26.0 [16.5, 36.0] (14%)
	n = 500		
IF-EIM-qual	741.0 [382.0, 1117.5]	816.0 [411.5, 1350.8]	951.0 [487.5, 1443.2]
IF-EIM-quant	23.5 [6.8, 124.0]	52.0 [11.0, 211.2]	192.0 [65.0, 465.5]
RF-V-pairs	1.0 [1.0, 1.0]	77.0 [16.0, 136.0]	343.0 [271.0, 399.8]
PA	1.0 [1.0, 2.0]	15.5 [9.0, 30.2]	195.5 [47.8, 714.2]
IMDMS	1.0 [1.0, 1.0]	13.0 [9.0, 18.0]	233.0 [143.8, 316.2]
iRF	1.0 [1.0, 2.0] (100%)	17.0 [11.0, 26.0] (98%)	52.0 [45.0, 63.5] (18%)
	n = 1000		
IF-EIM-qual	665.0 [372.8, 1031.0]	739.5 [345.0, 1134.2]	788.5 [328.2, 1328.0]
IF-EIM-quant	17.0 [5.0, 72.0]	31.0 [12.0, 140.0]	111.0 [29.8, 346.8]
RF-V-pairs	1.0 [1.0, 1.0]	79.0 [18.0, 136.0]	339.0 [282.0, 395.0]
PA	1.0 [1.0, 2.0]	19.5 [8.0, 83.5]	201.0 [64.5, 553.8]
IMDMS	1.0 [1.0, 1.0]	13.0 [10.0, 18.0]	203.0 [155.8, 282.0]
iRF	1.0 [1.0, 2.0] (100%)	11.0 [7.0, 17.0] (100%)	78.5 [63.0, 92.8] (27%)

Table S1: Simulation results – median ranks obtained for variable pairs with main effects, but without interaction effects. We considered the pair with the two variables that both have strong effects ("Strong"), the pair with the the two variables that both have moderate effects ("Moderate"), and the pair with the two variables that both have weak effects ("Weak"). The numbers show the median ranks the respective variable pairs obtained across the simulated data sets. The numbers in square brackets show the 25% quantiles and 75% quantiles of the ranks obtained for the simulated data sets. In the case of iRF, the percentages of the simulated data sets for which the respective pairs were selected using iRF are given in addition.

References

Amasyah, M. and Ersoy, O. (2008). Cline: A new decision-tree family. *IEEE Transactions on Neural Networks*, **19**(2), 356–363.

Basu, S., Kumbier, K., Brown, J. B., and Yu, B. (2018). Iterative random forests to discover predictive and stable high-order interactions. *Proceedings of the National Academy of Sciences (PNAS)*, **115**(8), 1943–1948.

Bertsimas, D. and Dunn, J. (2017). Optimal classification trees. *Machine Learning*, **106**, 1039–1082.

Breiman, L. (2001). Random forests. *Machine Learning*, **45**(1), 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Ston, C. J. (1984). *Classification and Regression Trees*. Wadsworth International Group, Monterey, CA.

Brodley, C. E. and Utgoff, P. E. (1995). Multivariate decision trees. *Machine Learning*, **19**, 45–77.

Bureau, A., Dupuis, J., Falls, K., Lunetta, K. L., Hayward, B., Keith, T. P., and Eerdewegh, P. V. (2005). Identifying SNPs predictive of phenotype using random forests. *Genetic Epidemiology*, **28**, 171–182.

Casalicchio, G., Bossek, J., Lang, M., Kirchhoff, D., Kerschke, P., Hofner, B., Seibold, H., Vanschoren, J., and Bischl, B. (2017). OpenML: An R package to connect to the machine learning platform OpenML. *Computational Statistics*, **32**(3), 1–15.

Chen, Z. and Zhang, W. (2013). Integrative analysis using module-guided random forests reveals correlated genetic factors related to mouse weight. *PLoS Computational Biology*, **9**(3), e1002956.

Couronné, R., Probst, P., and Boulesteix, A.-L. (2018). Random forest versus logistic regression: a large-scale benchmark experiment. *BMC Bioinformatics*, **19**, 270.

Dazard, J.-E., Ishwaran, H., Mehlotra, R., Weinberg, A., and Zimmerman, P. (2018). Ensemble survival tree models to reveal pairwise interactions of variables with time-to-events outcomes in low-dimensional setting. *Statistical Applications in Genetics and Molecular Biology*, **17**(1), 20170038.

Du, J. and Linero, A. (2019). Interaction Detection with Bayesian Decision Tree Ensembles. In K. Chaudhuri and M. Sugiyama, editors, *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 108–117.

Gama, J. and Brazdil, P. (1999). Linear tree. *Intelligent Data Analysis*, **3**(1), 1–22.

Gashler, M., Giraud-Carrier, C., and Martinez, T. (2008). Decision tree ensemble: Small heterogeneous is better than large homogeneous. In M. A. Wani, X.-W. Chen, D. Casasent, L. A. Kurgan, T. Hu, and K. Hafeez, editors, *Seventh International Conference on Machine Learning and Applications*, pages 900–905.

Gheyas, I. A. and Smith, L. S. (2010). Feature subset selection in large dimensionality domains. *Pattern Recognition*, **43**(1), 5–13.

Gupta, R., Smolka, S. A., and Bhaskar, S. (1994). On randomization in sequential and distributed algorithms. *ACM Computing Surveys*, **26**(1), 7–86.

Hornung, R. (2020). Diversity forests: Using split sampling to allow for complex split procedures in random forest. Technical report 234, Department of Statistics, University of Munich.

Hornung, R. and Wright, M. N. (2019). Block forests: random forests for blocks of clinical and omics covariate dat. *BMC Bioinformatics*, **20**, 358.

Ishwaran, H. (2007). Variable importance in binary regression trees and forests. *Electronic Journal of Statistics*, **1**, 519–537.

Ishwaran, H. and Kogalur, U. B. (2020). *randomForestSRC: Fast Unified Random Forests for Survival, Regression, and Classification (RF-SRC)*. R package version 2.9.3.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., and Lauer, M. S. (2008). Random survival forests. *Ann Appl Stat*, **2**(3), 841–860.

Ishwaran, H., Kogalur, U. B., Gorodeski, E. Z., Minn, A. J., and Lauer, M. S. (2010). High-dimensional variable selection for survival data. *Journal of the American Statistical Association*, **105**(489), 205–217.

Janitza, S., Strobl, C., and Boulesteix, A.-L. (2013). An AUC-based permutation variable importance measure for random forests. *BMC Bioinformatics*, **14**, 119.

Jiang, R., Tang, W., Wu, X., and Fu, W. (2009). A random forest approach to the detection of epistatic interactions in case-control studies. *BMC Bioinformatics*, **10**(Suppl. 1), S65.

Kelly, C. and Okada, K. (2012). Variable interaction measures with random forest classifiers. In *Proceedings of the 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 154–157.

Kolakowska, A. and Malina, W. (2005). Fisher sequential classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **35**(5), 988–998.

Ley, T. J., Miller, C., Ding, L., Raphael, B. J., Mungall, A. J., Robertson, A., *et al.* (2013). Genomic and epigenomic landscapes of adult de novo acute myeloid leukemia. *N Engl J Med*, **368**, 2059–2074.

Li, J., Malley, J. D., Andrew, A. S., Karagas, M. R., and Moore, J. H. (2016). Detecting gene-gene interactions using a permutation-based random forest method. *BioData Mining*, **9**, 14.

Li, X.-B., Sweigart, J. R., Teng, J. T., Donohue, J. M., Thombs, L. A., and Wang, S. M. (2003). Multivariate decision trees using linear discriminants and tabu search. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, **33**(2), 194–205.

Loh, W.-Y. and Shih, Y.-S. (1997). Split selection methods for classification trees. *Statistica Sinica*, **7**(4), 815–840.

López-Chau, A., Cervantes, J., López-García, L., and Lamont, F. G. (2013). Fisher's decision tree. *Expert Systems with Applications*, **40**(16), 6283–6291.

Manwani, N. and Sastry, P. (2012). Geometric decision tree. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **42**(1), 181–192.

Menze, B. H., Kelm, B. M., Splitthoff, D. N., Koethe, U., and Hamprecht, F. A. (2011). On oblique random forests. In D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, editors, *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 453–469.

Murthy, S. K., Kasif, S., and Salzberg, S. (1994). A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, **2**, 1–32.

Ng, V. W. and Breiman, L. (2005). Bivariate variable selection for classification problem. Technical report 692, Department of Statistics, University of California, Berkeley, CA.

Pastore, M. and Calcagnì, A. (2019). Measuring distribution similarities between samples: a distribution-free overlapping index. *Frontiers in Psychology*, **10**, 1089.

Probst, P., Boulesteix, A.-L., and Bischl, B. (2019). Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*, **20**(53), 1–32.

Rainforth, T. and Wood, F. (2015). Canonical correlation forests. arXiv:1507.05444.

Robertson, B., Price, C., and Reale, M. (2013). CARTopt: a random search method for nonsmooth unconstrained optimization. *Computational Optimization and Applications*, **56**(2), 291–315.

Rodríguez, J. J., Kuncheva, L. I., and Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(10), 1619–1630.

Sethi, I. K. and Yoo, J. H. (1994). Design of multicategory multifeature split decision trees using perceptron learning. *Pattern Recognition*, **27**(7), 939–947.

Sorokina, D., Caruana, R., and Riedewald, M. (2007). Additive groves of regression trees. In J. N. Kok, J. Koronacki, R. L. Mantaras, S. M. S, D. Mladenič, and A. Skowron, editors, *Proceedings of the 18th European conference on Machine Learning*, pages 323–334.

Sorokina, D., Caruana, R., Riedewald, M., and Fink, D. (2008). Detecting statistical interactions with additive groves of trees. In W. Cohen, A. K. McCallum, and S. T. Roweis, editors, *Proceedings of the 25th international conference on Machine learning*, pages 1000–1007.

Utgoff, P. E. and Brodley, C. E. (1990). An incremental method for finding multivariate splits for decision trees. In B. Porter and R. Mooney, editors, *Proceedings of the Sevent International Conference on Machine Learning*, pages 58–65.

Vanschoren, J., van Rijn, J. N., Bischl, B., and Torgo, L. (2013). OpenML: Networked Science in Machine Learning. *SIGKDD Explorations*, **15**(2), 49–60.

Wickramarachchi, D. C., Robertson, B. L., Reale, M., Price, C. J., and Brown, J. (2015). HHCART: An oblique decision tree. *Computational Statistics and Data Analysis*, **96**, 12–23.

Wright, M. N. and König, I. R. (2019). Splitting on categorical predictors in random forests. *PeerJ*, **7**, e6339.

Yıldız, O. T. and Alpaydın, E. (2001). Omnivariate decision trees. *IEEE Transactions on Neural Networks*, **12**(6), 1539–1546.

Yoshida, M. and Koike, A. (2011). SNPInterForest: A new method for detecting epistatic interactions. *BMC Bioinformatics*, **12**, 469.

Zhou, M., Dai, M., Yao, Y., Liu, J., Yang, C., and Peng, H. (2019). BOLT-SSI: A statistical approach to screening interaction effects for ultra-high dimensional data. arXiv:1902.03525.