LMU

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR STATISTIK

Roman Hornung, Marvin N. Wright

# Block Forests: random forests for blocks of clinical and omics covariate data

Technical Report Number 219, 2018
Department of Statistics
University of Munich

http://www.statistik.uni-muenchen.de

# Block Forests: random forests for blocks of clinical and omics covariate data

Roman Hornung[1]* Marvin N. Wright[2,3]

December 20, 2018

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

[2] Leibniz Institute for Prevention Research and Epidemiology - BIPS, Bremen, 28359, Germany

[3] Section of Biostatistics, Department of Public Health, University of Copenhagen, Copenhagen, 1014, Denmark

**Abstract**

In the last years more and more multi-omics data are becoming available, that is, data featuring measurements of several types of omics data for each patient. While using multi-omics data as covariate data in outcome prediction is promising, it is also challenging due to the complex structure of such data. Random forest is a prediction method known for its ability to render complex dependency patterns between the outcome and the covariates. Against this background we developed five candidate random forest variants tailored to multi-omics covariate data. These variants modify the split point selection of random forest to incorporate the block structure of multi-omics data and can be applied to any outcome type for which a random forest variant exists, such as categorical, continuous and survival outcomes. Using 20 multi-omics data sets with survival outcome we compared the prediction performances of the block forest variants, using random survival forest as a reference method. We also considered the common special case of having clinical covariates and measurements of a single omics data type available.

We identify one variant termed "block forest" that performed significantly better than standard random survival forest (adjusted $p$-value: 0.027). The two best performing variants have in common that the block choice is randomized in the split point selection procedure. In the case of having clinical covariates and a single omics data type available, the improvements of the variants over random survival forest were larger than in the case of the multi-omics data. In the former case four of the five variants performed significantly better than random survival forest. The degrees of improvements over random survival forest varied strongly across data sets.

The new prediction method block forest for multi-omics data can significantly improve the prediction performance of random forest. Block forest is particularly effective for the special case of using clinical covariates in combination with measurements of a single omics data type.

*Corresponding author. Email: hornung@ibe.med.uni-muenchen.de.

# 1 Background

In the last decade the measurement of various types of omics data, such as gene expression, methylation or copy number variation data has become increasingly fast and cost-effective. Therefore, there exist more and more patient data for which several types of omics data are available for the same patients. In the following, such data are denoted as multi-omics data and the different subsets of this data containing the individual data types are referred to as "blocks". Using multi-omics data in prediction modeling is promising because each type of omics data may contribute information valuable for the prediction of phenotypic outcomes. However, combining different types of omics data effectively is challenging for several reasons. First, the predictive information contained in the individual blocks is overlapping. Second, the levels of predictive information differ between the blocks and depend on the particular outcome considered [1]. Third, there exist interactions between variables across the different blocks, which should be taken into account [2].

While pioneering work in the area of prediction modelling using multi-omics covariate data was already published as early as 2004 [3], further methodological developments in this area do not seem to have been pursued until the last several years. This long-lasting lack of prediction methods tailored to multi-omics covariate data was probably due to the fact that multi-omics data had not been available on a larger scale until recently. Simon et al. [4] presented the sparse group lasso in 2013, a prediction method for grouped covariate data that automatically removes non-informative covariate groups and performs lasso-type variable selection for the remaining covariate groups. A disadvantage of the sparse lasso in applications to multi-omics data is that it does not explicitly take the different levels of predictive information of the blocks into account. This is different for the IPF-LASSO [5], a lasso-type regression method for multi-omics data in which each block is associated with an individual penalty parameter. Vazquez et al. [6] model the relationship between phenotypic outcomes and multi-omics covariate data fully Bayesian using a Bayesian generalized additive model. Mankoo et al. [7] consider a two-step approach: In the first step they aim to remove redundancies between the different blocks by filtering out highly correlated pairs of variables from different blocks and in the second step they apply standard $L_1$ regularized Cox regression [8] using the remaining variables. Seoane et al. [9] use multiple kernel learning methods, considering composite kernels as linear combinations of base kernels derived from each each block, where they incorporate pathway information in the selection of relevant variables. Similarly, Fuchs et al. [10] consider combining classifiers, each learned using one of the blocks. In the context of a comparison study, Boulesteix et al. [5] again consider an approach based on combining prediction rules, each learned using a single block: first, lasso is fitted to each block and, second, the resulting linear predictors are used as covariates in a low-dimensional regression model. In addition to the approach mentioned above, Fuchs et al. [10] also consider performing variable selection separately for each block and then learning a single classifier using all blocks. Klau et al. [11] present the priority-Lasso, a lasso-type prediction method for multi-omics data that differs from the approaches described above in that its main focus is not prediction accuracy but applicability from a practical point of view: With this method the user has to provide a priority order of the blocks that is for example motivated by the costs of generating each type of data. Blocks of low priority are likely to be automatically excluded by this method, which should frequently lead to prediction rules that are easy to apply in practice and, at the same time, feature a high prediction accuracy. A related method is the TANDEM approach [12], which attributes a lower priority to gene expressions than to the other omics data types in order to avoid the prediction rule to be strongly dominated by the gene expressions. For a recent overview of approaches for analyzing multi-omics data focused on data mining see Huang et al. [2].

Apart from multi-omics data, in most cases where a certain type of omics data is available, the corresponding phenotypic data set features several clinical covariates. The latter are often of great prognostic relevance and should be prioritized over or at least be used in addition to the omics data. Many of the methods described above can be used for such data as well, if the clinical data is treated as an omics data type from a methodological point of view. However, since this problem was known before the rise of multi-omics data, there also exist various strategies for effectively using the clinical information in combination with a single omics data type. See Boulesteix & Sauerbrei [13] for a detailed discussion of such approaches and De Bin et al. [14] for a comparison study illustrating their application.

The random forest algorithm is a powerful prediction method that is known to be able to capture complex dependency patterns between the outcome and the covariates. The latter feature makes random forest a promising candidate for developing a prediction method tailored to the challenges of multi-omics data. In this paper we set out to develop such a variant, where we initially consider five different candidate methods. Each of these five considered random forest variants differ from conventional random forests merely with respect to the selection of the split points in the decision trees constituting the forests. Therefore, most other components of a random forest are unchanged and each of these five variants can be applied to any outcome type, for which there exists a random forest variant, e.g., categorial, continuous and survival outcomes. We compared the prediction performances of the five variants with each other and with random survival forest (RSF) [15] on 20 real multi-omics data sets with survival outcome. RSF is known to be a strong prediction method for survival outcomes, see for example Bou-Hamad et al. [16] and Yosefian et al. [17] and the references therein. In this comparison study we identified one particularly well performing variant that performed best, both when considering all blocks and for the special case of having only clinical information and a single omics data type. This variant is denoted as the block forest algorithm in the following. We implemented all five variants for categorical, continuous and survival outcomes in our R package `blockForest`available from CRAN, where block forest is used by default. The other variants should be considered with caution only as these may deliver worse prediction results.

The paper is structured as follows. In the next session, after briefly describing the multi-omics data format and the splitting procedure performed by standard random forest, we detail each of the five considered random forest variants for multi-omics data. Here, we also discuss the rationale behind each procedure. Subsequently, we describe the design of the comparison study. The results of the comparison study are presented and described in Section 3. Finally, we summarize our main results and draw specific conclusions from the results of the comparison study in Section 4.

# 2 Methods

## 2.1 Multi-omics data format

A multi-omics data set with $n$ observations consists of $M$ covariate matrices $\boldsymbol{X}_1, \dots, \boldsymbol{X}_M$ and an outcome vector $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n$. The $m$th matrix $\boldsymbol{X}_m$ of dimension $n \times p_m$ contains, for each observation, the measurements of the $p_m$ variables in the $m$th block. The outcome values $\boldsymbol{y}_i$, $i = 1, \dots, n$, are most often scalars, for example $\boldsymbol{y}_i \in \{0, 1\}$ for binary outcomes or $\boldsymbol{y}_i \in \mathbb{R}$ for metric outcomes. They may, however, also take the form of vectors, for example $\boldsymbol{y}_i = \{y_{i,1}, y_{i,2}\}$ with $y_{i,1} \in \mathbb{R}_{>0}$ and $y_{i,2} \in \{0, 1\}$ for survival outcomes, where $y_{i,1}$ denotes the survival/censoring time of the $i$th observation and $y_{i,2}$ its value of the censoring indicator. The covariate matrices can be concatenated in order to form a single covariate matrix $\boldsymbol{X} := [\boldsymbol{X}_1, \dots, \boldsymbol{X}_M]$ with $p = \sum_{m=1}^{M} p_m$ columns.

## 2.2 Split selection procedures for random forests tailored to multi-omics data

As described in Section 1, we propose and study five random forest variants for multi-omics data, which differ merely with respect to split point selection. In the following, we first recall the standard random forest algorithm and the split point selection performed by this algorithm. Subsequently, we describe the split point selection procedures of the considered variants and briefly discuss the motivations behind each of these approaches. Each of these procedures involves block-specific parameters, which are chosen automatically using an optimization procedure described in Section 2.3.

### 2.2.1 Standard random forest

A random forest prediction rule is a collection of decision trees, where each of the latter is constructed using a subsample or bootstrap sample of a training data set. Each of the tree decision rules performs a series of binary decisions, where each decision is obtained using a threshold, called “split point”, in the values of one of the covariate variables available in the training data set. The decision trees are constructed by recursively dividing the available samples in two subgroups using the split points which are obtained during the construction of the trees. The nested subgroups are denoted as nodes.

In standard random forest a split point for a node in a tree is obtained as follows (assuming only continuous variables for ease of presentation). First, a number $mtry$ of variables is randomly sampled from all variables. Second, an optimal split point in the ordered values of the sampled variables is obtained in the following way: 1) Divide the node once according to each possible split point in each variable and for every division obtained, calculate the value of a certain quantitative split criterion; 2) Use that split point among all split points considered in 1) that was best according to the split point criterion. The block structure of multi-omics data is obviously not taken into account in this split point selection procedure. Note that the split point selection is performed slightly differently for nominal variables, for details see Hastie et al. [18] (chapter 9.2.4).

### 2.2.2 VarProb: Block-specific variable selection probabilities

The procedure of drawing from all variables without taking the block structure into account, as performed in the split point selection of standard random forest, has two main issues. First, with this procedure, blocks that involve fewer variables are underrepresented in the drawn variables. In many cases, it would be preferable if a variable from a block with fewer variables would be drawn more often than a variable from a block with many variables. This is because the predictive information in blocks with fewer variables tends to be more dense. For example, the block containing clinical information does usually contain a very small number of variables in comparison to the omics blocks, but a large fraction of this small number of variables can be expected to be highly predictive. The second, related issue is that even for equal block sizes, the procedure of drawing variables from different blocks with the same frequencies does not take into account that the blocks differ with respect to their levels of predictive information contained. It could be an advantage if variables from blocks with much predictive information would be drawn more often than variables from blocks with little predictive information.

Both of the above issues are addressed by using block-specific variable selection probabilities in the variant VarProb presented in this subsection.

This has the effect that the sampling probabilities of variables from some blocks are higher than those of variables from other blocks. With the split selection procedure of VarProb, a variable in block $m$ has sampling probability $v_m$, where $\sum_{m=1}^{M} p_m v_m = 1$. In order to fix the number of variables to be drawn for each split to *mtry* we proceed as follows: Each sampled variable is drawn one after another. If, in this process, a variable is drawn that is already in the set of drawn variables, the drawing is repeated until a variable is drawn that is not yet in the set of drawn variables. This process is repeated until *mtry* variables have been drawn. Finally, the best split point is determined in the *mtry* drawn variables as in standard random forest. The value of *mtry* is set to $\sum_{m=1}^{M} \sqrt{p_m}$.

### 2.2.3 SplitWeights: Block-specific weights of split criterion values

Using block-specific variable selection probabilities in VarProb has the effect of prioritizing some blocks over others. Another way to accomplish the latter is to use block-specific weights $w_m$ $(m = 1, \dots, M)$ for the split criterion values, where $w_1, \dots, w_m > 0$ and $\max\{w_1, \dots, w_M\} = 1$ for reasons of identifiability. With this procedure, variables from blocks with high $w_m$ values are prioritized over variables from blocks with low $w_m$ values. First, a number of $mtry = \sum_{m=1}^{M} \sqrt{p_m}$ variables are drawn from all variables. Second, the split criterion values associated with all split points in the sampled variables are calculated and these values are weighted using the block-specific weights $w_m$. Third, the split point is chosen that features the highest weighted split criterion value.

As in the case of VarProb, with SplitWeights variables from different blocks are prioritized differently by the procedure. Because the predictive information contained in blocks with large numbers of variables tends to be less dense, more variables need to be sampled from these blocks in order to increase the likelihood of obtaining reasonably informative variables. However, when sampling different numbers of variables from different blocks in this way, the best variable from a large block tends to separate the observations better than the best variable from a small block just by chance, even though the best variable from the small block is often actually more suitable. This problem occurs for the same reason as the bias towards variables with many possible split points described by Strobl et al. [19]. The tendency of selecting suboptimal variables from large blocks is avoided when the split criterion values of these blocks are attributed smaller weights $w_m$.

#### 2.2.4 BlockVarSel: Separate sampling of variables from each block and block-specific weights of split criterion values

A disadvantage of SplitWeights is that the smaller the number of variables in a block is, the smaller is the probability that this block is present in the variables sampled for a split. For example, a block containing clinical information most often contains only few variables, which is why there will be no clinical covariates among the sampled variables for most of the splits. Clinical covariates, however, often contain much predictive information and interact with omics variables, which is why it is detrimental if they are considered only infrequently.

In order to avoid the above shortcoming of SplitWeights, with the split selection procedure BlockVarSel presented in this subsection, for each split, we sample fixed numbers of variables from each block separately. More precisely, for $m = 1, \dots, M$ we sample $\sqrt{p_m}$ variables from block $m$. Subsequently, split point selection is performed as with SplitWeights, that is, using weighted split criterion values with block-specific weights $w_m$. Note that with this approach, as with SplitWeights, larger numbers of variables are sampled from larger blocks.

#### 2.2.5 RandomBlock: Random block selection

A key reason for the strong prediction performance of standard random forest is that through considering only a random subset of *mtry* variables for each split, the resulting trees are very dissimilar from each other. Roughly formulated, each tree captures different aspects of the complex dependency structure between the outcome and the variables. In the context of multi-omics data, we are particularly interested in rendering aspects of the interplay of the different blocks with respect to their influence on the outcome. Therefore, it seems beneficial to make the trees not only very dissimilar with respect to the involved variables in general, but also in particular with respect to the involvements of the different blocks.

On the basis of this idea, with the split selection procedure RandomBlock, first, one of the blocks is selected randomly and, second, a subset of variables from the selected block is sampled. The sampled subset of variables from the selected block is subsequently considered for splitting and split point selection is performed among the sampled subset of variables as in standard random forest. In order to account for the different levels of predictive information associated with the different blocks, block-specific selection probabilities $b_m$ are used, where $\sum_{m=1}^{M} b_m = 1$. After block selection, $\sqrt{p_m}$ variables are sampled from the selected block.

The fact that the succession of the blocks used for splitting is random within the trees makes the resulting forest put high emphasis on rendering interactions between variables across blocks. A further distinctive property of this split selection procedure is that there are no comparisons of split points made across blocks, since only variables from a single block are considered for each split. This avoids issues with differences in dimensionality between the blocks that have to be dealt with in the other approaches. Moreover, it avoids problems caused by correlations between variables from different blocks, where these correlations are associated with the fact that the predictive information contained in different blocks is overlapping. A further advantage of RandomBlock is that the $b_m$ values can give indications of the relative importances of the different blocks for prediction. The higher the $b_m$ value of a block is, the higher its importance compared to the other blocks will tend to be. However, small $b_m$ must be interpreted with great care, because important blocks can be attributed small $b_m$ values if these blocks share much predictive information with other important blocks. For details on this mechanism see Section E of Supplementary File 1 referenced in Section 3.1. The $b_m$ values may also be used to screen out blocks that are not relevant for prediction given the remaining blocks. This can be done by excluding blocks that feature very small $b_m$ values.

#### 2.2.6 BlockForest: Block sampling with separate sampling of variables from each block and block-specific weights of split criterion values

A major advantage of the procedure RandomBlock that was described in the previous subsection is that it adds the additional randomization component "block selection" to the standard random forest algorithm. For the procedure described in the current subsection we extended the procedure BlockVarSel by a block selection randomization procedure. The new procedure is performed as follows: 1) Obtain a subset of all $M$ blocks by selecting each block with probability 0.5, that is, all blocks are selected with probability $0.5^M$ and no block at all with the same probability; 2) If no block was selected, repeat 1) until at least one block is selected; 3) Perform the split selection procedure of BlockVarSel using only the blocks selected in 1) or 2), respectively. This new procedure leads to more strongly differing trees than BlockVarSel, because weaker blocks are better taken into account. We use the general term "BlockForest" or "block forest" for this procedure, because it performed best among the studied procedures in our comparison study.

### 2.3 Optimization of tuning parameters

Each of the split point selection procedures described above contains $M$ tuning parameters, where each tuning parameter is associated with one of the blocks. These tuning parameters are variable selection probabilities $v_1, \dots, v_M$ for VarProb, block weights $w_1, \dots, w_M$ for SplitWeights, BlockVarSel and BlockForest, as well as block selection probabilities $b_1, \dots, b_M$

for RandomBlock.

The optimization of the tuning parameter values is performed as follows for each of the five variants:

1. For $it = 1, \ldots, N_{\text{sets}}$:
   (a) Generate a random set $\boldsymbol{\mathcal{S}}_{it}$ of $M$ tuning parameter values.
   (b) Construct a forest with $num.trees_{\text{pre}}$ trees using the tuning parameter value set $\boldsymbol{\mathcal{S}}_{it}$ and record the out-of-bag prediction error of that forest.
2. Use that tuning parameter set out of $\boldsymbol{\mathcal{S}}_1, \ldots, \boldsymbol{\mathcal{S}}_{N_{\text{sets}}}$, for which the corresponding forest delivered the smallest out-of-bag prediction error.

The generation of the random sets of tuning parameter values is described in Section A of Supplementary File 1. The choice of the prediction error measure depends on the considered outcome. For survival data (see the comparison study presented in Section 2.4), we use one minus the value of Harrell's C index.

Note that while the optimization algorithm presented above is relatively inefficient, it has the important advantage that it is consistent with respect to the tuning parameter value set actually associated with the lowest out-of-bag prediction error. That is, for larger values of $N_{\text{sets}}$, the optimized tuning parameter set will approximate the optimal tuning parameter value set, that is, the set with lowest possible out-of-bag prediction error, increasingly well. By contrast, the properties of more sophisticated procedures tend to be less clear, which is why such procedures can be prone to result in local optima. In our analyses, we use the values $N_{\text{sets}} = 300$ and $num.trees_{\text{pre}} = 1500$, which are also the default values in our R package `blockForest`. Such large numbers of sets and trees per constructed forest are possible because the package `blockForest` is a fork of `ranger` [20], which is a fast C++ random forest implementation.

## 2.4 Comparison study

### 2.4.1 Data

We used 21 real multi-omics data sets with survival outcome, where each of these data sets contains measurements of patients with a certain cancer type. All data sets were downloaded from the database The Cancer Genome Atlas Project (TCGA) [21]. One of these data sets, the data set 'PRAD' (cf. Table 1), was excluded a posteriori for reasons unrelated to the results obtained with this data set, see Section 3 for details. The following blocks were present among the data sets: clinical information, miRNA data, mutation data, copy number variation measurements, and RNA data. Not every block was available for each data set: For two data sets ('CESC' and

'GBM') the miRNA block was not available and for one data set ('READ') there was no mutation data. While the numbers of variables available for each block do differ between the data sets, they are all in the same order of magnitude across data sets. On average, the following numbers of variables were available for each block: 4.5 (clinical block), 780.3 (miRNA block), 16,579.8 (mutation block), 57,888.4 (CNV block), 23,442.6 (RNA block). Table 1 provides an overview of the data sets. In Supplementary Table S1 (Supplementary File 1) we provide a more detailed overview of the data sets in which we provide the numbers of variables available for each block in each data set.

We used $k$ nearest neighbors imputation to impute missing values in the clinical block in data sets with more than two clinical covariates. We used the function `knnImputation` from the R package `DMwR` [22]. In cases with only two variables this was not possible because `knnImputation` is only applicable for a minimum of three variables. Here, we used univariate logistic regression for imputation. Note that by performing the imputation before the cross-validation used for performance estimation, we performed incomplete cross-validation, which, depending on the analysis step performed outside of the cross-validation, can lead to overoptimism in the resulting prediction performance estimates [23]. However, performing incomplete cross-validation with respect to imputation has been found to not affect the performance estimates to any relevant degree [23].

### 2.4.2 Study design

The following methods were compared: RSF, VarProb, SplitWeights, BlockVarSel, RandomBlock, and BlockForest. Note that since the data sets considered in the comparison study are survival data sets, the latter five forest variants for multi-omics data are variants of the RSF algorithm. We used 5-fold cross-validation repeated five times to measure the performance of each method on each data set. As a performance metric we used Harrell's C index for which we use the short form C index in the following. First, we calculated the C index on the left out fold in each iteration of the repeated cross-validation and, second, took the average of these values across the iterations and repetitions of the cross-validation. In very rare cases the C index was not calculable on the left out fold for an iteration, which was due to lack of comparable pairs of observations. These iterations were left out when calculating the corresponding averages.

As noted in Section 2.3 we set $N_{\text{sets}} = 300$ and $num.trees_{\text{pre}} = 1500$ for the forest variants. After optimizing the tuning parameter value sets, we constructed forests with 2000 trees. For RSF we optimized the values of $mtry$ using grid search: First, for each $mtry \in \{\lceil x\sqrt{p}\rceil | x \in \{0.1, 0.25, 0.5, 1, 2\}\}$ we constructed an RSF with 1500 trees and, second, used that of the $mtry$ values tried that was associated with the smallest

Table 1: Overview of the data sets used in the comparison study

| Name | Cancer type | Sample size | Uncensored observations |
|---|---|---|---|
| BLCA | Bladder Urothelial Carcinoma | 310 | 32 % |
| BRCA | Breast Invasive Carcinoma | 863 | 9 % |
| CESC | Cervical Squamous Cell Carcinoma and Endocervical Adenocarcinoma | 206 | 15 % |
| COAD | Colon Adenocarcinoma | 350 | 22 % |
| ESCA | Esophageal Carcinoma | 121 | 21 % |
| GBM | Glioblastoma Multiforme | 154 | 73 % |
| HNSC | Head and Neck Squamous Cell Carcinoma | 411 | 35 % |
| KIRC | Kidney Renal Clear Cell Carcinoma | 322 | 22 % |
| KIRP | Kidney Renal Papillary Cell Carcinoma | 249 | 10 % |
| LGG | Brain Lower Grade Glioma | 454 | 21 % |
| LIHC | Liver Hepatocellular Carcinoma | 298 | 28 % |
| LUAD | Lung Adenocarcinoma | 424 | 30 % |
| LUSC | Lung Squamous Cell Carcinoma | 365 | 39 % |
| OV | Ovarian Serous Cystadenocarcinoma | 261 | 54 % |
| PAAD | Pancreatic Adenocarcinoma | 142 | 49 % |
| PRAD | Prostate Adenocarcinoma | 425 | 2 % |
| READ | Rectum Adenocarcinoma | 138 | 16 % |
| SARC | Sarcoma | 183 | 16 % |
| SKCM | Skin Cutaneous Melanoma | 264 | 25 % |
| STAD | Stomach Adenocarcinoma | 284 | 27 % |
| UCEC | Uterine Corpus Endometrial Carcinoma | 503 | 13 % |

The following information is given: Name of the data set, cancer type, sample size and the percentage of observations for which the survival time was uncensored. Note that the TCGA Project ID of each data set is "TCGA-[Name]", with "[Name]" being the name of the data set (given in the first column).

out-of-bag prediction error. Subsequently, we constructed a forest with 2000 trees as in the case of the variants.

In the case of blocks with more than 2500 variables, we conducted supervised variable selection on the training data sets within cross-validation, selecting the 2500 variables that featured the smallest $p$-values in univariate Cox regression. This was performed both, for computational efficiency and because for ultra high-dimensional omics data types most variables can be assumed to be without effect. As split criterion we used the log-rank test statistic. Because of the computational expense of log-rank tests, evaluating each possible split point for each considered variable is too expensive in the context of forests applied to high-dimensional data. Therefore, for each variable tried, we merely considered one randomly sampled split point. This split point selection procedure is known as extremely randomized trees [24] for classification and regression trees. In an extensive comparison study, extremely randomized trees have been found to feature similar or even slightly better prediction performance than conventional trees for which each split point is tried out for each considered variable [24].

We performed the analysis once using all available blocks for each data set and once using only the clinical block and the RNA block. As described in Section 1, the latter setting of having clinical covariates plus a certain type of omics data available is frequently occurring in practice. For this reason we were interested in comparing the methods also with respect to their performance when applied in this setting. We chose the RNA block as the involved omics block in this analysis, because this omics data type is the one that is most commonly found in practice of those available for the considered data sets.

# 3 Results

## 3.1 Multi-omics data

Figure 1 shows the results of the multi-omics comparison study. The upper panel shows boxplots of the mean cross-validated C index values obtained for all methods and data sets, the middle panel shows the differences between the mean cross-validated C index values obtained for the different methods and those obtained using RSF, and the lower panel shows boxplots of ranks of the methods calculated using the mean cross-validated C index values of the data sets. On average, BlockForest and RandomBlock performed best among all methods. Specifically, BlockForest achieved the highest median C index across the data sets and a median rank of two. The mean ranks of the methods are as follows (from best to worst): 1.95 (BlockForest), 2.40 (RandomBlock), 3.75 (RSF), 4.10 (BlockVarSel), 4.20 (VarProb), 4.60 (SplitWeights). In Supplementary Figures S1 and S2 (Supplementary File 1) the C index values obtained for the individual repetitions of the cross-validation are shown separately for each data set.

In Figure 2, for each data set, the differences between the data set spe-

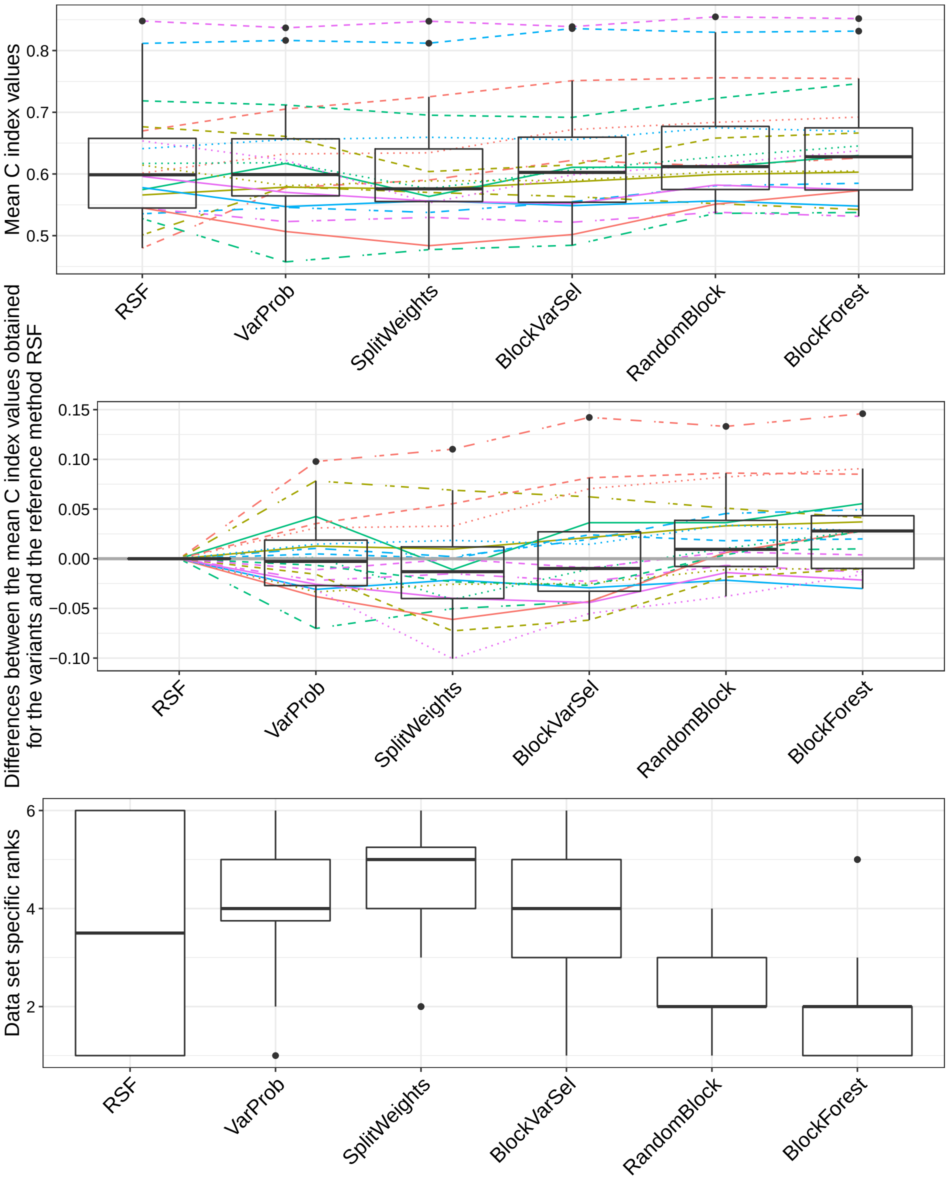

Figure 1: Multi-omics data: Performances of all six considered methods. Upper panel: Mean C index values for each of the 20 data sets and each of the six methods considered. Middle panel: Differences between the mean C index values obtained using the different methods and that obtained using RSF. Lower panel: Data set specific ranks of each method among the other methods in terms of the mean cross-validated C index values. The colored lines connect the results obtained for the same data sets.

cific mean C index values obtained using BlockForest and RSF and the differences between the data set specific mean C index values obtained using RandomBlock and RSF are shown. Both BlockForest and RSF were better than RSF for 14 of the 20 data sets (70%), where the degrees of these improvements differ quite strongly across the data sets.

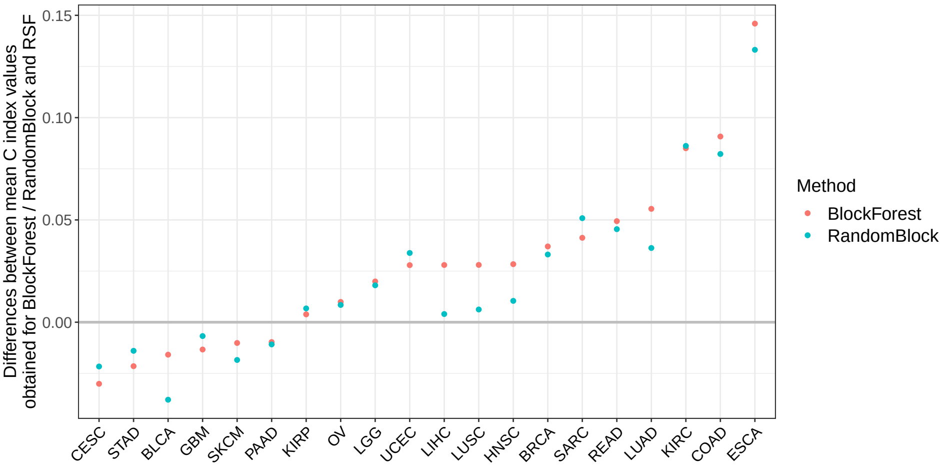

Figure 2: Multi-omics data: Performances of BlockForest and RandomBlock relative to that of RSF. Differences between the mean C index values obtained using BlockForest / RandomBlock and that obtained using RSF, ordered by difference between the values obtained for BlockForest and RSF.

In Section D of Supplementary File 1 we analyze the influence of data set characteristics on the performance of BlockForest relative to that of RSF. The improvements of BlockForest over RSF tended to become greater for larger sample sizes and slightly lower in cases in which single blocks dominated the other blocks with respect to their importances for prediction. Moreover, for weaker biological signals the degrees of improvement attained through using BlockForest instead of RSF varied more strongly, that is, for weaker signals there were more often stronger improvements, but also more often merely weak improvements and also (slight) impairments.

We performed statistical testing to investigate whether the improvements over RSF are statistically significant. We performed a one-sided paired Student's $t$-test for each variant with the null hypothesis of non-inferiority of RSF over the considered variant. The five $p$-values for the variants were adjusted for multiple testing with the Bonferroni-Holm method. The following adjusted $p$-values were obtained: 1.000 (VarProb), 1.000 (SplitWeights), 1.000 (BlockVarSel), 0.056 (RandomBlock), 0.027 (BlockForest). Thus, BlockForest performed significantly better than RSF, while RandomBlock showed merely a weakly significant improvement over RSF (adjusted $p$-value larger than 0.05, but smaller than 0.10).

In Section E of Supplementary File 1 we provide in-depth analyses of the optimized values of the tuning parameters associated with the different variants for each data set. In the following we will merely present some important conclusions obtained in these analyses, for details see Supplementary File 1. As noted in Section 2, the optimized block selection probabilities $b_1, \ldots, b_M$ associated with RandomBlock can give indications of the relative importances of the different blocks for prediction. We obtained the following mean block selection probabilities across the data sets (sorted from highest to lowest): 0.43 (mutation), 0.29 (RNA), 0.12 (clinical), 0.11 (CNV), 0.07 (miRNA). Thus, the mutation block and the RNA block seem to be by far the most important blocks. Moreover, we found the correlations between the $b_m$ values (averaged per data set) obtained for the mutation block and that obtained for the RNA block to be strongly negative. This suggests that there is a strong overlap in the information contained in these two blocks. The correlation between the $b_m$ values of the clinical block and that of the mutation block was negative, but that between the $b_m$ values of the clinical block and that of the RNA block was very weak and positive. This suggests that the additional predictive value of the mutation block to the clinical block might in general be smaller than that of the RNA block to the clinical block. A strong additional predictive value of the RNA block over the clinical block would make it particularly effective to exploit the predictive information contained in clinical covariates in situations in which such variables are available in addition to RNA measurements.

As mentioned previously we excluded the data set PRAD a posteriori from the results. The survival times in this data set were censored for more than 98% of the patients, leaving merely seven observed events as opposed to 418 censored events. This resulted in extremely unstable performance estimates, since the C index cannot compare pairs of observations for which the shorter time is censored. Note, however, that it is a very delicate issue to exclude a data set from a study after having observed the results for this data set. Doing so offers potential for mechanisms related to fishing for significance [25, 26]. We obtained the following mean cross-validated C index values for the data set PRAD (sorted from largest to smallest): 0.584 (RSF), 0.545 (BlockVarSel), 0.522 (BlockForest), 0.504 (SplitWeights), 0.502 (RandomBlock), 0.467 (VarProb). Note that the variant BlockForest that performed best overall, performed worst in comparison to RSF for this data set among all data sets. We suppose this bad result obtained with BlockForest to be related to overfitting with respect to the optimized tuning parameter values. While Random Forest is quite robust with respect to the choice of *mtry* [27], this is likely not the case for the variants with respect to their block-specific tuning parameter values. As described above, the C index values obtained for this data set are very variable due to the small number of observed survival times, which is why the optimized tuning parameter values can be expected to be very unreliable for this data set. Therefore, the optimized tuning parameter values may be far from the values that are actually optimal for these parameters, which could explain the bad prediction performance.

### 3.2 Clinical covariates plus RNA measurements

Figure 3 shows the results obtained for the analysis in which we used only the clinical block and the RNA block. The results are presented in the same form as in Figure 1. Again, BlockForest performed best, achieving the highest average C index value and a median rank of one.

Figure 4 shows the differences between the data set specific mean C index values obtained using BlockForest and RSF and the differences between the data set specific mean C index values obtained using RandomBlock and RSF. BlockForest performed better than RSF for 17 of the 20 data sets (85%), where for a part of these data sets the improvement of BlockForest over RSF was strong, while there was only a mild improvement for other data sets. RandomBlock showed an improvement over RSF for 15 of the 20 data sets (75%). The distributions of the ranks of the methods (Figure 3, lower panel) reveal that BlockForest sets itself apart more strongly from RandomBlock than in the analysis of the multi-omics data. Moreover, excluding SplitWeights, now also the other variants outperform RSF, with the latter having a median rank of six. The performance of SplitWeights is very similar to that of RSF (Figure 1, middle panel) for the great majority of data sets. The reason for this is probably that with SplitWeights as with RSF the clinical covariates are selected very infrequently, which is why the obtained predictions do not differ strongly between these two methods if there is only a clinical block plus a single omics block. We obtain the following mean ranks for the methods (from best to worst): 1.80 (BlockForest), 3.10 (VarProb), 3.20 (RandomBlock), 3.70 (BlockVarSel), 4.55 (RSF), 4.65 (SplitWeights). Note that VarProb worked much better here than for the multi-omics data. However, since BlockForest still performed considerably better than VarProb, the latter cannot be recommended for the case of having a clinical block plus an omics block available (nor for the multi-omics case). The C index values obtained for the individual repetitions of the cross-validation separately for each data set are shown in Supplementary Figures S14 and S15 (Supplementary File 1).

In the same manner as for the multi-omics data, we performed $t$-tests to test for superiority of each of the variants over RSF, again adjusting for multiple testing using Bonferroni-Holm. We obtained the following adjusted $p$-values: 0.019 (VarProb), 0.238 (SplitWeights), 0.026 (BlockVarSel), 0.019 (RandomBlock), 0.01 (BlockForest). Thus, there is a significant improvement over RSF for all of the variants except for SplitWeights.

As for the analysis of the multi-omics data, we investigated the influ-

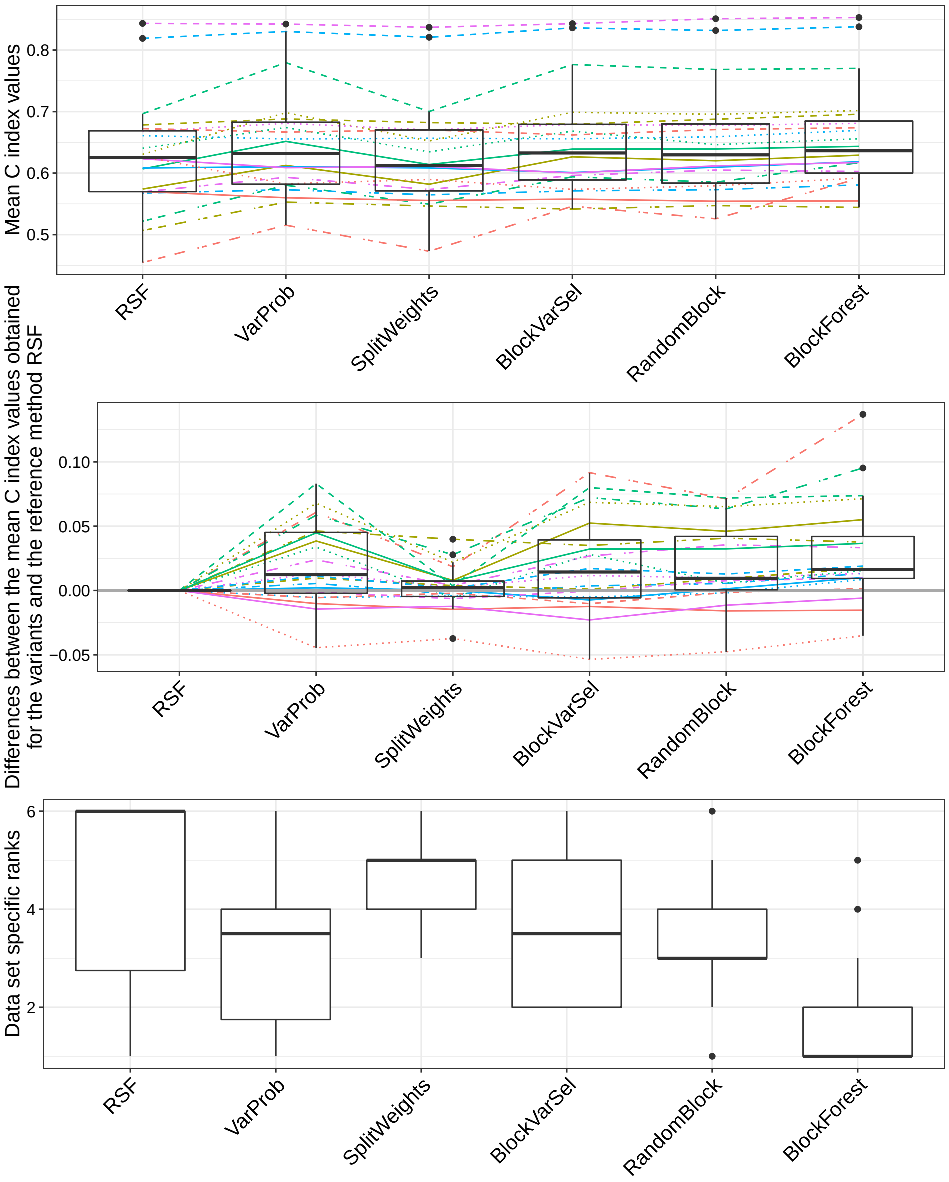

Figure 3: Clinical covariates plus RNA measurements: Performances of all six considered methods. Upper panel: Mean C index values for each of the 20 data sets and each of the six methods considered. Middle panel: Differences between the mean C index values obtained using the different methods and that obtained using RSF. Lower panel: Data set specific ranks of each method among the other methods in terms of the mean cross-validated C index values. The colored lines connect the results obtained for the same data sets.

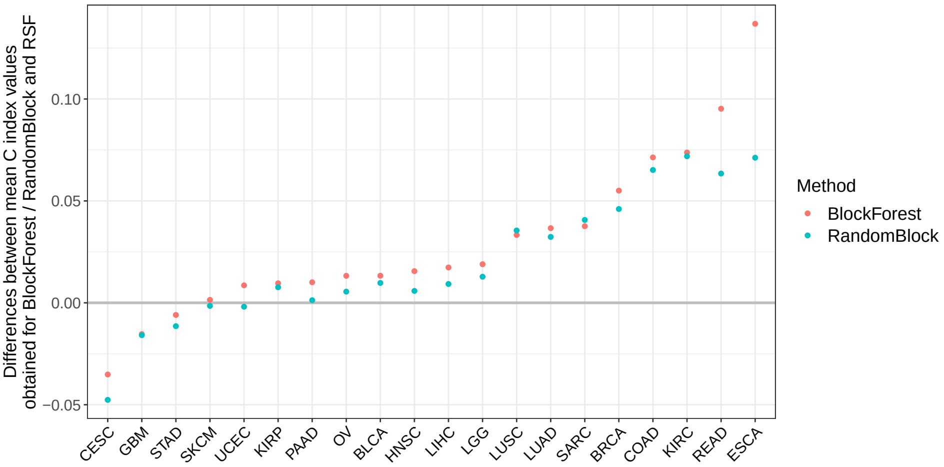

Figure 4: Clinical covariates plus RNA measurements: Performances of BlockForest and RandomBlock relative to that of RSF. Differences between the mean C index values obtained using BlockForest / RandomBlock and that obtained using RSF, ordered by difference between the values obtained for BlockForest and RSF.

ences of different data set characteristics on the performance of BlockForest relative to that of RSF. There was no clear relation between the sample sizes and the improvements of BlockForest over RSF, which could be related to the fact that when considering only two blocks instead of five (or four), less tuning parameter values have to be optimized. Moreover, the larger the $b_m$ value of the clinical block optimized using RandomBlock was, the greater the improvement of BlockForest over RSF tended to be. This suggests that BlockForest performs particularly strong in situations in which there are highly predictive clinical covariates. Lastly, the improvements of BlockForest over RSF tended to be greater for weaker biological signals. See Section G of Supplementary File 1 for details.

Analogous to the multi-omics data case, in Section H of Supplementary File 1 we provide detailed analyses of the optimized tuning parameter values associated with the variants for each data set. The mean optimized block selection probabilities associated with RandomBlock across data sets were as follows: 0.24 (clinical), 0.76 (RNA). In the median, the block selection probability of the RNA block was 3.7 higher than that of the clinical block. This can be interpreted as that the RNA block was in the median 3.7 times as important for prediction as the clinical block for this collection of data sets when considering only the clinical block and the RNA block. For further details, see Supplementary File 1.

# 4 Discussion

The information overlap between the blocks that manifests itself in negative correlations between the $b_m$ values (associated with RandomBlock) obtained for different blocks, might help to explain why the two methods, block forest aka BlockForest and RandomBlock performed best in our comparison study. With both of these methods the block choice is randomized for each split. By avoiding to consider all blocks for each split, the splits differ more strongly, because the predictive information considered with different splits is more heterogeneous. If two blocks with highly overlapping predictive information are always considered simultaneously, the splits will mostly consider that part of the predictive information contained in these blocks that is common to both blocks. This is because the overlapping information is considered twice when sampling from both blocks and because this overlapping information, mirroring important biological dependencies, tends to be strong. By contrast, if the two blocks blocks are (also) considered individually, the parts of the predictive information that are distinctive to either block will be exploited more in the splitting. Thus, randomizing the block selection helps to exploit the individual contributions of the different omics data types to the predictive information contained in the covariates.

The fact that four of the five variants delivered better results than RSF for the case of using clinical covariates plus RNA measurements indicates that it is crucial to treat the clinical covariates differently than the omics variables. As already mentioned in Section 1, clinical covariates often have high prognostic relevance. Due to their small number in comparison to the omics variables, these variables are too infrequently used for splitting in a standard random (survival) forest, which is why their prognostic value is not exploited in standard random (survival) forest. Of course, the latter is also true for any other prediction method for high-dimensional variable data applied to clinical covariates plus a certain type of omics variables that does not differentiate between clinical and omics variables. While it can therefore be very effective to allow for prioritization of the clinical block over high-dimensional omics blocks, less benefit can be expected from imposing different priorizations between omics blocks of similar size. This is because, if the variables from two omics blocks of similar size are assigned the same prioritizations, the variables from the more informative block will still be used more often for splitting than the variables from the less informative block. These considerations also affect the interpretation of the results obtained for the analysis of the multi-omics data. As described previously, in the analysis of the multi-omics data, for blocks with more than 2500 variables, we pre-selected the 2500 variables with smallest $p$-values from univariate Cox regressions. By doing so, three of the four omics blocks had 2500 variables after pre-selection. The only block with less than 2500 variables was the miRNA block, which was, however, non-informative for almost all data sets.

Thus, all influential omics blocks had the same numbers of variables (after pre-selection) in the analysis of the multi-omics data. For this reason, the different prioritizations of the omics blocks might not have been that crucial compared to situations in which the omics blocks feature highly differing numbers of variables. In such situations, the expected performance gain by using BlockForest instead of standard RSF might be larger.

Comparing the mean C index values obtained when using all available blocks (Figure 1) and when using clinical covariates plus RNA measurements (Figure 3) an interesting observation can be made: The mean C index values are in most cases higher when using clinical covariates plus RNA measurements only than when using all available blocks. This is true for both standard RSF as well as for the variants. When using all available blocks there is more predictive information available than when using only a single block, which is why it appears contradictory at first that the prediction performance is worse when considering all available blocks for prediction. However, we can make three observations that help to explain, why the prediction rules obtained using the combination of the clinical block and the RNA block performed that strongly. First, the results obtained in Section E of Supplementary File 1 suggested that the predictive information contained in the two most important blocks, the mutation block and the RNA block, is highly overlapping. The evidence that supported this assumption was that the method RandomBlock in the great majority of cases attributed a very large value of the selection probability to one of these two blocks and a very small value to the respective other block. The strong overlap in information between the mutation block and the RNA block has the effect that there tends to be limited gain in prediction accuracy by including the mutation block in addition to the RNA block even though the mutation block does contain much predictive information. Second, the RandomBlock algorithm attributed in most cases small selection probabilities to the remaining blocks, which suggests that these most often provided limited additive predictive value over the mutation block and/or the RNA block. Third, the results obtained with all blocks also suggested that the additional predictive value of the RNA block over the clinical block is higher than that of the mutation block over the clinical block. For these three reasons, in practice it should in many cases be sufficient to consider only the clinical information plus the RNA measurements. Such prediction rules are quite convenient to handle. This is because it is merely required that the necessary clinical information and the RNA measurements are available instead of measurements of a multitude of different omics data types, both, for the purpose of constructing the prediction rule and, even more importantly, for the purpose of applying it.

As noted in Section 1, the considered variants are applicable for any types of outcomes (e.g., categorical or continuous outcomes), for which there exists a random forest variant. Our comparison studies were limited to survival outcomes. However, the discussed advantage of block forest (and RandomBlock) that it is better able to exploit the individual contributions of the different omics data types holds also for other types of outcomes.

## 5 Conclusions

Using a collection of 20 real multi-omics data sets we compared the prediction performance of the random survival forest algorithm with five different candidate variants of the latter that take the block structure of multi-omics data into account. We also considered the common situation of having only the clinical block plus a single type of omics data (in our case RNA measurements) available. Both, for the latter case and for the multi-omics data the variant "BlockForest" or "block forest" performed best and significantly better than the reference method random survival forest. Therefore, we recommend using block forest in applications to exploit effectively the predictive information contained in combinations of clinical data and one or several types of omics data. The other random forest variants can be consulted for academic purposes, for example, in the context of further methodological developments.

**Acknowledgements** The authors thank Shuangge Ma and Xingjie Shi for advice and for providing us with R scripts they had used in [1]. This work was supported by the German Science Foundation (DFG-Einzelförderung BO3139/6-1 to Anne-Laure Boulesteix).

**Supplementary Material**

**Supplementary Material 1**: PDF file with further contents referred to in the paper

URL (use copy and paste): `http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/070_drittmittel/hornung/blockforest_suppfiles/suppmat1_hornungwrighttr.pdf`

This PDF file contains the following sections:

- A Algorithms used for generating random sets of tuning parameter values in the tuning parameter value optimization
- B Overview of the data sets used in the comparison study
- C Multi-omics data: C index values obtained for the individual repetitions of the cross-validation
- D Multi-omics data: Analysis of the influence of data set characteristics on the performance of BlockForest relative to that of RSF
- E Multi-omics data: Optimized block-specific tuning parameter values associated with the different variants

- F Clinical covariates plus RNA measurements: C index values obtained for the individual repetitions of the cross-validation
- G Clinical covariates plus RNA measurements: Analysis of the influence of data set characteristics on the performance of BlockForest relative to that of RSF
- H Clinical covariates plus RNA measurements: Optimized block-specific tuning parameter values associated with the different variants

**Supplementary Material 2**: Electronic Appendix

URL (use copy and paste): `http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/070_drittmittel/hornung/blockforest_suppfiles/suppmat2_hornungwrighttr.zip`

This folder contains all R Code written to perform the analyses presented in this article and in Supplementary Material 1 as well as Rda files enabling fast evaluation of the results. The pre-processed versions of the data sets as used in the analysis are not included in Supplementary Material 2 due to their large sizes. However, they are available from the corresponding author upon request.

# References

[1] Zhao Q, Shi X, Xie Y, Huang J, Shia B, Ma S. Combining multidimensional genomic measurements for predicting cancer prognosis: observations from TCGA. Brief Bioinform. 2015;16(2):291–303.

[2] Huang S, Chaudhary K, Garmire LX. More is better: Recent progress in multi-omics data integration methods. Front Genet. 2017;8:84.

[3] Lanckriet GRG, De Bie T, Cristianini N, Jordan MI, Noble WS. A statistical framework for genomic data fusion. Bioinformatics. 2004;20(16):2626–2635.

[4] Simon N, Friedman J, Hastie T, Tibshirani R. A Sparse-Group Lasso. J Comput Graph Stat. 2013;22(2):231–245.

[5] Boulesteix AL, De Bin R, Jiang X, Fuchs M. IPF-LASSO: Integrative $L_1$-penalized regression with penalty factors for prediction based on multi-omics data. Comput Math Method M. 2017;p. 1–14.

[6] Vazquez AI, Veturi Y, Behring M, Shrestha S, Kirst M, Resende MFR Jr, et al. Increased proportion of variance explained and prediction accuracy of survival of breast cancer patients with use of whole-genome multiomic profiles. Genetics. 2016;203:1425–1438.

[7] Mankoo PK, Shen R, Schultz N, Levine DA, Sander C. Time to recurrence and survival in serous ovarian tumors predicted from integrated genomic profiles. PLoS ONE. 2011;6(11):e24709.

[8] Park MY, Hastie T. $L_1$-regularization path algorithm for generalized linear models. J R Stat Soc Ser B. 2007;69:659–677.

[9] Seoane JA, Day INM, Gaunt TR, Campbell CA. A pathway-based data integration framework for prediction of disease progression. Bioinformatics. 2014;30(6):838–845.

[10] Fuchs M, Beißbarth T, Wingender E, Jung K. Connecting high-dimensional mRNA and miRNA expression data for binary medical classification problems. Comput Meth Programs Biomed. 2013;111(3):592–601.

[11] Klau S, Jurinovic V, Hornung R, Herold T, Boulesteix AL. Priority-Lasso: a simple hierarchical approach to the prediction of clinical outcome using multi-omics data. BMC Bioinform. 2018;19:322.

[12] Aben N, Vis DJ, Michaut M, Wessels LFA. TANDEM: a two-stage approach to maximize interpretability of drug response models based on multiple molecular data types. Bioinformatics. 2016;32(17):i413–i420.

[13] Boulesteix AL, Sauerbrei W. Added predictive value of high-throughput molecular data to clinical data and its validation. Brief Bioinform. 2011;12(3):215–229.

[14] De Bin R, Sauerbrei W, Boulesteix AL. Investigating the prediction ability of survival models based on both clinical and omics data: two case studies. Stat Med. 2014;33:5310–5329.

[15] Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS. Random survival forests. Ann Appl Stat. 2008;2:841–860.

[16] Bou-Hamad I, Larocque D, Ben-Ameur H. A review of survival trees. Stat Surv. 2011;5:44–71.

[17] Yosefian I, Farkhani EM, Baneshi MR. Application of random forest survival models to increase generalizability of decision trees: A case study in acute myocardial infarction. Comput Math Methods Med. 2015;p. 1–6.

[18] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning. 2nd ed. New York, NY, USA: Springer; 2009.

[19] Strobl C, Boulesteix AL, Zeileis A, Hothorn T. Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinform. 2007;8(1):25.

[20] Wright MN, Ziegler A. ranger: A fast implementation of random forests for high dimensional data in C++ and R. J Stat Softw. 2017;77:1–17.

[21] Ley TJ, Miller C, Ding L, Raphael BJ, Mungall AJ, Robertson A, et al. Genomic and epigenomic landscapes of adult de novo acute myeloid leukemia. N Engl J Med. 2013;368:2059–2074.

[22] Torgo L. DMwR: Functions and data for 'Data Mining with R'; 2013. R package version 0.4.1.

[23] Hornung R, Bernau C, Truntzer C, Wilson R, Stadler T, Boulesteix AL. A measure of the impact of CV incompleteness on prediction error estimation with application to PCA and normalization. BMC Med Res Methodol. 2015;15:95.

[24] Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. Mach Learn. 2006;63:3–42.

[25] Yousefi MR, Hua J, Sima C, Dougherty ER. Reporting bias when using real data sets to analyze classification performance. Bioinformatics. 2010;26(1):68–76.

[26] Boulesteix AL, Hable R, Lauer S, Eugster MJA. A statistical framework for hypothesis testing in real data comparison studies. Am Stat. 2015;69(3):201–212.

[27] Probst P, Bischl B, Boulesteix AL. Tunability: Importance of hyperparameters of machine learning algorithms; 2018. arXiv/1802.09596.